MASKED FACE RECOGNITION WITH SIAMESE NETWORK-BASED METRIC LEARNING

KEVIN RASIKBHAI AKBARI

Final Thesis Report

JUNE 2022

DEDICATION

Dedicated to my mother Ranjan, and father Rasik, who have always supported me at every step of my life.

ACKNOWLEDGEMENTS

I would like to thank Dr. Ahmed Kaky for his guidance and support resulting in the writing of this report. I would also like to acknowledge the help provided by Dr. Silvester Czanner vide all the sessions taken via Upgrad portal.

My sincere thanks also to my thesis supervisor Aayushi Verma for all the valuable inputs in framing and formatting the structure of this thesis report. A special thanks to my brother Kishan Akbari for assisting me in gathering relevant research work to complete this report.

ABSTRACT

Face recognition has become a widespread technology everywhere. It is not only being used to grant access to different places but is also considered useful in identifying prospective threats by keeping track of the movements of people. However, these days more and more people have started wearing masks to protect themselves from harmful fumes, pollution, viruses, etc. When it comes to masked-face recognition, models trained for unmasked face recognition task fails to perform to the desired level. Occlusion caused due to the usage of the mask makes it tough for machine learning algorithms to differentiate between genuine and impostor pairs of face images. This work studies a Siamese network-based model for the development of a robust masked face recognition system that can give accurate results in tasks such as face verification - one on one mapping and face recognition - one to many mapping. We also use pre-trained attention-based layers to build the core model of the network architecture to focus on the relevant portion of face images. The study starts with using the pre-trained SENet50 model for getting the face embeddings to training the last 70 layers of the pre-trained model with Siamese architecture and custom loss, achieving the highest difference between the median Genuine and Impostor distances as 0.441. The experimental results conclude that the attention based core model aids to focus more on regions in and around the eyes rather than on the occluded areas for the masked face recognition task.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CBAM……..	Convolutional Block Attention Module
CIFAR……..	Canadian Institute for Advanced Research
CNN…….....	Convolutional Neural Network
COVID-19……..	Coronavirus Disease of 2019
FDC…….....	Fisher Discriminant Contrastive
FDT…….....	Fisher Discriminant Triplet
GAN….....	Generative Adversarial Network
GMean…….	Genuine Mean Distance
ILSVRC……...	ImageNet Large Scale Visual Recognition Challenge
IT…….	Information Technology
IMean……...	Impostors Mean Distance
IMFD……...	Indian Masked Faces in the Wild
LFW……	Labeled Faces in the Wild
ResNet…….	Residual Neural Network
RMFD…….	Real-World Masked Face Dataset
RMFRD……	Real-world masked face recognition dataset
RMFVR……	Real-world masked face verification dataset
SE…….	Squeeze and Excitation
SMFRD……	Simulated masked face recognition datasets
SOTA……...	State of the Art

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

Ever since the discovery of Convolutional Neural Network (CNN) architectures, the challenge of modeling unstructured data is eased to a great extent. It is especially true for the domain of image, and video analysis wherein these computer vision algorithms are being used predominantly. The success of CNN in this domain comes from its ability to detect higher-end correlations/patterns amongst the neighboring features/pixels in the image.

Convolutional Neural Networks are being used for image classification tasks wherein they are required to be trained with enough training examples per class, to achieve good performance. However, when it comes to the problem of face verification/ recognition, the number of classes can be in the thousands. Hence, it is not always feasible to obtain a sufficient number of images per class to train the network, which may lead to the problem of class imbalance. A common solution to the above problem is to use a metric learning-based approach and learn semantically-sound lower dimension representation of higher dimension features (Chopra et al., 2005). Here, semantically sound means finding a function that can preserve the similarity and/or dissimilarity of different higher dimensional data points in the corresponding lower dimension feature space. Researchers have proposed Siamese Network to model such a function.

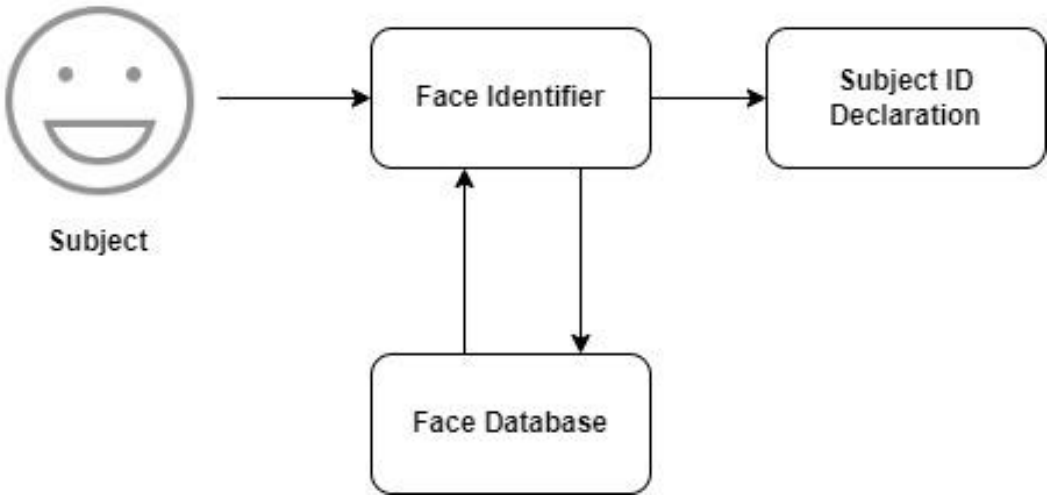


Figure 1.1 Typical arrangement of face identification system

One of the tasks from this domain is the challenge of face recognition and verification. A typical arrangement of the face identification system is depicted in Figure 1.1 above. Face recognition is a many-to-one mapping where a target face and a database of different subject face images are available and the task is to declare the identity of the target subject based on the distance measure. Likewise, face verification is one-to-one mapping where a threshold is present that decides whether two face images belong to the same subject or not (Ding et al., 2020). Face recognition, verification tasks are not new to the research community. Solutions with human-level performance (Wang and Deng, 2021) are in place to precisely differentiate between face images of the same subject - genuine pair, different subject - impostor pair. However, wearing a face mask has become a common trend among people these days due to rising pollution levels, the outbreak of deadly viruses, etc. Making people remove their masks for their identity verification purpose is neither logical nor safe. Besides, the existing face recognition systems have their limitations when it comes to occluded face recognition and verification tasks. The purpose of this study is to understand the problem of masked-face recognition and propose a probable solution that can improve the performance of the existing unmasked face recognition models on this task.

1.2 Problem Statement

Face recognition and verification technology are being used these days for non-trivial tasks like authentication, attendance systems, phone unlocking, finding the subject of interest from surveillance clips, etc. Researchers have proposed various methods to achieve state-of-the-art metric values. As per the author's knowledge, probably the first breakthrough occurred with DeepFace (Taigman et al., 2014) model. They derived face representation from modeled 3D faces and were able to achieve close to 97.35% accuracy on the Labeled Faces in the Wild (LFW) dataset. Baidu (Liu et al., 2015) proposed a deep CNN-based model, employing metric learning and achieved pairwise verification accuracy close to 99.77% on the LFW dataset. They argued that their proposed model could achieve this feat because of the usage of features from different patches of the image. In addition to experimenting with the different network architectures, researchers have also explored various loss functions and proposed losses viz. Marginal Loss (Deng et al., 2017), Cosface (Wang et al., 2018), and Arcface (Deng et al., 2019), etc., and were able to achieve SOTA accuracy on the LFW dataset.

Obstruction caused due to foreign object(s) kept against the area of interest is called Occlusion. Facial occlusion could be caused by apparel items like a mask, scarf, cap, hat, etc. General occlusion scenarios that are quite normal are depicted below in Figure 1.2.

Figure 1.2 Different occlusion scenarios

Mask has become a common cause of occlusion as the practice of wearing the same while in public places has become a common trend these days due to rising air pollution, the spread of contagious diseases, the outbreak of viruses, etc. Hence, the application of machine learning techniques on masked faces has become a trending area of research. Broadly there are two main tasks namely face mask detection and masked face recognition. The objective behind the former is to check whether the subject is wearing a mask or not. While the latter challenge is to identify the subject while the mask is covering the face. The focus of the proposed work is on the problem of masked-face recognition.

Recognizing the face of the subject while having the mask on is an arduous task since the major portion of the face gets occluded. When it comes to the task of face recognition, nose and mouth areas are important as semantically they reveal the identity of the subject. So, the normal algorithm for face recognition becomes less useful (Mishra et al., 2021b) for the masked face recognition task. Of let, computer vision researchers have turned their focus toward this field but still, it is relatively less developed. Two broad approaches are being adopted for solving this problem. One is recovery/unmasking of the masked part with generative models and the second is discarding the occluded/masked part of the face and considering only the visible region for the analysis. (Ud Din et al., 2020), (Li et al., 2020) proposed a Generative Adversarial Network (GAN) based method to recover the masked portion of the face and produced a qualitatively as well as quantitatively well-performing model. However, the recovery-based methods are computationally expensive.

Several models have been developed which altogether discard the occluded region and use the remaining portion of the face for the task of recognition. (Hariri, 2021) discarded masked area and used the non-occluded region to fine-tune the pre-trained CNN model. The features extracted from the deep CNN model were used to achieve high classification accuracy on the Real-World Masked Face Dataset. The features extracted from the unmasked portion of the image are more useful and hence more weightage should be given to them as compared to masked region features. (Song et al., 2019) used the pairwise differential Siamese network to eliminate the corrupt portion of the face and utilized the rest for face recognition. (Li et al., 2021) adopted a mix of discarding and attention mechanism to achieve high metric value in the task of masked face recognition.

While identifying a person with a mask, humans intuitively focus on the area near the eyes. It is known that in the masked face recognition task, the occlusion always starts from just above the nose tip and ends at the chin part of the face, and hence human eyes effortlessly pay more attention to the non-occluded part, but the model doesn't. With this knowledge in mind, it is proposed to incorporate the attention-based module in deep CNN and make the model focus more on the unmasked portion of the face rather than the masked portion for the proposed problem.

1.3 Aim and Objectives

The main aim of this research is to propose an attention-based approach to enhance the capability of the existing face recognition model using the Real World Masked Face Dataset (RMFD). The goal of this project is to come up with a robust masked face recognition model such that occlusion due to the mask does not restrict the ability of the model to differentiate between the impostor and genuine face images.

The research objectives formulated based on the aim of this study are as follows:

- To investigate the capability of an attention-based convolutional neural network in achieving the tasks of masked and unmasked face recognition.
- To experimentally determine the optimal parameter settings of the model with the Siamese network-based training.
- To evaluate the performance of the Siamese network with various distance measures.

1.4 Research Questions

The research questions proposed to be answered for the different research objectives set are brought out below:

- Whether the attention mechanism effective in tackling the less important region generated due to occlusion?
- Is it possible to train a deep learning model for the masked face recognition task using Siamese architecture?
- Do different distance measures have different results in Siamese network-based learning?

1.5 Scope of the Study

The scope of this study is to explore the effectiveness of incorporating attention mechanisms in the masked face recognition task. Tasks like human face detection in the image, and face-mask detection are not the objectives of this study. Further, the development of a state-of-the-art (SOTA) attention module/ layer for a deep convolutional neural network model is also not in the current scope. The scope of the study is explicitly defined beforehand to adhere to the limited time, and resources available for the proposed project.

1.6 Significance of the Study

Given the current scenario of pandemic and its prospects, working towards the development of occlusion robust systems is of utmost importance. Such technologies could help in promoting the usage of face masks and hence reducing the spread of deadly viruses. It is expected that this study would bring the focus of the research community towards the application of attention mechanisms in solving the problem of face recognition with different occlusion levels. Usage of such technology could aid in upkeeping law and order in society with minimal impact on basic human rights.

1.7 Structure of the Study

The structure of this study is divided into chapters. The content of those chapters is as follows. Chapter 1 presents total 5 sub-sections. Section 1.1 briefs the background of the research in the face recognition domain. It also briefly discusses the importance of CNN in this task. Next, section 1.2 states the problem statement of this project. The work of different researchers has been reviewed shortly in the referred section. The study's aim and objectives are provided in section 1.3. Section 1.4 raised the research questions from this current study while section 1.5

outlines the scope of the present study. Subsequently, the significance is explained in section 1.6.

Chapter 2 describes the required theoretical context and outlines the shortcomings presented in the previous chapter through a review of the existing technology in the masked and unmasked face recognition domain. The application of Convolutional Neural Network architecture in the computer vision domain is presented in section 2.2. Subsequently, the Attention mechanism is introduced in section 2.3. Various approaches by researchers in instilling attention to the CNN model are discussed in the same. Section 2.4 presents the Siamese network architecture to train the CNN model. The task of face recognition and verification are described and different approach to solving these tasks is presented in section 2.5. Section 2.6 elaborates upon the masked face challenge and discusses work done in the domains of face mask detection, masked face recognition, and verification. Sections 2.7 and 2.8 present the available open-source datasets and benchmarks for unmasked and masked face recognition tasks. Discussion is presented in section 2.9 and the chapter is concluded in 2.10.

Chapter 3 discusses the selection of the dataset and the proposed network architecture for the task at hand. The rationale behind the selection of the dataset and the pre-processing and class balancing steps involved are given in section 3.2 of this chapter. Section 3.3 presents the detailed design of the network building block and an overview of the proposed model. It also depicts the signal dimension at different blocks of the network. Training and the evaluation metric are also presented in this section. The last section summarizes the entire chapter in brief. Chapter 4 discusses the analysis and the design steps involved in the work. The actual data partitioning and preparation steps are elaborated in section 4.2. The subsequent section 4.3 details the model implementation, which discusses the data staging and model building in detail. Finally, various model settings and the pertinent hyperparameters are discussed. The last section succinctly summarizes the entire chapter.

Chapter 5 is about the obtained results and the model evaluation. Section 5.2 discusses various model training and validation loss and the corresponding interpretation. Section 5.3 describes the trained model performance on the test dataset. Subsequent section 5.4 checks the model performance on Grad-CAM and shows whether the model focuses on the relevant part while predicting. Later section 5.5 entails testing of the trained model on random scraped masked-unmasked face image pairs. The last section gives a brief of the entire chapter. Chapter 6

concludes and recommends based on the work and various experimentations. Section 6.2 focuses on the final discussion and the conclusion of the done work. Section 6.3 deliberates upon the contribution to the knowledge and finally the last section briefs the entire chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter first reviews the convolutional neural network and how the incorporation of the attention mechanism in the same impacts the performance of various tasks. Subsequently, it reviews the Siamese Network and various ways to train this architecture. Network performance under different loss functions has also been explained. Then the tasks of face recognition and verification have been described and explained how these are similar yet different from one another. Different works by researchers in the field of face recognition have been reviewed. Following the same, the concept of occlusion and different challenges for the occluded face are explained. Work towards face mask detection is reviewed. Further, masked face recognition is explained and elaborated on how occlusion due to objects like face masks can affect the existing model performance. Afterward, different techniques used by the research community to tackle the issue of masks in face recognition tasks have been reviewed. Different datasets open-sourced by the researchers have been reviewed and explained. After the same, the discussion has been put on. In the end, the entire literature review work has been summarized.

2.2 CNN for Computer Vision

Neural networks are the foundation of deep learning. With the help of these simple yet powerful networks, computers these days can generate any function. However complex the relation may be, neural networks can map them. Due to such capability of neural networks, it is being used widely in many industrial applications.

Ever since the discovery of digital cameras, it has become relatively easy and cheap to click images and videos. Unlike tabular data which are structured in nature, these kinds of data are called unstructured data because they don't have any specific structure in which they are stored in the databases. The onset of the social network era has especially given rise to people capturing and sharing digital images and videos online. Due to the same, the generation of these kinds of unstructured data has become exponential. Analysis of this kind of data for profitability has become a new trend now. This is altogether a new subdomain of deep learning called Computer vision. It is a field in which computers are used to interpret or analyze images and videos. Automation of these tasks can help save time as well as money as analysis of

unstructured data is a cumbersome job. Due to its usefulness in other domains, researchers have explored the usage of neural networks in the field of Computer Vision. However, simple neural networks have their limitations when it comes to the analysis of unstructured data like videos, audio, images, etc. Though they can be used on unstructured datasets but not with optimum efficiency. The network becomes bulky and hence computationally expensive when simple neural networks are used for these unstructured data.

An experiment by Hubel and Wiesel revealed that a cat's visual cortices, which contain neurons, individually give response to tiny regions of the visual field. The same knowledge was used by the research community to come up with a modified architecture called Convolutional Neural Network (CNN). The problem of the bulky network was solved by the discovery of CNN architecture. When a simple neural network and CNN are built to analyze the same image, CNN needs a smaller number of parameters as compared to a simple neural network.

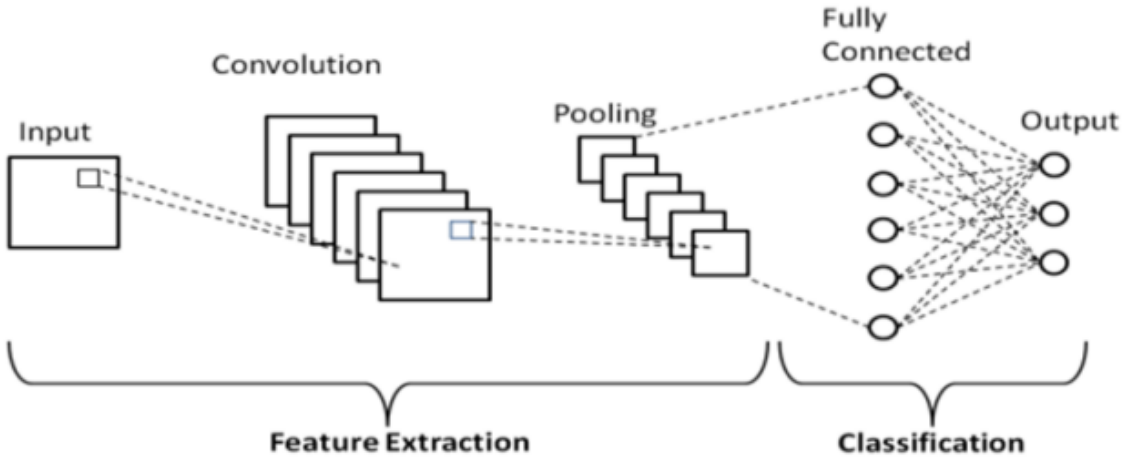


Figure 2.1 Schematic structure of CNN from (Suresh et al., 2021)

Figure 2.1 above depicts the architecture of CNN. The success of CNN in this domain comes from its ability to detect higher-end correlations/patterns amongst the neighboring features/pixels in the image. The structure of a CNN layer is such that with the use of various filters/kernels, it tries to get multi-level information from the same patch of input image/information and tries to get out the specific pattern in the same. These filters/kernels are trainable and hence they can be trained for different tasks. By stacking these convolution layers, CNN can produce the output that represents a hierarchical pattern (Hu et al., 2017).

2.3 Attention Mechanism in CNN

While doing any task, a human being tends to focus on certain things more than others. It is very natural for a human to focus on handwriting while reading a page rather than horizontal or vertical lines drawn on it. The same does not come naturally to these machine learning models. Bringing such sense to a model can aid in reducing the difficulty level of the task at hand. Attention can be defined as a way of making the machine focus more on certain features of the input signal which are most informative, as compared to the rest. The attention mechanism is simple yet very powerful and hence being utilized in many tasks like image captioning (You et al., 2016), image processing (Zheng et al., 2017), natural language understanding (Zhou et al., 2016), speech recognition (Chorowski et al., 2015), etc.

Researchers have proposed various architectures to instill attention mechanisms in CNN. (Wang et al., 2017) have proposed a model architecture that contains attention modules that generate attention-aware features. They achieved the same by using a bottom-up top-down feed-forward structure in those modules. Utilizing this network, they were able to achieve state-of-the-art accuracy on CIFAR-100 and ImageNet datasets. In addition to that, they claimed that their residual attention learning allows training of very deep networks, and hence complex detection/ classification tasks can be solved with the same.

One more excellent work on this is Squeeze and Excitation network proposed by (Hu et al., 2017). Here the input features are passed through squeeze operation under which different feature map values are summed across their spatial dimensions. These features are then passed through a gating mechanism to produce features that are applied to input feature maps. The various operations are depicted in Figure 2.2. These features give more attention to important feature maps than unimportant ones.

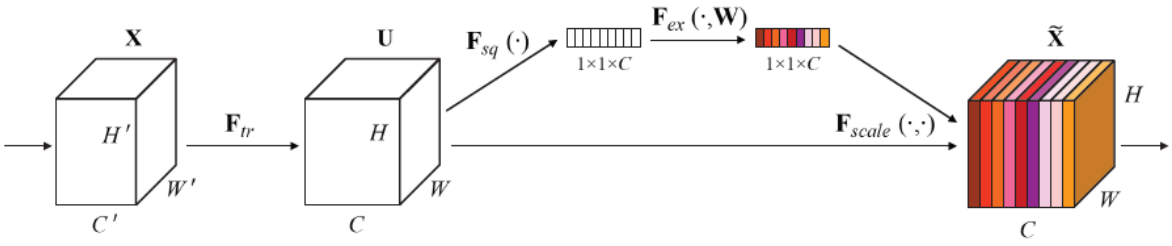


Figure 2.2 Squeeze and Excitation layer proposed by (Hu et al., 2017)

The model based on this mechanism has produced state-of-the-art (SOTA) performance in ILSVRC 2017 classification challenge. The extension of the above work is Convolutional Block Attention Module (CBAM) proposed by (Woo et al., 2018). Figure 2.3 below describes the two additional intermediate operations performed to bring attention mechanism in convolution operation. The channel attention module and Spatial attention module are stacked one after another to have the model focus more on important features of the input signal.

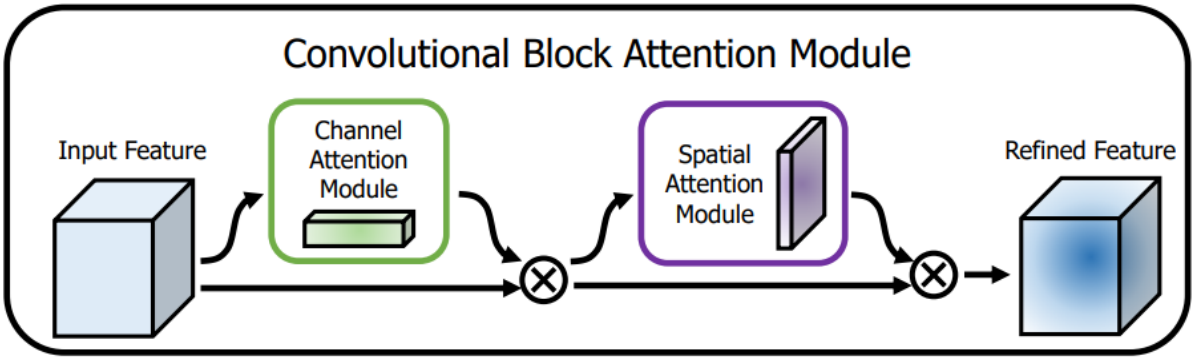


Figure 2.3 CBAM overview from (Woo et al., 2018)

Here unlike (Hu et al., 2017), (Woo et al., 2018) not only incorporated the spatial attention module but also the channel attention module. They proposed MaxPool and AvgPool for both channel and spatial attention modules and produced SOTA results on tasks like Image Classification and object detection.

(Fu et al., 2017) proposed recurrent attention convolution neural network for image recognition when the discriminative parts are too subtle. Their proposed model learns discriminative regions and the corresponding features, without the need for annotations. Their experiments on fine-grained tasks gave them SOTA accuracy, without the usage of much of the computation resources.

2.4 Training with Siamese Architecture

In the image classification task, a generally sufficient number of examples per class is available for training the corresponding deep learning model. However, Face recognition is a task where all the test images are never available at the time of training. Hence, to build something generalizable enough to work on unknown faces, it is prudent to come up with something which does not need the context of test images during training. (Chopra et al., 2005) proposed Siamese

network architecture, which maps higher dimensional input images to the corresponding lower dimensional embedding, while preserving the semantic distance among them. Figure 2.4 below depicts the training mechanism of CNN under the Siamese network architecture arrangement.

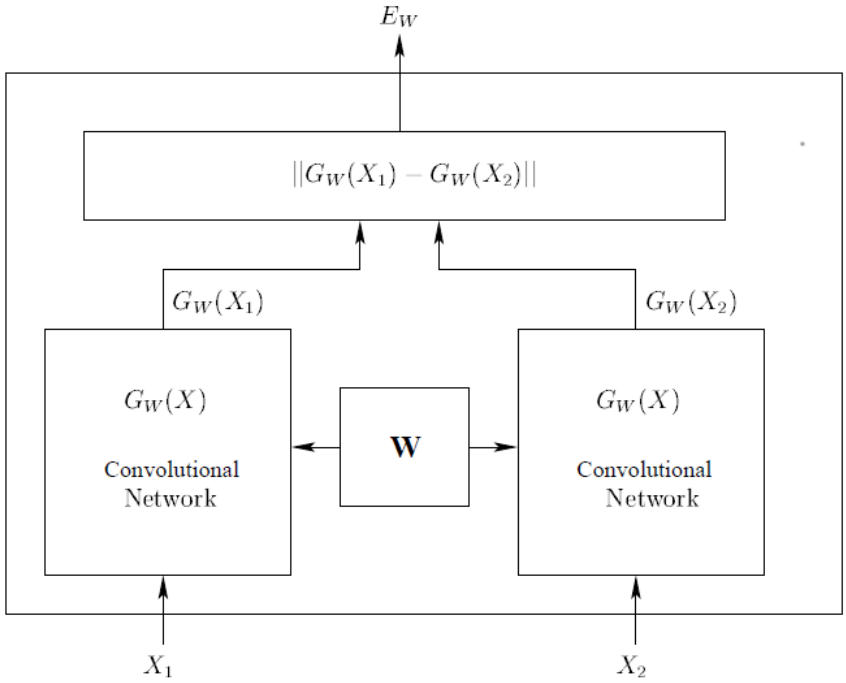


Figure 2.4 Siamese network architecture proposed by (Chopra et al., 2005)

They used a convolutional neural network to come up with this mapping function. The beauty of CNN is such that it is invariant to the distortions caused by variation in the pose. They built a face verification model based on this, which produces vectors that are closers for face images of the same subject and farther for face images of different subjects. Like the Siamese network, the other similar network architecture is proposed by (Hoffer and Ailon, 2015) and they called it Triplet Network. Unlike in Siamese Network with two tied CNNs, Triple Network uses three networks with tied weights. The advantage here is that it does not need a threshold for decision-making on whether the images belong to the same class or not.

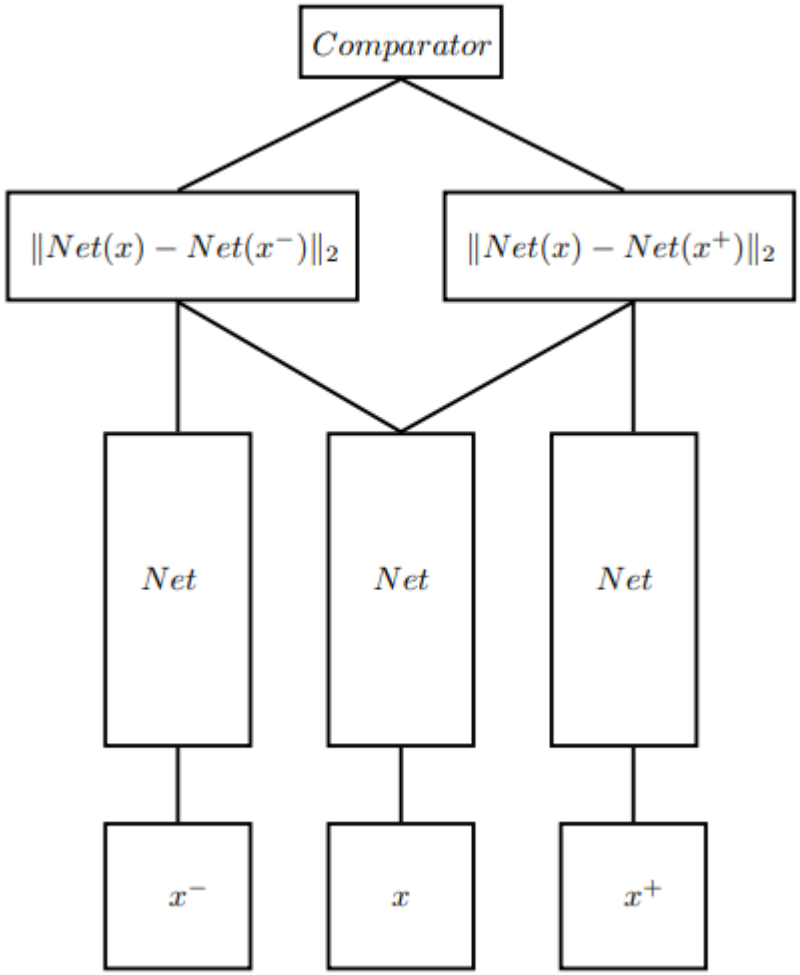


Figure 2.5 Triplet Network from (Hoffer and Ailon, 2015)

Figure 2.5 above described the CNN training scheme proposed by (Hoffer and Ailon, 2015). Here, the need for a threshold is dropped since the network is comparing the distance of the genuine pair with the impostor pair directly and making the decision.

Siamese network is a technique of training models to effectively, and efficiently extract features and track them. They are also useful for task line metric learning (Kumar BG et al., 2016) and few-shot learning (Koch et al., 2015). (Koch et al., 2015) got state of the art results on Omniglot data set – a dataset of a different class of letters. They even argued that Siamese Network architecture can help solve one-shot learning problems in other domains as well. Siamese network architecture is used for image matching tasks by (Melekhov et al., 2016). Employing the Siamese network for training the pre-trained CNN model trained on a similar dataset, they

could get good generalization on the different unseen landmark images. They argued that contrastive loss is a better choice for the image matching task.

Different loss functions have been proposed by researchers to train these network architectures. The aim behind all of them is to increase the inter-class distance and decrease the intraclass distance among the network-generated embeddings. One-shot learning is a task under which the model has to classify given only a single example per new class. (Ghojogh et al., 2020) proposed two different losses viz. Fisher Discriminant Triplet (FDT) and Fisher Discriminant Contrastive (FDC) and proved experimentally that these losses perform better than the traditional losses for Siamese Network training.

2.5 Face Verification and Recognition

Just like image classification, the task of face verification and recognition is not new to the research community. It is a kind of software that analyses and confirms the identity of a person. The application of this technology is tremendous. They are being used these days for non-trivial tasks like authentication, attendance systems, phone unlocking, finding the subject of interest from surveillance clips, etc. Many researchers have proposed various methods to achieve state-of-the-art metric values for this task.

As per the author's knowledge, probably the first breakthrough occurred with DeepFace (Taigman et al., 2014) model. They derived face representation from modeled 3D faces and were able to achieve close to 97.35% accuracy on the Labeled Faces in the Wild (LFW) dataset. (Schroff et al., 2015) used the online triple mining method to directly optimize the generated embeddings out of the network and achieved SOTA accuracy on LFW and YouTube Faces DB. Baidu (Liu et al., 2015) proposed a deep CNN-based model, employing metric learning and achieved pairwise verification accuracy close to 99.77% on the LFW dataset. They argued that their proposed model could achieve this feat because of the usage of features from different patches of the image. (Parkhi et al., 2015) employed their custom network viz. face CNN and trained the same with triplet loss. They claimed to achieve state-of-the-art results on LFW and YTF datasets. In addition to experimenting with the different network architectures, researchers have also explored various loss functions and proposed losses viz. Marginal Loss (Deng et al., 2017), Cosface (Wang et al., 2018), and Arcface (Deng et al., 2019), etc., and were able to achieve SOTA accuracy on the LFW dataset.

2.6 Challenges with Mask-Occluded Faces

These days, however, face recognition has not stayed that easy job. Various challenges come along when one is building a face recognition model. Probably the biggest of all is the occlusion caused by different objects. Obstruction caused due to foreign object(s) kept against the area of interest is called Occlusion. Some of them are fashion objects like glasses, scarfs, hats, etc. while others are protective gear, to protect from various hazards. Facial occlusion could be caused by the mask is one of the most challenging hindrances for face recognition tasks as it hides most of the key discriminative features of a human face. Mask has become a common cause of occlusion as the practice of wearing the same while in public places has become a common trend these days due to rising air pollution, the spread of contagious diseases, the outbreak of viruses, etc. Broadly there are two main tasks namely face mask detection and masked face recognition.

2.6.1 Face Mask Detection

Mask detection is a non-trivial task under which the model checks whether the subject is wearing a mask or not. The importance of this task is apparent given the current situation of a pandemic. As per WHO, wearing a mask is the best means to prevent the spread of the deadly COVID-19 virus. Governments all around the world have made it mandatory for their citizens to wear a mask while in public places. Automating the task of identifying those who are evading such a law is important. Various models have been proposed by different researchers to solve this problem and most of them have shown good accuracy. The solution provided by (Sanjaya and Rakhmawan, 2020) entails the use of MobileNetV2 to detect whether the subject is wearing a mask or not. Their model claimed to give an accuracy of around 96.85% on the face mask detection task.

2.6.2 Masked-face Verification and Recognition

Masked face verification and recognition are relatively different and challenging tasks as compared to face mask detection. Here the aim is to classify the identity of the subject while the mask is occluding the face. A Block diagram differentiating the masked face verification and recognition tasks is presented in Figure 2.6. Recognizing the face of the subject while having the mask on is challenging since the key features of the face get obstructed. General intuition says that while identifying the subject, attention should be on the uncovered region of the face rather than the masked region. The face verification task is one to one mapping where

a threshold is needed to check whether the given pair of images belong to the same subject while face recognition is many to one mapping where a subject and a database of different subjects are available and the task is to declare the identity of a person based on the one which is semantically closest.

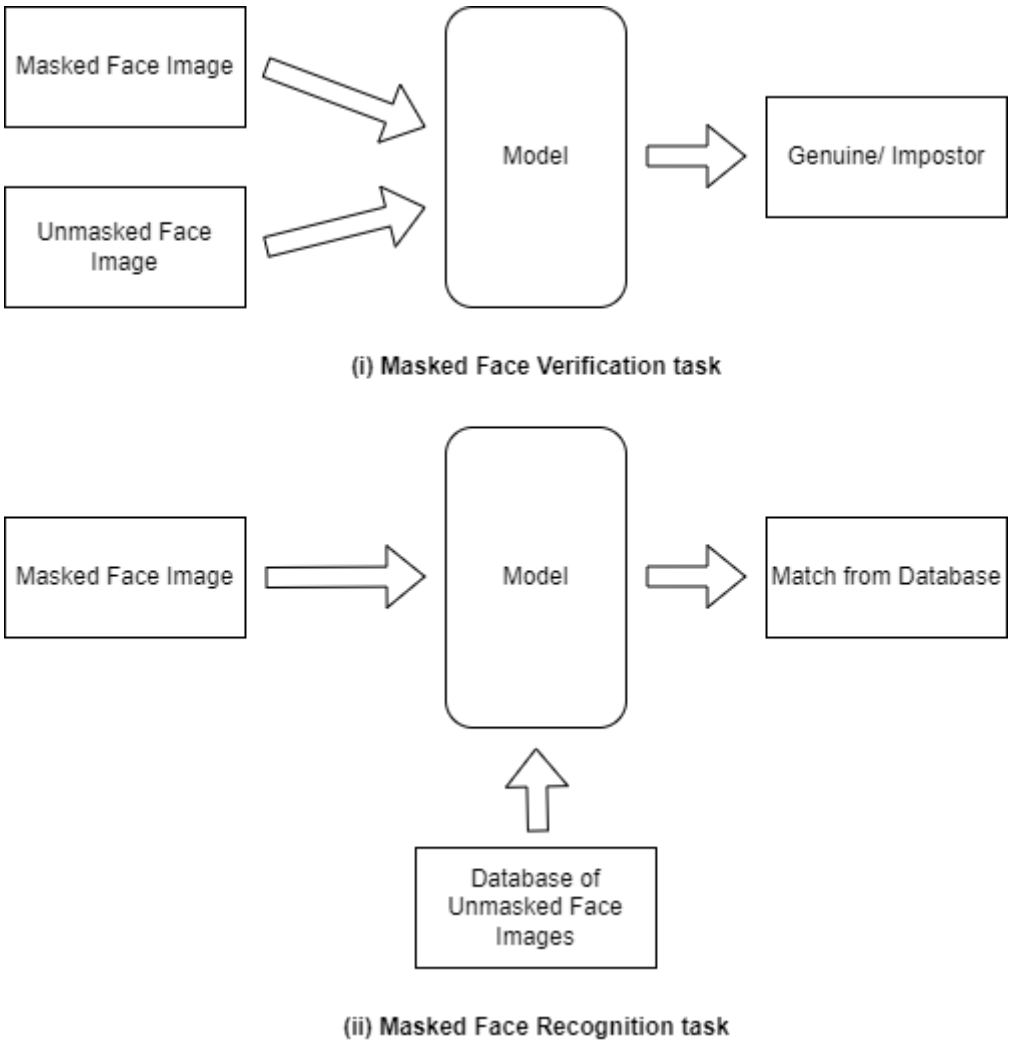


Figure 2.6 Overview of masked face verification and recognition tasks

2.6.3 Effect of Mask and Way Around

When it comes to face recognition, the mouth, nose, and eyes are the most prominent features. But when the subject is wearing a mask, mostly mouth and nose regions are covered and hence occluded. So, only one majorly discriminating feature is left i.e., the eyes. Now while performing any task, human beings instinctively focus on the most important part and provide less attention to the less useful parts. However, for a computer algorithm, it is naturally tough to make such decisions. This challenge is not new to the research community and many

proposals have been made to solve this arduous challenge. One way is to recover/ unmask the masked region of the face using generative models while the other is discarding the occluded part altogether and using the remaining part of the face to identify the person.

2.6.3.1 Recovery of Masked-part with Generative Models

(Ud Din et al., 2020), (Li et al., 2020) proposed a Generative Adversarial Network (GAN) based method to recover the masked portion of the face and produced a qualitatively as well as quantitatively well-performing model. (Ud Din et al., 2020) divided the task into two main sub-tasks. In the first one, they binarize the masked region from the rest of the uncovered area of the face. In the second stage, they removed the masked region and synthesized the occluded region while maintaining face structure and details. Employing the above method, they were able to get SOTA quality results as compared with other image editing methods. (Li et al., 2020) used distillation module - which uses GAN – that takes the general face recognition model as an instructor and transfers its learning to train the student model to complete the occluded part of the image. Their experiments employing this technique have shown promising results. But these GAN-based models are computationally heavy and not that effective at recovering faces for the task of face recognition.

2.6.3.2 Discarding the Mask-occluded part of the Face

Recovering the masked image is not that useful since the downstream task of face recognition needs accurate recovery without which the model might misidentify the subject. If not done right, this could lead to more trouble than aid. The alternative is to discard the mask-occluded region and use the uncovered region to train the recognition model. (Hariri, 2021) discarded masked area and used the non-occluded region to fine-tune the pre-trained CNN model.

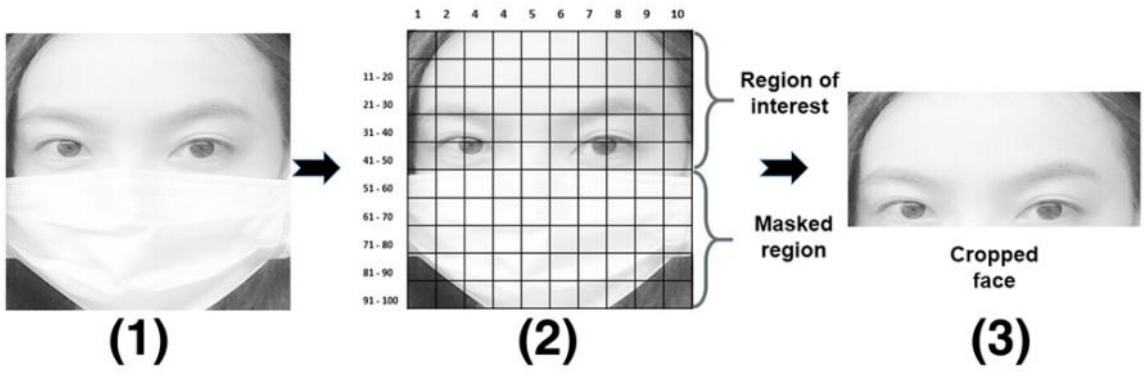


Figure 2.7 Region of interest selection filter employed by (Hariri, 2021)

Figure 2.7 above depicts the different stages a face image goes through before getting fed into the model employed by (Hariri, 2021). Upon locating the face region, the filter divides the image into 100 pixel blocks out of which only the top 50 are used while the rest are simply cropped out. The features extracted from the deep CNN model were used to achieve high classification accuracy on the Real-World Masked Face Dataset. The features extracted from the unmasked portion of the image are more useful and hence more weightage should be given to them as compared to masked region features.

(Yi et al., 2014) proposed latent part detection to locate those particular regions of the face which are invariant to wearing a mask. One model can identify those areas, those regions were used to further extract features that are discriminative enough to be used in the face recognition model. (Song et al., 2019) used the pairwise differential Siamese network to eliminate the corrupt portion of the face and utilized the rest for face recognition. (Li et al., 2021) adopted a mix of discarding and attention mechanism to achieve high metric value in the task of masked face recognition.

2.7 Unmasked-face Datasets

Ever since the onset of interest of the research community in solving the human face recognition task, the Labelled Faces in the Wild (LFW) dataset has played the role of a benchmark. Introduced by (Huang et al., 2008), it is one of the largest human face datasets with varied poses, focus, age, race, gender, quality, etc. It has a total of 13,233 different images of 5749 different subjects. Some of the images however contain multiple faces in them but the main subject of interest in those pictures is always at the center of it.

(Cao et al., 2018)introduced face image dataset viz. VGGFace2 contains around 3.3 million face images of 9131 different subjects. These images are mostly downloaded from Google Search and contain celebrities. Because of the same, this collection has a wide range of poses, ages, and lighting conditions, Figure 2.8 below depicts the same. These features of this dataset make it more challenging for face recognition tasks. It has a good balance of two genders with around 59% male face images. The author of the dataset has annotated bounding box around the face along with fiducial key points.

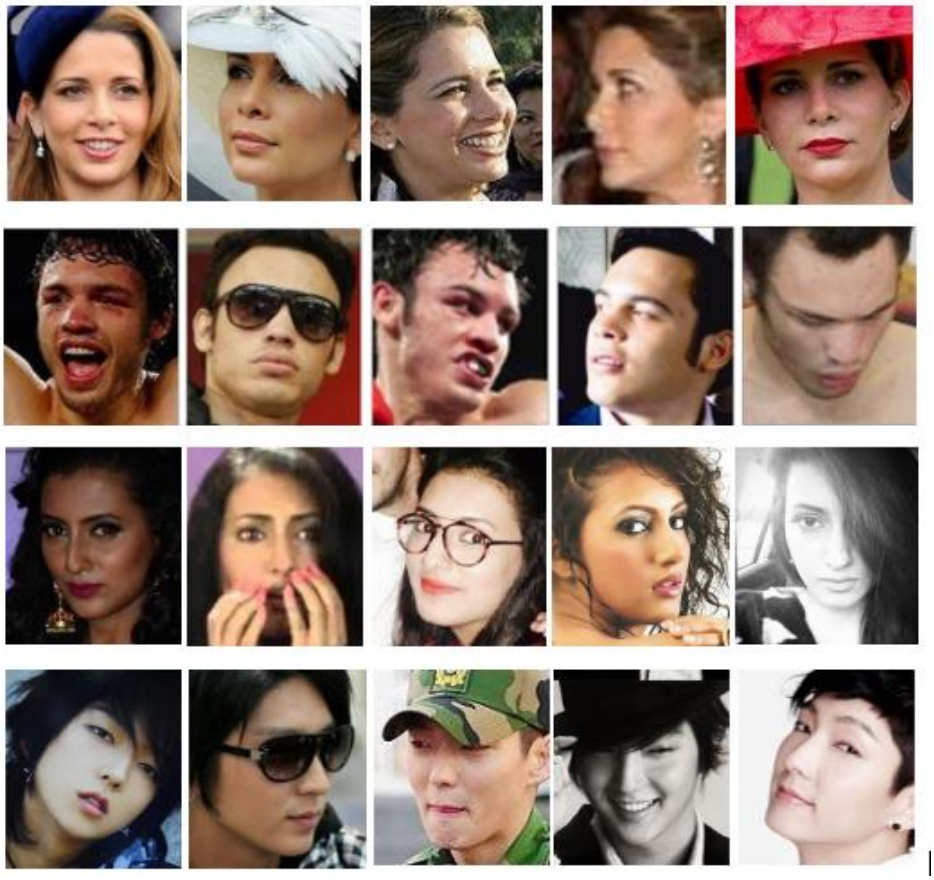


Figure 2.8 Sample of face images from (Cao et al., 2018)

2.8 Masked-face Datasets

Various masked face datasets have been proposed by the research community for the training of these models. Two of the biggest are the Real-world Masked Face Recognition Dataset (RMFRD) and Indian Masked Face dataset.

2.8.1 Real-world Masked Face Recognition Dataset

It is proposed to use the Real-world masked face recognition dataset (RMFRD) for the proposed project. It contains different masked as well as unmasked face images of different subjects. Those face images have varied expressions, and poses, which can help the model generalize better. The originally published dataset with the title Real-World Masked Face Dataset (RMFD) has three subsets viz. Real-world masked face recognition dataset (RMFRD), Simulated masked face recognition datasets (SMFRD), and Real-world masked face verification dataset (RMFVR). The real-world masked face recognition dataset (RMFRD) is our focus of interest. A sample of the same is presented in Figure 2.9. It contains web-crawled colored images of about 525 different subjects. There is a total of around 5,000 masked and 90,000 unmasked

faces for those 525 subjects. The ethnicity of all the subjects in this dataset is Asian, however, that should not restrict our model from learning mask face verification and the recognition tasks.

Figure 2.9 Sample masked- unmasked image pairs from the RMFRD dataset

2.8.2 Indian Masked Faces in the Wild Dataset

One more such dataset was made publicly available by (Mishra et al., 2021b) with the title Indian Masked Faces in the Wild (IMFW). They rationalize the importance of their contribution based on the argument that the existing available masked face dataset is not culturally varied enough. Further, in their proposed dataset, faces are not only wearing standard masks but also items like stoles, handkerchiefs, colored towels, etc. This could further challenge the existing face mask recognition as well as masked face verification models.

Figure 2.10 Sample of IMFW dataset from (Mishra et al., 2021b)

Figure 2.10 shows a sample of the Indian Masked Faces in the Wild (IMFW) dataset. It has a total of 630 masked and 744 unmasked images of a total of 200 different subjects. They managed to gather this dataset from the internet and crowdsourcing. Their experiments show that masked face recognition is a challenging task, especially when the data collected is not in a specific order and constrain.

2.9 Discussion

Face recognition is the task of recognizing the input face image and mapping the same with one of the faces from the database. The direct comparison of the face images is not possible due to variation in poses, lighting conditions, picture quality, size, etc. These factors change the relative pixel values to a very high extent, which makes the direct comparison extremely difficult. The task at hand is a two-class classification problem where it is to be declared whether the input face image pair is genuine or an impostor. The proposed model aims to differentiate any input face image pair regardless of whether the same was part of the training dataset or not. For the classic way of training a classification model, the test images should be similar to the training images. Further for the classic way, there have to be as many numbers of different classes in the last layers as the total number of different subjects, which is practically not possible. Hence, the traditional way of training the model for classification tasks is not suitable for our problem.

In a bid to come up with a solution for this problem, the metric learning-based approach is recommended by the research community where the CNN model is used to extract the important features of face images. The features may be the shape of a face, size of the nose, eye color, or maybe some abstract version of their combination. The extracted features certainly depend upon the network architecture of the core mode used to derive those features from the face images. The deeper the core CNN model goes, the more interlinks get generated by it. These features are then compared directly on distance measures like L_1, L_2, cosine distance, etc., and similarity or dissimilarity among different faces is declared.

The major issue with the masked face images is that the core CNN model does not get to see much of the discriminating regions like face shape, mouth, and nose. Instead, it gets masked along with the uncovered region of the face as the input. Here the mask is the non-discriminating feature and the model should know that hence somehow learn to ignore that input region while deriving the final set of features of the input face image. For a vanilla CNN model, it is a

challenging task since it doesn't have dedicated architecture or mechanism to learn such functionality. Generative models like (Ud Din et al., 2020) and (Li et al., 2020) have their limitations when it comes to recovering the occluded portion of the face image for recognition tasks. Though their recovery performance might be good enough for other tasks, face recognition needs the input features which are too delicate to regenerate with these techniques.

The attention mechanism has become very popular in the research community due to its effectiveness in solving various tasks with ease. Over the past few years, this particular subdomain of the field has grown rapidly. To solve the problem at hand, the attention mechanism incorporated in the CNN model could be promising. There are plenty of different attention modalities available for solving different tasks. The channel and spatial attention proposed by (Woo et al., 2018) and (Hu et al., 2017) seem the best fit for our requirement. It gives different weightage to the different spatial components as well as channels to bring focus to the relevant part of the image rather than the entire input image.

2.10 Summary

In summary, CNN has transformed the computer vision domain and greatly contributed to reducing the number of parameters of deep learning models. Various kinds of attention mechanisms proposed by the research community have further eased the difficulty in focusing on the relevant part of the input image and ignoring the irrelevant one. Subsequently, the task of face recognition and verification are discussed and rationalized how the Siamese Network architecture helped tackle the challenge of novel faces during testing. It is concluded that the metric learning-based approach is more effective, and efficient compared to the standard classification-based approach in solving the problem of face recognition. The occlusion created by masks and similar objects could cause the existing face recognition models to fail miserably. The way around for the masked faces is cropping the occluded region or regenerating the masked region by a network like GAN. Later on, various available masked and unmasked face datasets and the corresponding features are discussed. Finally, the discussion was held on the broad picture in the domain of face recognition and how attention could solve the problem of occlusion due to face masks.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter starts with the selection of the masked-unmasked face dataset for the proposed work. Different features of the chosen dataset and the rationale behind the selection of this particular dataset are explained. Going forward, the data pre-processing and transformations are brought into the discussion. Subsequently, the class balancing process adopted for feeding the face images into the model for training and validation is explained. Then the proposed method for the work is elaborated in detail. Here, the Squeeze and Excitation mechanism is explained mathematically and then the proposed network architecture is elaborated in detail along with all the network input and output dimensions. Later, the choice of loss function for training the model is rationalized. Afterward, the evaluation matric for checking the performance of the trained model is explained. Lastly, the chapter ends with a summary of all the included sections in this chapter. The overall methodology flowchart for the proposed work is depicted in Figure 3.1 below.

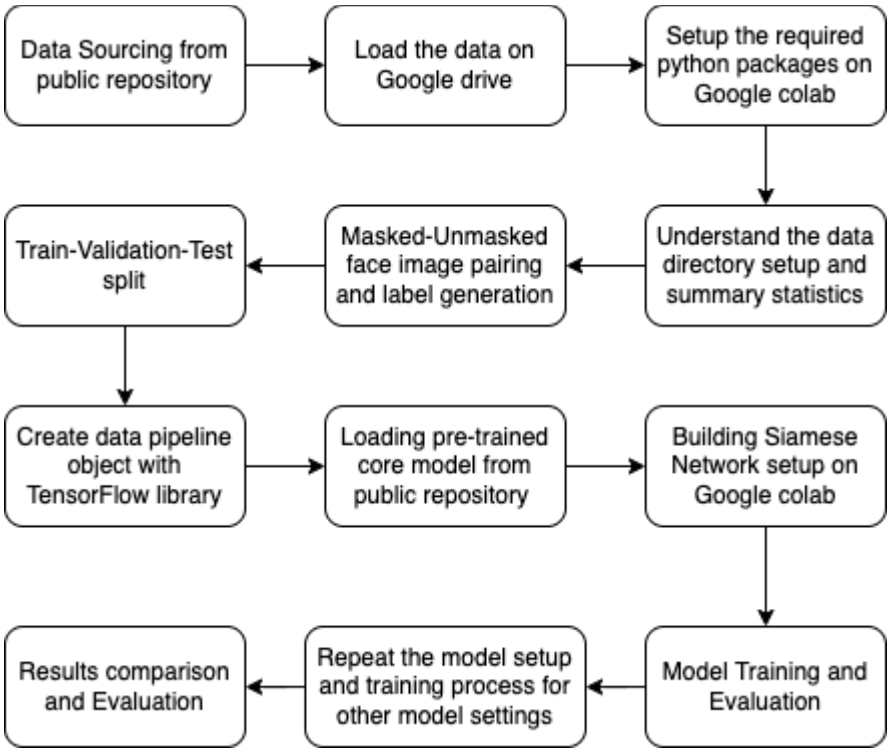


Figure 3.1 Methodology workflow diagram depicting the various actions taken to achieve the objective of study

3.2 Choice of Dataset and Various Operations

The task at hand needs a dataset large enough to help train a deep attention-based CNN model by metric learning approach. The following sections describe the rationale behind the selection of a particular dataset and the various data pre-processing steps involved before training the model.

3.2.1 Masked-Unmasked face Dataset

For this study, it is proposed to use the RMFRD dataset introduced by (Wang et al., 2020). As per the author, it is the largest open-sourced compilation of masked-unmasked face image dataset available to the research community. A summary of the data is depicted in Table 3.1 below.

Table 3.1 Summary statistics of the RMFRD dataset

	Masked faces folder	*Unmasked faces folder*
No. of subjects	525	460
Total No. of face images	5002	90468

The contributor of the dataset claims that it is the world's largest dataset with masked faces of humans. The same is the main reason behind the selection of this dataset for the proposed work. The aim behind the creation of this dataset is to aid the research community in building intelligence management systems and products that help in managing public safety in the events like COVID-19. The dataset contains web crawled images of 525 different subjects. Images are accurately cropped such that there is only one face per image and the same occupies the central pixels of each image. All the images are labeled properly without error and hence no additional data cleaning is required. All the subjects are of Asian ethnicity. Though such commonality might cause bias in the trained model however researching in that direction is reserved for future work.

3.2.2 Data Pre-processing and Transformation

The face images in the RMFRD dataset are arranged in two main folders viz. masked and unmasked faces. Both the folders have sub-folders with different subject names and those folders have face images of the corresponding subjects. Each image is well cropped such that it only includes the face of the subject, so there is no need to crop the images. Images are colored

and have RGB channels. However, the shape of images is not common. So, it is pertinent to resize the images to a standard size to make them compatible with the input layer, before feeding the image pairs to the model.

3.2.3 Class Balancing

The task of our model is to take in a masked and unmasked image of the same or different subjects and identify whether they belong to the same subject or different. So, we need to feed image pairs into the model for training. The real-world masked face recognition dataset (RMFRD) contains a total of around 5,000 masked and 90,000 unmasked faces of 525 different subjects. The number of unmasked images is far more than the number of masked images. Moreover, there are only 525 different subjects that are key for data preparation for the model training procedure. It is proposed to sample 400 subjects randomly from the dataset and use their masked and unmasked images for the training and validation dataset. The masked-unmasked images of the rest of the 125 subjects are proposed to be used for test dataset preparation.

The number of masked and unmasked images is not the same for all the subjects. However, for all the subjects, the number of masked images is far less than the number of unmasked images. The proposed model's ultimate goal is to classify whether the fed masked-unmasked image pair belongs to the same subject or not. So, it is a two-class classification problem that the model is trying to solve. To prepare the train, validation, and test datasets, it is proposed to randomly sample an equal number of genuine pairs as impostor pairs to avoid class imbalance. Otherwise, our model might always predict the class as an impostor since we have a limited number of unique subjects while the number of images per subject is large.

3.3 Proposed Method

Transfer learning is utilizing the learnings from one task for another one. It is a resource-efficient and time-saving way of training large models. It should be especially used when the task at hand is large while the available resources are not sufficient enough to complete the same. Transfer learning is most effective when the existing learned model is trained for a similar task. Our task under the proposed work is to build a masked face recognition model. Hence it would be prudent if we use a model which is trained on the human faces dataset.

(Cao et al., 2018)trained Squeeze and Excitation CNN network architecture proposed by (Hu et al., 2017) over the VGGFace2 dataset. They trained the model on the subject identity classification task and then used the layer adjacent to the classification layer to extract the face embeddings of different images. They used these embeddings as a comparator to come up with a face verification model. The difference between their problem and our task at hand is that we have masked-unmasked faces to compare while they have unmasked-unmasked faces to compare and verify the identity.

3.3.1 Squeeze and Excitation

Squeeze and Excitation (SE) is proposed by (Hu et al., 2017) to give rise to important features and take away the focus from the irrelevant ones. They felt the need for interaction among different channels of the CNN layer so that network can enhance the focus on an important chunk of the input signal.

To make the model aware of the context of the entire image, the squeeze operation is performed on the output of the CNN layer. It takes the spatial global average of each feature map. For a given intervening feature map F with dimension $R^{C\times H\times W}$, the squeeze operation brings the dimension down to $R^{C\times 1\times 1}$. This reduced dimensional map is called a channel attention map, say M_c. The squeeze operation is mathematically described by Equation 3.1 below. Following the same, channel attention map M_c is passed through two layers of the neural network, which enable the model to learn the internal dependencies among different channels. The original work by (Hu et al., 2017) used these layers along with the ReLU activation to first reduce the dimensions and then matched the dimension of the output signal with that of the input signal. The operation is called excitation and the same is given by Equation 3.2.

$$M_c = \frac{1}{H * W}\sum_{i=1}^{H}\sum_{j=1}^{W} F(i,j) \qquad (3.1)$$

$$S = \sigma\left(W_2 * \left(\delta(W_1 * M_c)\right)\right) \qquad (3.2)$$

The final output of the SE block is obtained by channel-wise multiplying the S obtained after the excitation operation to the input feature map F. So, the SE block in the network architecture recalibrates the input features and passes the signal to the next layer. Given the similarity of our

task at hand and the model objective in (Cao et al., 2018), it is proposed to use the model trained by (Cao et al., 2018) as a base CNN model for the masked face recognition task.

3.3.2 Overall network architecture

The SE block employed for building the base CNN model is depicted in Figure 3.2. The incoming signal is first batch normalized and then fed to the GlobalAveragePooling layer, to perform the squeeze operation on the same. The above output is then fed to two layers of neural networks, which first reduce the dimensions of the signal and then match it to the dimensionality of the input signal. These layers perform the excitation operation on the incoming signal. Subsequently, the output of the excitation block is multiplied by the original incoming signal to give the final output from the Shift and Excitation Block.

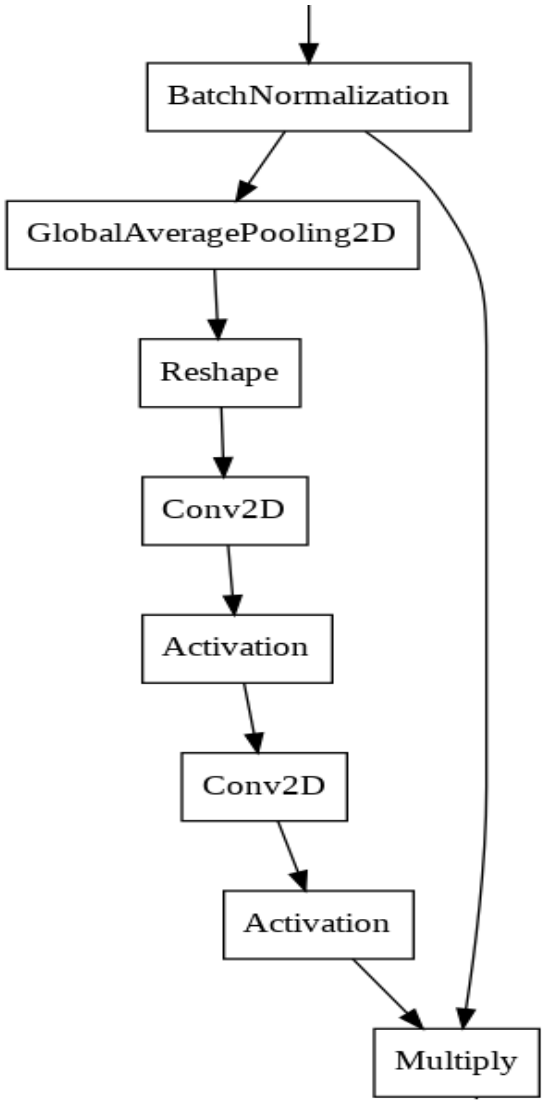


Figure 3.2 Squeeze and Excitation block of SENet50 from (Cao et al., 2018)

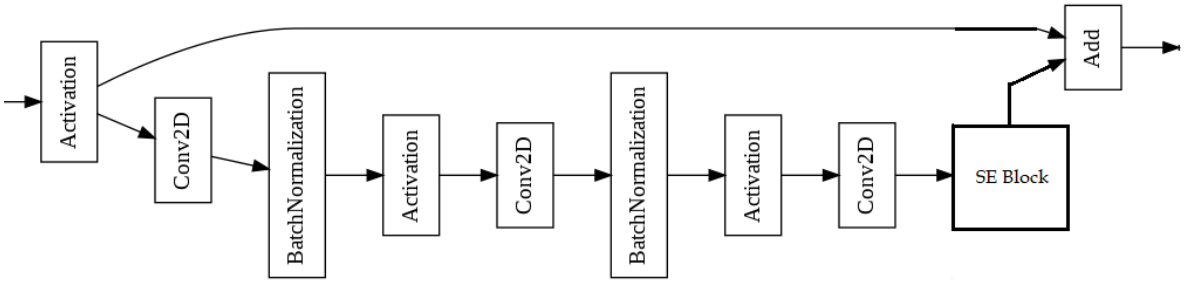


Figure 3.3 SE-ResNet block of the core CNN model

Figure 3.3 above depicts a SE-ResNet block of the SENet50 model proposed for this work. Here residual skip connection is used which is skipping the SE block in the core model. This gives the model the flexibility to utilize the SE block or skip it. The SENet50 is proposed to be the core model, which has a total 16 number of such blocks. The entire core model has a total of 287 different layers. The pre-trained weights are initiated from the model trained by (Cao et al., 2018) on the VGGFace2 dataset.

Table 3.2 The model blocks with corresponding input-output signal dimensions

Layer description	***Input dimension***	***Output dimension***
Input layers	(Batch, 224, 224, 3)	(Batch, 55, 55, 64)
SE-ResNet block 1	(Batch, 55, 55, 64)	(Batch, 55, 55, 256)
SE-ResNet block 2	(Batch, 55, 55, 256)	(Batch, 55, 55, 256)
SE-ResNet block 3	(Batch, 55, 55, 256)	(Batch, 55, 55, 256)
SE-ResNet block 4	(Batch, 55, 55, 256)	(Batch, 28, 28, 512)
SE-ResNet block 5	(Batch, 28, 28, 512)	(Batch, 28, 28, 512)
SE-ResNet block 6	(Batch, 28, 28, 512)	(Batch, 28, 28, 512)
SE-ResNet block 7	(Batch, 28, 28, 512)	(Batch, 28, 28, 512)
SE-ResNet block 8	(Batch, 28, 28, 512)	(Batch, 14, 14, 1024)
SE-ResNet block 9	(Batch, 14, 14, 1024)	(Batch, 14, 14, 1024)
SE-ResNet block 10	(Batch, 14, 14, 1024)	(Batch, 14, 14, 1024)
SE-ResNet block 11	(Batch, 14, 14, 1024)	(Batch, 14, 14, 1024)
SE-ResNet block 12	(Batch, 14, 14, 1024)	(Batch, 14, 14, 1024)
SE-ResNet block 13	(Batch, 14, 14, 1024)	(Batch, 14, 14, 1024)

SE-ResNet block 14	(Batch, 14, 14, 1024)	(Batch, 7, 7, 2048)
SE-ResNet block 15	(Batch, 7, 7, 2048)	(Batch, 7, 7, 2048)
SE-ResNet block 16	(Batch, 7, 7, 2048)	(Batch, 7, 7, 2048)
Core model output layer	(Batch, 7, 7, 2048)	(Batch, 2048)
Distance operator	(Batch, 2048)	(Batch, 1)

The proposed model is a Siamese network with SENet50 as the core model. As described in Figure 3.4 below, the overall model takes two inputs, one is the masked image and the other one is an unmasked image of either same or the different subjects. If the face images belong to the same subject, the input face image pair is called the genuine pair and if they are of different subjects, it is called the impostor pair. The face image pair is fed into the Siamese network that converts the higher dimensional input face images into lower-dimensional embeddings. The dimensionality of the signals at different layers is described in Table 3.2. The final embedding dimension output by the model is of size 2048, which is fed into the distance operator unit. Since one masked and one unmasked face image are fed into the model at a time, the model output two such face embeddings. These face embeddings are then normalized, elementwise subtracted, squared, and summed, to get the distance between them. The same is described in Equation 3.3 below.

$$O = Sum(\ (E_{norm}^{masked} - E_{norm}^{unmasked})^2\ ) \qquad (3.3)$$

where,

E_{norm}^{masked} = Normalized face embedding of a masked input image

$E_{norm}^{unmasked}$ = Normalized face embedding of an unmasked input image

Since the range of the normalized distance operation is not constrained, it is perceived that the output of distance operation shall be low for genuine masked-unmasked pairs while the same shall be relatively high for impostor masked-unmasker pairs.

3.3.3 Loss function and Model Training

The output of the model is not constrained; however, a high value indicates impostor pair while a low one signifies genuine face image pair. Given the nature of the output of the model, the loss function should be such that it updates the weights of the model to bring the output value

low for genuine face image pairs and high for impostor face image pairs. The proposed loss function for the network is described mathematically in Equation 3.4.

$$Batch\ Loss\ = \sum_{i=1}^{batch\ size} y_{true}^{i} * \left(y_{pred}^{i}\right)^2 + \left(1 - y_{true}^{i}\right) * \max\left(0, \left(A\ -\ y_{pred}^{i}\right)\right)^2 \qquad (3.4)$$

Here, the y_{true} is annotated in such a way that its value is 1 for genuine pairs while 0 for impostor pairs. Max function is used in the impostor pair part of the loss function which will derive the model to update weights to give the output as high for all the impostor face image pairs. Here, A is a positive integer whose choice can define the separation between the genuine and impostor class. The greater the value of A, the more rigorously the model's weights get updated in training steps.

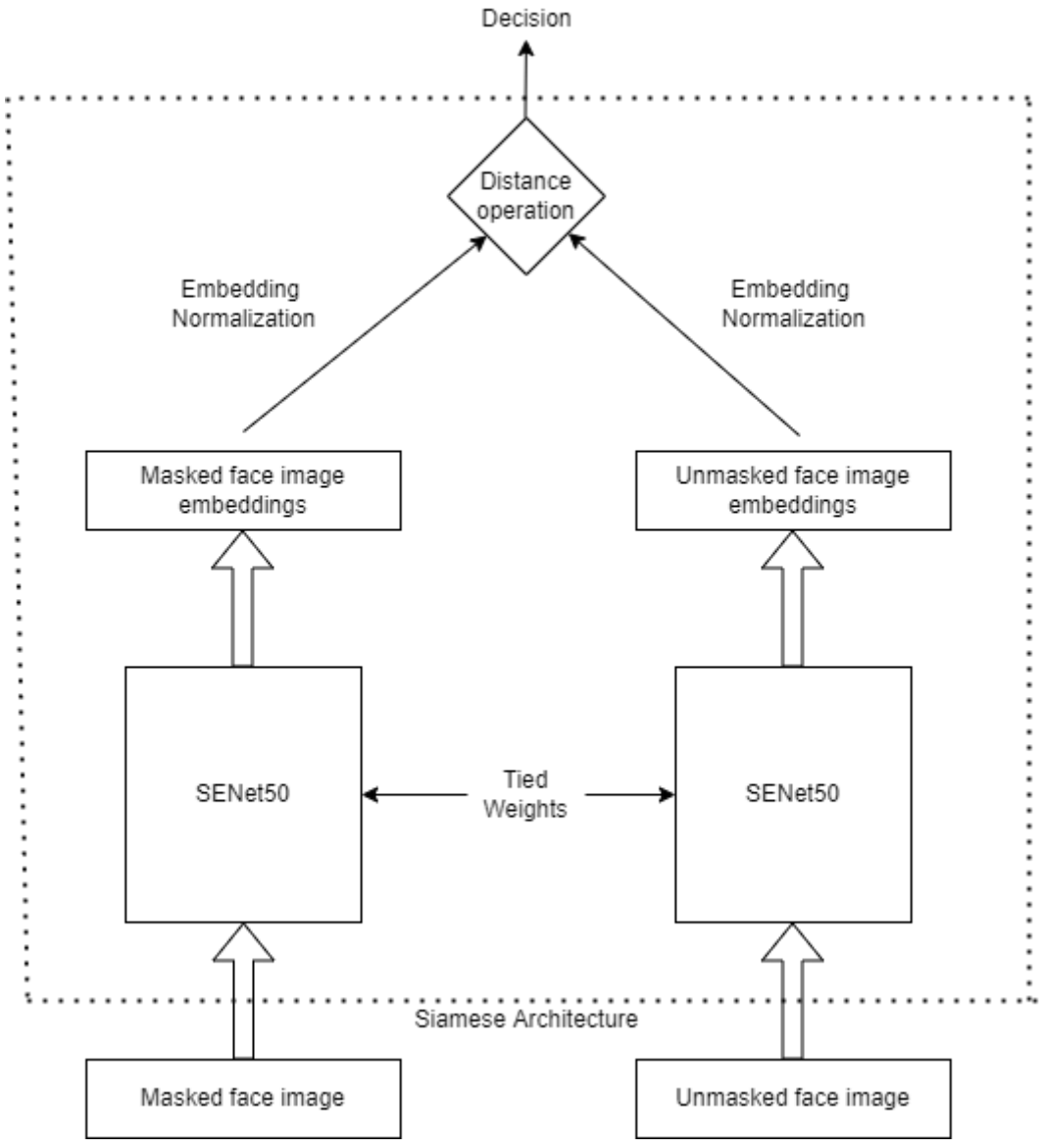


Figure 3.4 Siamese Architecture with SENet50 as the core network.

3.3.4 Evaluation Metric

To test the differentiation ability of the trained model between genuine and impostor pairs of input masked, unmasked face images, it is proposed to use Genuine Mean Distance (GMean) and Impostors Mean Distance (IMean), as used by (Neto et al., 2021). GMean is the mean L_2 / cosine distance among all the genuine pairs of face images while IMean is the mean L_2 / cosine distance among all the impostor pairs. These distances are described mathematically by Equations 3.5 and 3.6 below.

$$GMean = \left(\frac{1}{Count_{Genuine}}\right) \sum_{Distance_i \in Genuine} Distance_i \quad (3.5)$$

$$IMean = \left(\frac{1}{Count_{Impostor}}\right) \sum_{Distance_i \in Impostor} Distance_i \quad (3.6)$$

It is estimated that the model would be able to give face embeddings such that GMean is less than the IMean. The difference between these two mean values could be used to come up with a threshold value, which is used to determine whether the pair of images are genuine or impostors during testing.

3.4 Summary

In summary, the RMFRD dataset is selected for the proposed work being the largest available open-source dataset to the research community. The face images from the dataset do not need much pre-processing and are almost ready to be fed into the model. The squeeze and Excitation module is deemed an important part of the proposed network architecture, as it instills an attention mechanism in the core model. The core model is trained via a metric learning approach, employing Siamese network architecture and a loss function similar to the contrastive loss. Finally, the GMean and IMean are the proposed evaluation metric to test the learning capability of the model.

CHAPTER 4

ANALYSIS AND DESIGN

4.1 Introduction

This chapter first describes the count statistics of the chosen RMFD dataset and depicts various data partitioning and preparation steps along with the rationale behind adopting such steps. Thereafter, data staging is described succinctly. Further, steps taken in analyzing the model pre-trained on the VGGFace dataset are described in brief. Subsequently, how the pre-trained SENet50 model is used as the core for Siamese Network architecture is explained. Moreover, different model training settings are explained briefly. Model training steps and the hyperparameters of the model are explained at the end.

4.2 Data Partitioning and Preparation

The dataset chosen for training the proposed model is the Real World Masked Face Dataset (RMFD). Broadly it entails two folders viz. masked as well as unmasked face images. Both the folders have a different number of subfolders containing face images belonging to different subjects. However, there are only 442 common subjects. These are our subjects and are proposed to be used in the preparation of training, validation, and test datasets. The first 80% of these subjects i.e. 353 are used to build training and validation sets while the rest 20% i.e. 89 subjects are used to build test datasets. The data is proposed to be partitioned in this fashion so that there is no data leakage while the model training process and the learned model get tested without any bias.

Table 4.1 A sample of generated Train dataset and corresponding label scheme

Masked Image	***Unmasked Image***	***Generated Label***
 *Baike*	 *Baike*	1.0

 *Baike*	 *Denglun*	0.0
 *Denglun*	 *Denglun*	1.0
 *Denglun*	 *Baike*	0.0

The number of face images per subject is different for different subjects. So, it is proposed to limit the number of masked, unmasked face images picked for a particular subject for the preparation of training and validation datasets. A ceiling of 20 is set here to avoid the probable bias while model training. After picking the face images, these images are randomly shuffled and paired. While pairing the masked-unmasked images, the label is generated where 1 indicates the image pair belong to a common subject i.e. genuine pair while 0 indicates otherwise i.e. impostor pair. A sample of the paired Train dataset and the corresponding generated labeling scheme is shown in Table 4.1 above. The obtained dataset is partitioned into the 80-20 ratio where the 80% is named as the training dataset while the remaining 20% is named as the validation dataset. The prepared train dataset has a total of 42904 different image

pairs wherein half of them are genuine pairs while the other half of impostor pairs. Similar data distribution is true for validation images with a total of 10726 face image pairs.

Table 4.2 Summary of statistics for Train, Validation, and Test datasets

	Train	***Validation***	***Test***
No. of subjects	353		89
Total no. of unmasked images	7034		1766
Total no. of masked images	1343		455
Total no. of masked-unmasked pairs	42904	10726	18190
Total no. of Genuine masked-unmasked pairs	21452	5363	9095
Total no. of Impostor masked-unmasked pairs	21452	5363	9095

A similar strategy is adopted for the preparation of the test dataset with a total of 89 different subjects. These are different than the 353 subjects used for the preparation of training and validation datasets. The prepared test dataset has a total of 18190 masked-unmasked image pairs with 9095 i.e. 50% of total genuine pairs and the rest 50% as impostor pairs. A summary of statistics of the partitioned train, validation, and the test datasets are briefed in Table 4.2 above.

4.3 Model Implementation

The major steps of the Model Implementation section are data pre-processing and standardizing, as required by the VGG face dataset along with the model building with different training settings. The same is described in the following subsections.

4.3.1 Data Staging

The proposed core model for the work is a pre-trained SENet50 model trained on the VGGFace dataset, based on work by (Cao et al., 2018), available at (rcmalli, 2020). The original objective of this trained model is to classify the input face image belonging to different subjects. The input shape requirement of this pre-trained model is (224, 224, 3). Now since the Real World Masked Face Dataset (RMFD) has face images of different shapes and sizes, their images are first to be resized to the size required by the model. Further, as per the requirement, the input images are to be standardized before feeding the same to the core pre-trained model.

4.3.2 Model Building

The performance of the proposed model could truly be tested by comparing the same with the pre-trained model trained on the VGGFace dataset. Given the same, the model building section is broadly divided into two subsections. The first one covers the procedure adopted in getting the face embeddings out of the pre-trained SENet50 model, while the second subsection entails the different model training settings to further train the pre-trained SENet50 model with siamese network architecture and custom training loss.

4.3.2.1 Pre-trained SENet50 for prediction

The Pre-trained SENet50 which is proposed to be used as a core model for this work is trained on the VGGFace dataset, for the subject face classification task. The face images in the VGGFace dataset are unmasked and without much occlusion. Given the similarity of the task, it is prudent to use this model as a benchmark and check whether further training helps in improving the model performance of the task of masked face recognition. So, the last classification layer of this SENet50 model is discarded and the rest of the model with pre-trained weights is used for generating embeddings for different masked and unmasked images. The test dataset is used here so that performance can be directly compared.

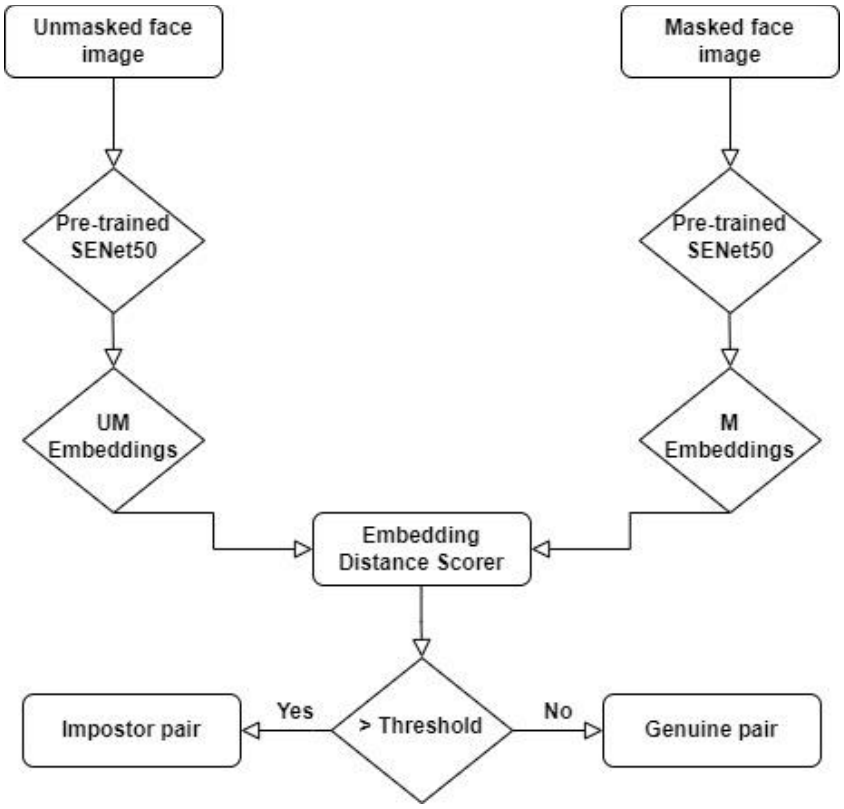


Figure 4.1 Flowchart for prediction with Pre-trained SENet50 model

The obtained embeddings are first normalized and then L2 distance among various masked-unmasked pairs was found. The Genuine masked-unmasked face image pairs are expected to have lower distances while the Impostor masked-unmasked face image pairs to have relatively higher distances. A broad flowchart indicating the prediction with the Pre-trained SENet50 model is depicted in Figure 4.1 above.

4.3.2.2 Building Siamese network using pre-trained SENet50

In this experimental setting, the weights of the pre-trained SENet50 mentioned above are used as a starting point for building the model. Both the cores of the Siamese Network architecture share the same set of weights at the onset of the model training. In the end, the math layer of the TensorFlow library is used to build a function that computes the L2 distance between the two obtained embeddings. The total number of paraments including trainable as well as non-trainable weights is 26,092,144. The bifurcation of these weights into trainable and non-trainable weights of course depends upon the number of layers of the core SENet50 model set to trainable. A custom loss function is created with the TensorFlow library to train the built model in Siamese architecture, intending to increase the distance among embeddings of Impostor masked-unmasked face image pairs while decreasing the distance among embeddings of Genuine masked-unmasked face image pairs.

Core SENet50 model has a total of 16 Squeeze and Excitation layers. To check the impact of the Squeeze and Excitation (SE) layer in achieving the objective, the number of trainable layers in the model is set to increase from 10 layers to the last 70 layers with an increasing interval of 10. Note here that the entire core model could be set to trainable but the objective here is to check the impact of the SE layer and not to build the best possible model to tackle the task at hand. Hence, by increasing the number of trainable layers, it is aimed to increase the presence of trainable SE layers in the model. With the last 10 trainable layers, the model has 1 trainable SE layer while with 70 trainable, it has 5 trainable SE layers. With a greater number of trainable SE layers, the model is expected to perform better for the masked-unmasked face recognition task.

For all the aforementioned experimental settings, model training is performed in the batch of 32. Since the core SENet50 model has multiple BatchNormalization layers, care is taken while fine-tuning the pre-training model under siamese network architecture, otherwise the same could result in loss of learning. Hence in line with Keras' documentation, BatchNormalization

is kept in inference mode during the model finetuning, by setting the training argument as false. TensorFlow's tf.data API is used to build the data pipeline and hence to train the model in batches, given the limitations of available resources. In the same pipeline, the required image resizing, and pre-processing steps are incorporated. The test dataset pipeline is also framed similarly and the predictions are made in batches, to avoid the excess usage of limited system memory.

4.4.3 Model hyperparameters

For any machine learning model, the selection of the right set of hyperparameters is very crucial. The model's success could be defined by adopting the right strategy to select these hyperparameters. The selection of hyperparameters and the rationale behind taking such steps are brought under.

The chosen SENet50 model is used for all the model training settings. The last layer of the same i.e. the embedding layer has the option to get either the maximum value or the average value at the output. Here, the average value is chosen since it is giving slightly better results as compared to its counterpart. The model has a total of 16 SE layers and the same is kept as it is, to avoid the loss of valuable learning by the pre-trained model. One important thing to keep in mind while fine-tuning the pre-trained model is the presence of the BatchNormalization layer. We followed the Keras documentation and kept all the BatchNormalization layers in inference mode, otherwise, the updates applied to the non-trainable weights may ruin what the model has learned from its previous training.

Table 4.3 Model hyperparameters

Hyperparameter	***Value***
Input shape	(224, 224, 3)
No. of Shift and Excitation layers	16
Core model top layer pooling	Average pooling
Parameter A in the loss function	5.0
Gradient descent learning rate	1e-5
Embedding dimensions	(1, 2048)
Embedding distance	Sum of squared differences

The chosen custom loss function has the flexibility of choosing the value of hyperparameter A such that the sharp weight update in the model may result in a steep difference in the distance score of Genuine and Impostor pairs. However, its value is chosen as 5.0, to main the stability while model training. Moreover, stochastic gradient descent with the learning rate of 1e-5 is chosen to aid the model to reinforce its learning slowly and smoothly. The consolidated list of hyperparameters and the corresponding choice is described succinctly in Table 4.3 above.

4.4 Summary

This chapter presented the Analysis and Research Design phases, which depicted the broad view of the proposed work. Subsequently, we studied the statistics of the image data and described how the entire data is partitioned while taking care of various issues like data leakage, class imbalance, etc. Later, we defined the benchmark model and described the various experimental settings, to test the proposed hypothesis for the given task at hand. Finally, we described the setup of the model pipeline and discussed the model configurations and the relevant hyperparameters.

CHAPTER 5

RESULTS AND EVALUATION

5.1 Introduction

This chapter discusses the different results and evaluates the model performance with the set benchmark for this work and different model training settings. It starts with visualizing training and validation loss for the different number of epochs. Thereafter, models with different training settings are evaluated on the test dataset, and performances are compared with the benchmark and one another. Subsequently, the performance of the best model is evaluated with the use of Grad-CAM. Finally, the distances were predicted for the randomly chosen masked-unmasked face image pairs, scraped from the internet.

5.2 Model Training and Validation loss

The model training loss versus the number of epochs of training for different model settings during the training phase is depicted in Figure 5.1 below. The aim here is to visualize whether the model fine-tuning is resulting in value addition into the pre-trained model. It is apparent from the referred figure that the training loss is decreasing by setting the higher number of model layers as trainable and training the model for a greater number of epochs.

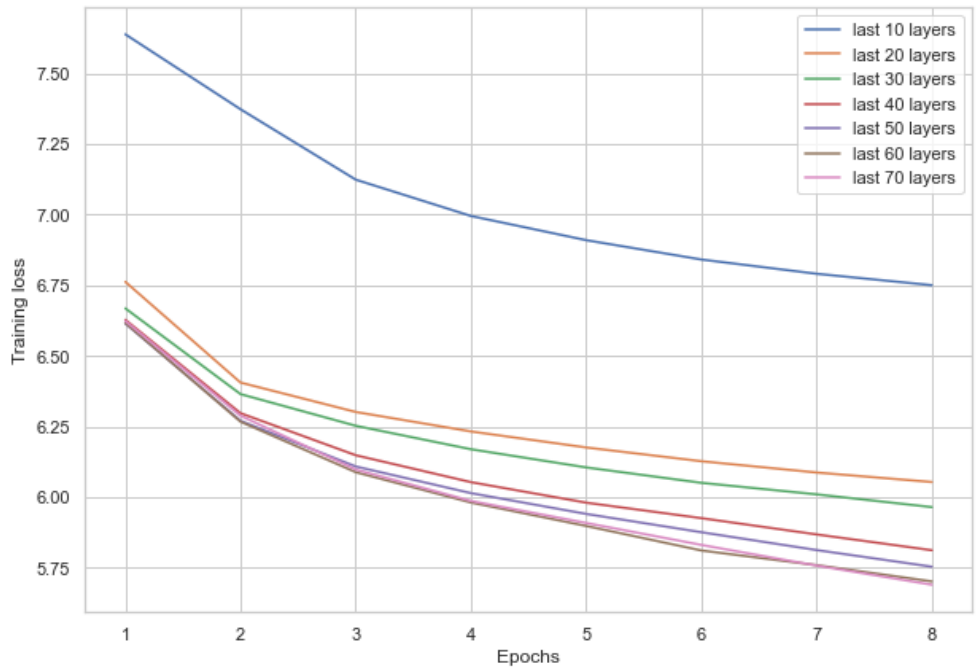


Figure 5.1 Training loss versus Epochs for different model training settings

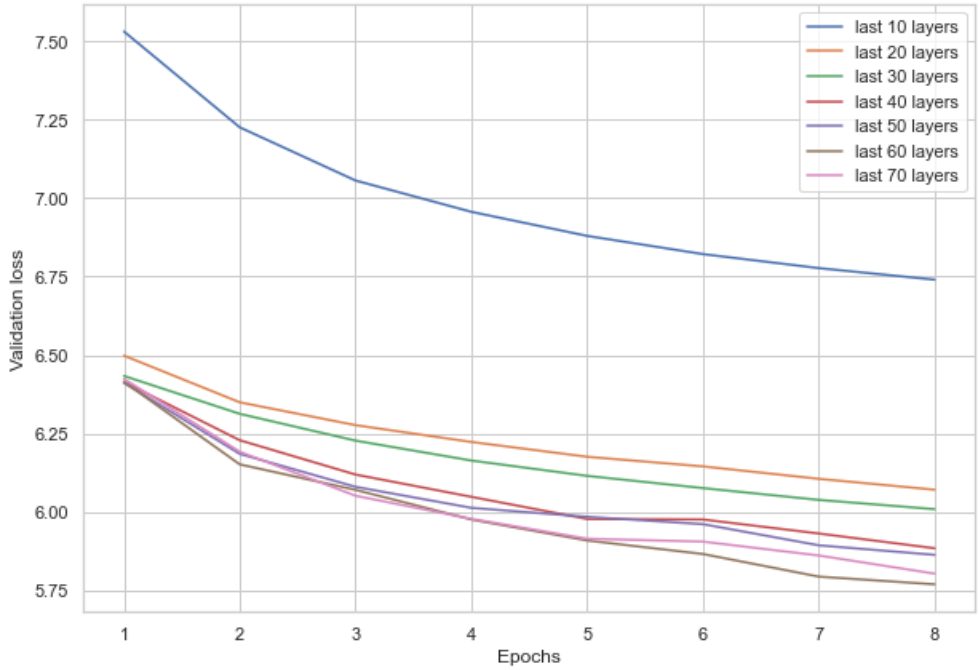


Figure 5.2 Validation loss versus Epochs for different model training settings

Further, the validation loss for a different number of epochs and different model training settings are also plotted. From Figure 5.2 above, we note the clear advantage in training the model for a greater number of epochs and with the higher number of layers set as trainable. The negative side of the above result however is that by increasing the number of trainable layers during the transfer learning process, the improvement in the model performance is not very substantial. Perhaps the same might be due to the bulkiness of the chosen core model.

5.3 Model Performance Evaluation of Testset

Models with different settings are trained for 8 epochs. The performance of trained models with different settings is tested on the test dataset. The various box plots depicted in Figure 5.3 below convey the distance predicted by the models with different settings for the input Genuine and Impostor masked-unmasked face image pairs from the test dataset. Here, Genuine pairs are encoded as 1 while the impostor pairs are encoded as 0. It is prudent to note that for all the models, the distance predicted for Genuine pairs is less than that for the Impostor pairs. As the number of trainable parameters increases in the core SENet50 model, the gap between predicted distance scores for genuine and impostor pairs rises. The verdict from the above experimentation is that through model training, we can obtain higher differences in these distances so that it becomes easy to choose a threshold that can differentiate Genuine from the

Impostor pairs. The more flexibility we have in choosing the distance threshold, the easier it becomes to build a masked-unmasked face recognition system employing this trained model.

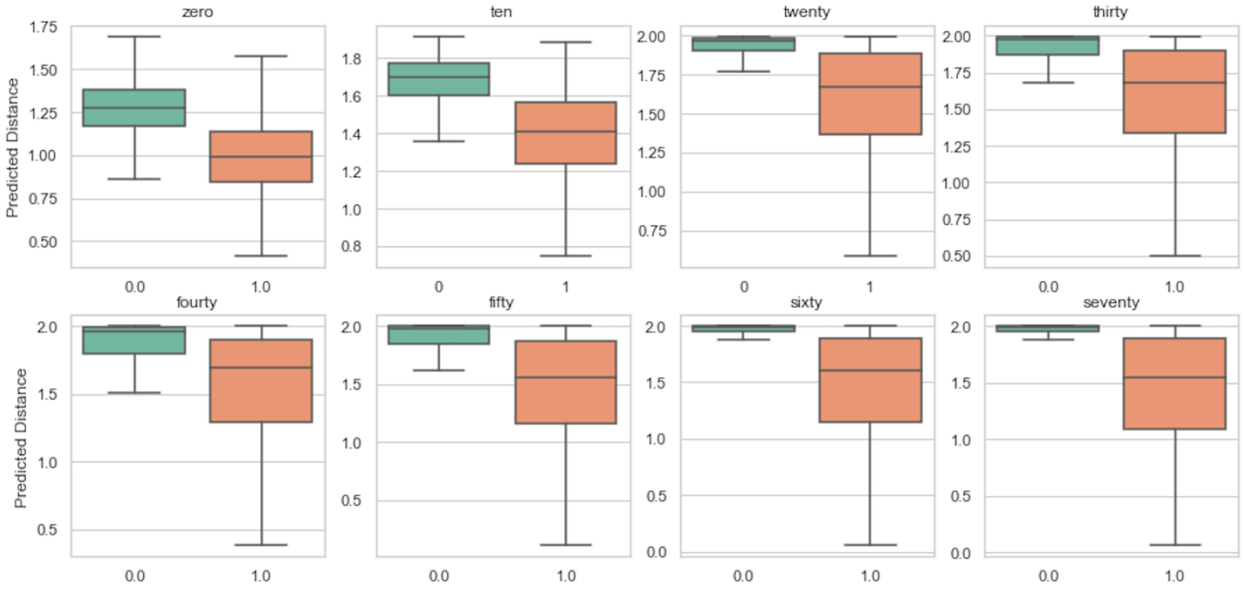


Figure 5.3 Predicted distance scores for different model training settings

To better interpret the obtained results, the median of the distance score for Genuine and Impostor pairs from the test dataset was calculated and compared. The median values of the above-predicted distances are brought out in Table 5.1 below.

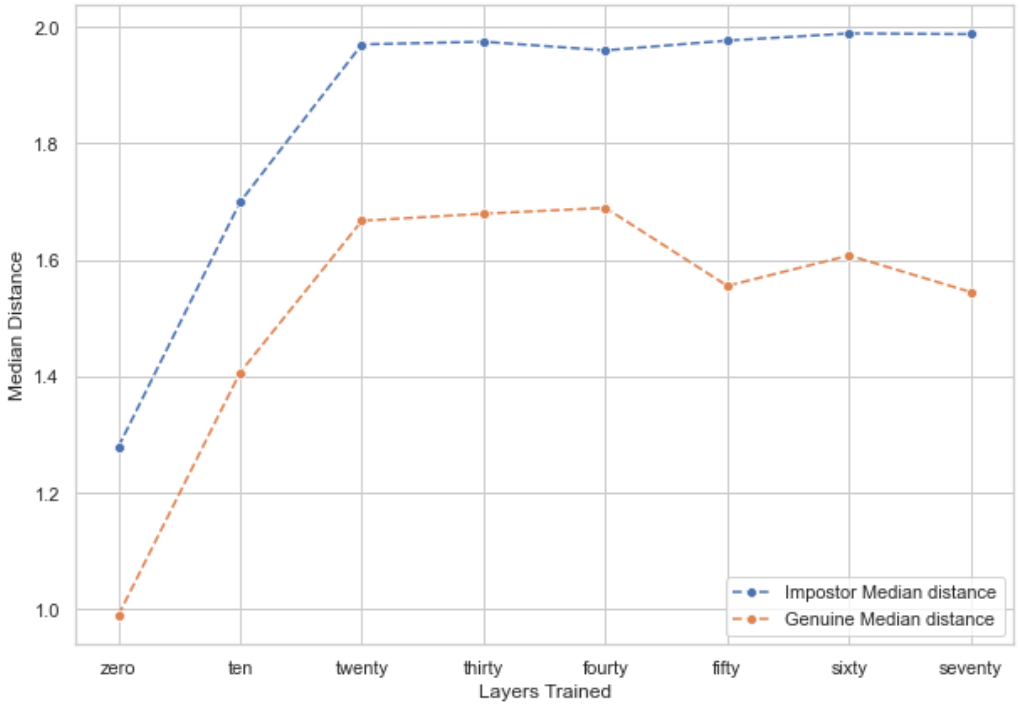


Figure 5.4 Median predicted distance score for different model training settings

As depicted in Figure 5.4 above, by increasing the number of trainable layers in the core model, though the overall distance predicted by the model increases, the difference between the median value of distance score for Genuine and Impostor pairs widen. The same is crucial since with this we could have more flexibility in choosing the threshold value which defines whether the input pair is Genuine or Impostor.

Table 5.1 Median Genuine and Impostor pair distance score for different model training settings

Model setting trained for 8 epochs	***Median Impostor Distance***	***Median Genuine Distance***
SENet50 pre-trained	1.278899	0.989988
SENet50 Siamese trained last 10 layers	1.699373	1.406768
SENet50 Siamese trained last 20 layers	1.969977	1.667269
SENet50 Siamese trained last 30 layers	1.974622	1.679480
SENet50 Siamese trained last 40 layers	1.959761	1.689339
SENet50 Siamese trained last 50 layers	1.976508	1.555529
SENet50 Siamese trained last 60 layers	1.988818	1.607727
SENet50 Siamese trained last 70 layers	1.987619	1.544770

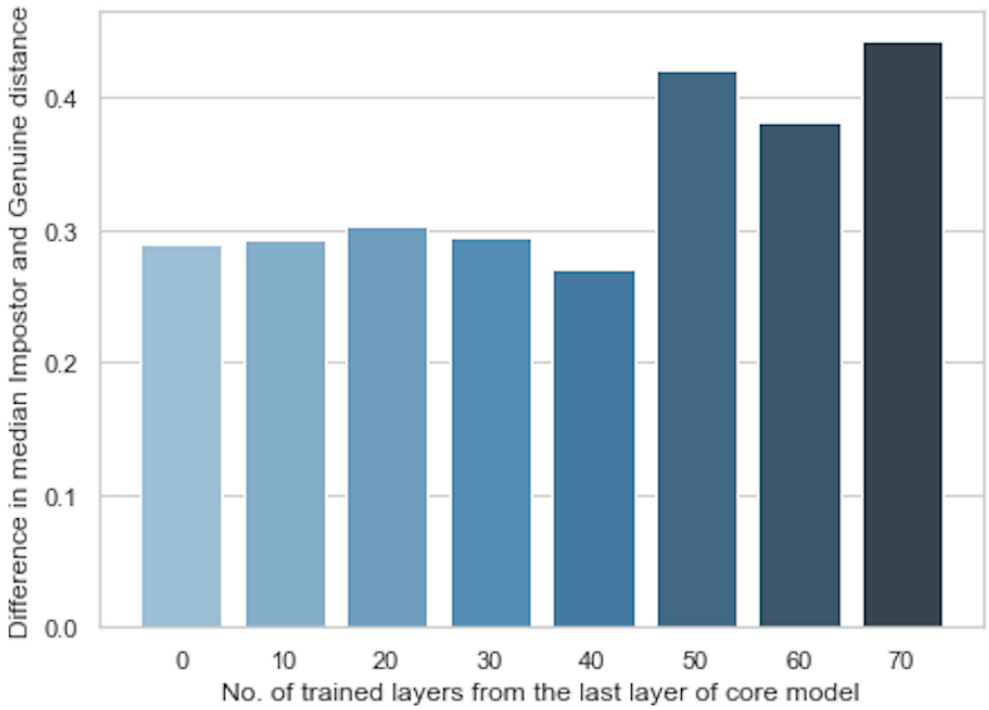


Figure 5.5 Difference in the median distance for Impostor and Genuine pairs for different model training settings

Further, to check whether training the higher number of layers in the core model with the Siamese Network architecture and proposed customer loss function results in a greater difference in the distances of Genuine masked-unmasked face images pairs and Impostor masked-unmasked face image pairs, a graph is plotted between the difference in median Impostor and Genuine distance scores versus the different number of trained layers of the core SENet50 model. The dataset used here is a test set, to avoid data leakage and compare the actual performance of the trained model. As illustrated in Figure 5.5 above, the difference in the distance between the Genuine and Impostor masked-unmasked face image pair shows a positive trend with the increase in the number of trained layers.

5.4 Grad-CAM output

One of the research questions for this study is to check whether the SE layer aids in turning the focus of the model to the non-occluded portion of the face. To test which part of the face the trained model is paying attention to while generating the embeddings for the input masked face image, the Grad-CAM model is used, and the corresponding output is depicted below in Figure 5.6.

Figure 5.6 Grad-CAM output for masked face images

Grad-CAM was proposed by (Selvaraju et al., 2016) wherein they differentiated the output with respect to different feature maps, which are then normalized and summed to obtain a coarse heat map. This heatmap is then overlapped on the actual input image to create a visual depicting where the model is stressing more to come up with the specific output embeddings.

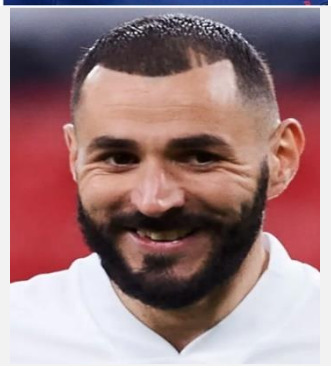
We used the model with the last 70 layers train for 8 epochs on our partitioned train dataset, to come up with the Grad-CAM visual of our own. We used some sample masked images from the test dataset to see where our model is focusing while identifying the subject. As illustrated in Figure 5.6 above, the portion of the face around the eyes and some parts of the forehead are faint yellow while the rest are dark bluish in the color. The same indicates that our trained model pays more attention to the non-occluded region as compared to the mask-occluded region, during prediction.

5.5 Performance Test on Random Image Pairs

Testing the model performance on the dataset with altogether different demography could be a real challenge as the skin color, face symmetry, and eye shape could be very different from the face dataset on which the model is trained. To test the model on the subject with different demography, a few images were scrapped from the internet and the model prediction was run on the same. The results obtained are described in Table 5.2 below.

Table 5.2 Checking the trained model performance on random masked-unmasked face image pairs

Input Masked Image	***Input Unmasked Image***	***Distance Score***
		1.3827795

		1.0776693
		1.6497619
		1.9071597
		1.9340677

It could be noted from the referred table that Genuine pairs i.e. masked-unmasked face image pairs belonging to the common subject have low distance scores as compared to the Impostor pairs. It is noted here that the gap between the distance score of Genuine and Impostor pairs is not substantial, however, we hypothesize that the same could be enhanced by training the model to more depth and for a greater number of epochs. We reserve the same for future work due to the limitations of the available resources.

5.6 Summary

This chapter deliberated the obtained results on the different training settings and evaluated the model performance on the test as well as random face image pairs. The model training phase is properly monitored to notice the abnormalities during the training. Thereafter, model performance is evaluated on the test dataset to check the effectiveness of the transfer learning with siamese architecture. Next, the Grad-CAM algorithm was employed to check whether the SE layers employed in the model are aiding in turning the focus on the non-occluded region of the face while giving out the embeddings of the input faces. Finally, the chapter ends with testing the model performance on the randomly scraped internet images.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Introduction

This chapter discusses the various conclusions drawn and the recommendations made based on the conducted study. The first section discusses the overview of the various outcomes and results in addition to concluding the same. The next one put up the contribution made by this work to the knowledge of the research community. Finally, the last section gives recommendations and directions for future work in this field of study.

6.2 Discussion and Conclusion

These days, wearing a mask has become a new trend. People wear it for various reasons but this trend has become a great challenge for the face recognition models. Occlusion could be challenging for human eyes as well, let alone the deep learning models. Hence, focusing on the important, non-occluded region is important while identifying the subject. It is apparent that by nature, machine learning models do not have the capability of focusing on the important stuff. So, instilling this value in a machine learning model is important to ease the arduous job of identifying different subjects wearing a face mask. To move toward this direction, we decided to use the available attention mechanism could and check whether it aids the model in any possible way, in identifying the subject wearing the mask. Specifically in this study, we explored the capability of the Shift and Excitation layer in tackling the task of masked-unmasked face recognition.

Ideally, a face recognition model should be extremely generalizable in a way that it could differentiate genuine face image pairs from the impostor ones. Keeping this at the back of the head, it was decided to choose a pre-trained model that is already trained on the face images dataset and fine-tune the same for our task-specific dataset. That way, we not only avoid the risk of overfitting the model on the chosen dataset but also save the limitedly available computational resources. Hence, the SENet50 model pre-trained on the VGGFace dataset is chosen as a base core model and we decided to build the siamese network on top of the same and fine-tune it on the chosen Real-World Masked Face Dataset.

Various experiments were run and various tests were conducted, with different model training settings, to draw different conclusions out of the same. The major one is that the attention mechanism aid in making the model learn to focus on the important region of the face image rather than the non-important, occluded one. This answers the first research question that we initially proposed. We further noted that the model fine-tuning with matric learning and a custom loss function aid in instilling the generalization capability for the masked-unmasked face recognition challenge. This answers that second research question of ours affirmatively. We further agree that the transfer learning with a model pre-trained on a similar task helps in achieving good results with a fewer number of iterations.

6.3 Contribution to Knowledge

From the experimental results obtained, we may draw that this study could have many-fold implications for the research community. Firstly, when some novel architectural ideas are produced, we may need a benchmarking scheme with which we could compare the performance and hence evaluate the same. Secondly, siamese architecture could lead to promising results in this field of study due to its generalization capability. Thirdly, though the experimental results provide that the involvement of the Shift and Excitation layer could aid in inculcating the attention mechanism in the model and hence helps in achieving the target results, further shreds of evidence could always firm up the produced statement.

6.4 Limitations of the Work

This is work, the proposed model training with various settings is shown, and through firm results, the proposed hypothesis was proven on Real-World Masked Face Dataset (RMFD). But there may be some limitations when it comes to training the same architecture, and model arrangements on a similar dataset from a different ethnic group. RMFD contains face images of subjects from the Asian community, with most of the faces without items like face tattoos, special clothes, pieces of jewelry, etc. The presence of these objects might hinder or limit the model to get trained at its best and the same is not explored, or presented in this work.

Since the attention mechanism is deemed a crucial part of the success of this work, the latest state-of-the-art models like a transformer, etc. could enhance the metric value however, employability of the same is not explored in this work since these models are overwhelmingly bulky and have relatively very high number trainable parameters. One of the aims at the onset

of this work was also to come up with a simple and easy to train solution and hence proposed model was chosen for exploration.

6.5 Future Work

This study could prove to be an important reference point in the field of masked-unmasked face recognition challenges where traditional face recognition models could not perform that well. Building on the top of the presented work, different layers with attention mechanisms could be explored that may provide a relatively better edge as compared to the Shift and Excitation layer in focusing on the relevant part of the face images. The convolutional block attention module layer could be one of the prospective promising contenders. Other metric learning architecture could be tried with similar experimental settings to truly realize the power of the network. The Triplet Network architecture with similar custom loss might lead to promising results, given the existing literature within the research community. We also think that training the model for an optimal number of iterations could further improve the obtained results but parallelly care should be taken to avoid the model overfitting. This work should encourage the private IT giants and government organizations to attempt to build more robust systems and open-source larger masked-unmasked face datasets that can aid the research community in contributing more to this field.

REFERENCES

Cao, Q., Shen, L., Xie, W., Parkhi, O.M. and Zisserman, A., (2018) Vggface2: A dataset for recognising faces across pose and age. In: *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, pp.67–74.

Chopra, S., Hadsell, R. and LeCun, Y., (2005) Learning a similarity metric discriminatively, with application to face verification. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, pp.539–546.

Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K. and Bengio, Y., (2015) Attention-Based Models for Speech Recognition. In: C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett, eds., *Advances in Neural Information Processing Systems*. [online] Curran Associates, Inc. Available at: https://proceedings.neurips.cc/paper/2015/file/1068c6e4c8051cfd4e9ea8072e3189e2-Paper.pdf.

Deng, J., Guo, J., Xue, N. and Zafeiriou, S., (2019) Arcface: Additive angular margin loss for deep face recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp.4690–4699.

Deng, J., Zhou, Y. and Zafeiriou, S., (2017) Marginal loss for deep face recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pp.60–68.

Ding, F., Peng, P., Huang, Y., Geng, M. and Tian, Y., (2020) Masked face recognition with latent part detection. In: *Proceedings of the 28th ACM International Conference on Multimedia*. pp.2281–2289.

Fu, J., Zheng, H. and Mei, T., (2017) Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.4438–4446.

Ghojogh, B., Sikaroudi, M., Shafiei, S., Tizhoosh, H.R., Karray, F. and Crowley, M., (2020) Fisher discriminant triplet and contrastive losses for training siamese networks. In: *2020 international joint conference on neural networks (IJCNN)*. IEEE, pp.1–7.

Hariri, W., (2021) Efficient Masked Face Recognition Method during the COVID-19 Pandemic. *Signal, Image and Video Processing*. [online] Available at: https://arxiv.org/abs/2105.03026v1 [Accessed 28 Jan. 2022].

Hoffer, E. and Ailon, N., (2015) Deep metric learning using triplet network. In: *International workshop on similarity-based pattern recognition*. Springer, pp.84–92.

Hu, J., Shen, L., Albanie, S., Sun, G. and Wu, E., (2017) Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, [online] 428, pp.2011–2023. Available at: https://arxiv.org/abs/1709.01507v4 [Accessed 6 Mar. 2022].

Huang, G.B., Mattar, M., Berg, T. and Learned-Miller, E., (2008) Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In: *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*.

Koch, G., Zemel, R. and Salakhutdinov, R., (2015) Siamese neural networks for one-shot image recognition. In: *ICML deep learning workshop*. Lille, p.0.

Kumar BG, V., Carneiro, G. and Reid, I., (2016) Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.5385–5394.

Li, C., Ge, S., Zhang, D. and Li, J., (2020) Look through Masks: Towards Masked Face Recognition with De-Occlusion Distillation. *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia*, [online] 20, pp.3016–3024. Available at: https://doi.org/10.1145/3394171.3413960 [Accessed 28 Jan. 2022].

Li, Y., Guo, K., Lu, Y. and Liu, L., (2021) Cropping and attention based approach for masked face recognition. *Applied Intelligence*, [online] 515, pp.3012–3025. Available at: https://link.springer.com/article/10.1007/s10489-020-02100-9 [Accessed 28 Jan. 2022].

Liu, J., Deng, Y., Bai, T., Wei, Z. and Huang, C., (2015) Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*.

Melekhov, I., Kannala, J. and Rahtu, E., (2016) Siamese network features for image matching. In: *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, pp.378–383.

Mishra, S., Majumdar, P., Singh, R. and Vatsa, M., (2021a) Indian Masked Faces in the Wild Dataset. *arXiv preprint arXiv:2106.09670*.

Mishra, S., Majumdar, P., Singh, R. and Vatsa, M., (2021b) Indian Masked Faces in the Wild Dataset. pp.884–888.

Neto, P.C., Boutros, F., Pinto, J.R., Saffari, M., Damer, N., Sequeira, A.F. and Cardoso, J.S., (2021) My Eyes Are Up Here: Promoting Focus on Uncovered Regions in Masked Face Recognition. [online] Available at: http://arxiv.org/abs/2108.00996.

Parkhi, O.M., Vedaldi, A. and Zisserman, A., (2015) Deep face recognition.

Sanjaya, S.A. and Rakhmawan, S.A., (2020) Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic. In: *2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*. pp.1–5.

Schroff, F., Kalenichenko, D. and Philbin, J., (2015) FaceNet: A Unified Embedding for Face Recognition and Clustering. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, [online] 07-12-June-2015, pp.815–823. Available at: http://arxiv.org/abs/1503.03832 [Accessed 6 Mar. 2022].

Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D. and Batra, D., (2016) Grad-CAM: Why did you say that? *arXiv preprint arXiv:1611.07450*.

Song, L., Gong, D., Li, Z., Liu, C. and Liu, W., (2019) *Occlusion Robust Face Recognition Based on Mask Learning With Pairwise Differential Siamese Network*.

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L., (2014) Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.1701–1708.

Ud Din, N., Javed, K., Bae, S. and Yi, J., (2020) A Novel GAN-Based Network for Unmasking of Masked Face. *IEEE Access*, 8, pp.44276–44287.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X. and Tang, X., (2017) Residual Attention Network for Image Classification. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, [online] 2017-January, pp.6450–6458. Available at: https://arxiv.org/abs/1704.06904v1 [Accessed 6 Mar. 2022].

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z. and Liu, W., (2018) Cosface: Large margin cosine loss for deep face recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.5265–5274.

Wang, M. and Deng, W., (2021) Deep face recognition: A survey. *Neurocomputing*, 429, pp.215–244.

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. *arXiv preprint arXiv:2003.09093*.

Woo, S., Park, J., Lee, J.-Y. and Kweon, I.S., (2018) *CBAM: Convolutional Block Attention Module*.

Yi, D., Lei, Z., Liao, S. and Li, S.Z., (2014) Learning Face Representation from Scratch. [online] Available at: http://arxiv.org/abs/1411.7923 [Accessed 10 Mar. 2022].

You, Q., Jin, H., Wang, Z., Fang, C. and Luo, J., (2016) Image captioning with semantic attention. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.4651–4659.

Zheng, H., Fu, J., Mei, T. and Luo, J., (2017) Learning multi-attention convolutional neural network for fine-grained image recognition. In: *Proceedings of the IEEE international conference on computer vision*. pp.5209–5217.

Zhou, X., Wan, X. and Xiao, J., (2016) Attention-based LSTM network for cross-lingual sentiment classification. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. pp.247–256.

APPENDIX A: RESEARCH PROPOSAL

MASKED FACE RECOGNITION WITH SIAMESE NETWORK-BASED METRIC LEARNING

KEVIN RASIKBHAI AKBARI

Research Proposal

OCTOBER 2021

Abstract

Face recognition has become a widespread technology everywhere. It is not only being used to grant access to different places but is also considered useful in identifying prospective threats by keeping track of the movements of people. However, these days more and more people have started wearing masks to protect themselves from harmful fumes, pollution, viruses, etc. When it comes to masked-face recognition, models trained for unmasked face recognition task fails to perform to the desired level. Occlusion caused due to the usage of the mask makes it tough for machine learning algorithms to differentiate between genuine and impostor pairs of face images. This study proposes a Siamese network-based model for the development of a robust masked face recognition system that can give accurate results in tasks such as face verification - one on one mapping and face recognition - one to many mapping. It is also proposed to use attention-based layers in network architecture to focus on the relevant portion of face images. Hence, the Siamese network along with the attention mechanism is a proposed network architecture for this project. The purpose behind the proposed architecture is to make the model learn to focus more on regions in and around the eyes rather than the occluded areas. It is expected that the model would give higher weightage to the non-occluded part of the face rather than the masked/ occluded part and hence learn to perform better on the masked face recognition task.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CNN.....	Convolutional Neural Network
GAN.....	Generative Adversarial Network
GMean.....	Genuine Mean Distance
IMean.....	Impostors Mean Distance
LFW.....	Labeled Faces in the Wild
RMFD.....	Real-World Masked Face Dataset
RMFRD.....	Real-world masked face recognition dataset
RMFVR.....	Real-world masked face verification dataset
SMFRD.....	Simulated masked face recognition datasets
SOTA.....	State of the Art

1. Background

Ever since the discovery of Convolutional Neural Network (CNN) architectures, the challenge of modeling unstructured data is eased to a great extent. It is especially true for the domain of image, and video analysis wherein these computer vision algorithms are being used predominantly. The success of CNN in this domain comes from its ability to detect higher-end correlations/patterns amongst the neighboring features/pixels in the image.

Convolutional Neural Networks are being used for image classification tasks wherein they are required to be trained with enough number of training examples per class, to achieve good performance. However, when it comes to the problem of face verification/ recognition, the number of classes can be in the thousands. Hence, it is not always feasible to obtain a sufficient number of images per class to train the network, which may lead to the problem of class imbalance. A common solution to the above problem is to use a metric learning-based approach and learn semantically-sound lower dimension representations of higher dimension features. Here, semantically-sound means to find a function that can preserve the similarity and/or dissimilarity of different higher dimensional data points in the corresponding lower dimension feature space. Researchers have proposed Siamese Network to model such a function.

One of the tasks from this domain is the challenge of face recognition and verification. Face recognition is a many-to-one mapping where we have a target face and a database of different subject face images and we declare the identity of the target subject based on the distance measure. Likewise, face verification is one-to-one mapping where we have a threshold that decides whether two face images belong to the same subject or not (Ding et al., 2020). Face recognition, verification tasks are not new to the research community. Solutions with human-level performance (Wang et al., n.d.) are in place to precisely differentiate between face images of the same subject - genuine pair, different subject - impostor pair. However, wearing a face mask has become a common trend among people these days due to rising pollution levels, the outbreak of deadly viruses, etc. Making people remove their masks for their identity verification purpose is neither logical nor safe. Besides, the existing face recognition systems have their limitations when it comes to occluded face recognition and verification tasks. The purpose of this study is to understand the problem of masked-face recognition and

propose a probable solution that can improve the performance of the existing unmasked face recognition models on this task.

2. Problem Statement

Face recognition and verification technology are being used these days for non-trivial tasks like authentication, attendance systems, phone unlocking, finding the subject of interest from surveillance clips, etc. Researchers have proposed various methods to achieve state-of-the-art metric values. As per the author's knowledge, probably the first breakthrough occurred with DeepFace (Taigman et al., 2014) model. They derived face representation from modeled 3D faces and were able to achieve close to 97.35% accuracy on the Labeled Faces in the Wild (LFW) dataset. Baidu (Liu et al., 2015) proposed a deep CNN-based model, employing metric learning and achieved pairwise verification accuracy close to 99.77% on the LFW dataset. They argued that their proposed model could achieve this feat because of the usage of features from different patches of the image. In addition to experimenting with the different network architectures, researchers have also explored various loss functions and proposed losses viz. Marginal Loss (Deng et al., 2017), Cosface (Wang et al., 2018), and Arcface (Deng et al., 2019), etc., and were able to achieve SOTA accuracy on the LFW dataset.

Obstruction caused due to foreign object(s) kept against the area of interest is called Occlusion. Facial occlusion could be caused by apparel items like a mask, scarf, cap, hat, etc. Mask has become a common cause of occlusion as the practice of wearing the same while in public places has become a common trend these days due to rising air pollution, the spread of contagious diseases, the outbreak of viruses, etc. Hence, the application of machine learning techniques on masked faces has become a trending area of research. Broadly there are two main tasks namely face mask detection and masked face recognition. The objective behind the former is to check whether the subject is wearing a mask or not. While the task under the latter is to identify the subject while he/she is wearing a mask. The focus of the proposed work is on the problem of masked-face recognition.

Recognizing the face of the subject while having the mask on is an arduous task since the major portion of the face gets occluded. When it comes to the task of face recognition, nose and mouth areas are important as semantically they reveal the identity of the subject. So, the normal algorithm for face recognition becomes less useful (Mishra et al., 2021) for the

masked face recognition task. Of let, computer vision researchers have turned their focus toward this field but still, it is relatively less developed. Two broad approaches are being adopted for solving this problem. One is recovery/unmasking of the masked part with generative models and the second is discarding the occluded/masked part of the face and considering only the visible region for the analysis. (Ud Din et al., 2020), (Li et al., 2020) proposed a Generative Adversarial Network (GAN) based method to recover the masked portion of the face and produced qualitatively as well as quantitatively well-performing model. However, the recovery-based methods are computationally expensive.

Several models have been developed which altogether discard the occluded region and use the remaining portion of the face for the task of recognition. (Hariri, 2021) discarded masked area and used the non-occluded region to fine-tune the pre-trained CNN model. The features extracted from the deep CNN model were used to achieve high classification accuracy on the Real-World Masked Face Dataset. The features extracted from the unmasked portion of the image are more useful and hence more weightage should be given to them as compared to masked region features. (Song et al., 2019) used the pairwise differential Siamese network to eliminate the corrupt portion of the face and utilized the rest for face recognition. (Li et al., 2021) adopted a mix of discarding and attention mechanism to achieve high metric value in the task of masked face recognition.

While identifying a person with a mask, we as a human intuitively focus on the area near the eyes. We know that in the masked face recognition task, the occlusion always starts from just above the nose tip and ends at the chin part of the face, and hence we effortlessly pay more attention to the non-occluded part, but the model doesn't. We use this knowledge and propose incorporating the attention-based module in deep CNN and make the model focus more on the unmasked portion of the face rather than the masked portion for the proposed problem.

3. Aim and Objectives

The main aim of this research is to propose an attention-based approach to enhance the capability of the existing face recognition model using the Real World Masked Face Dataset (RMFD). The goal of this project is to come up with a robust masked face recognition model such that occlusion due to the mask does not restrict the ability of the model to differentiate between the impostor and genuine face images.

The research objectives formulated based on the aim of this study are as follows:

- To investigate the capability of an attention-based convolutional neural network in achieving the tasks of masked and unmasked face recognition.
- To experimentally determine the optimum parameter settings of the Siamese network.
- To propose an attention-based Siamese network to detect the similarity among different faces.
- To evaluate the performance of the Siamese network with various distance measures.

4. Significance of the Study

Given the current scenario of pandemic and its prospects, working towards the development of occlusion robust systems is of utmost importance. Such technologies could help in promoting the usage of face masks and hence reducing the spread of deadly viruses. It is expected that this study would bring the focus of the research community towards the application of attention mechanisms in solving the problem of face recognition with different occlusion levels. Usage of such technology could aid in upkeeping law and order in society with minimal impact on basic human rights.

5. Scope of the Study

The scope of this study is to explore the effectiveness of incorporating an attention mechanism in the masked face recognition task. Tasks like human face detection in the image, and face-mask detection are not the objectives of this study. Further, the development of a state-of-the-art attention module/ layer for a deep convolutional neural network model is also not in the current scope. The scope of the study is explicitly defined beforehand to adhere to the limited time, and resources available for the proposed project.

6. Research Methodology

6.1 Introduction and proposed flow

The proposed methodology involves key processes such as choosing the dataset, pre-processing the chosen data, transforming the data into the proper format, balancing the masked-unmasked dataset, training the model, and evaluating the performance using evaluation metrics. The tentative flowchart for the proposed project is shown in Figure 1.

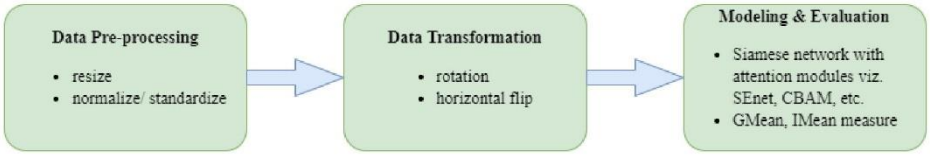


Figure 1. Project flow chart

6.2 Dataset description

It is proposed to use Real-world masked face recognition dataset (RMFRD) for the proposed project. It contains different masked as well as unmasked face images of different subjects. Those face images have varied expressions, and poses, which can help our model generalize better. The originally published dataset with the title Real-World Masked Face Dataset (RMFD) has three subsets viz. Real-world masked face recognition dataset (RMFRD), Simulated masked face recognition datasets (SMFRD), and Real-world masked face verification dataset (RMFVR). Real-world masked face recognition dataset (RMFRD) is our focus of interest. It contains web-crawled colored images of about 525 different subjects. There is a total of 5,000 masked and 90,000 unmasked faces for those 525 subjects. The ethnicity of all the subjects in this dataset is Asian.

6.3 Data preprocessing and augmentation

These face images in RMFRD are already centered and cropped properly to be readily used by any machine learning model. However, the size of those images is not the same. Hence it is proposed to resize them to a standard size for compatibility purposes. Further, the images are in raw jpeg format so the model-specific preprocessing namely normalizing or standardizing could be performed before feeding them into the model. The description of the data is such that the number of masked images is far lesser as compared to the unmasked images. The same may cause class imbalance. So, if needed, we may go ahead with the image augmentation on minority classes with techniques like rotation, horizontal flipping, etc.

6.4 Modelling Techniques

Since we want to make a model generalize well on unseen subjects, training the model for classification tasks is certainly not a good option. When it comes to face recognition, we can never have images of all the subjects for training the model so, a matric learning-based approach is preferred for such tasks. It is proposed to make a deep convolutional neural

network learn to map higher-dimensional images to the corresponding lower-dimensional embeddings. A Siamese Network-based learning, utilizing contrastive loss as an optimization function is proposed, to come up with such a mapping function. This could help the model in learning to transform the higher-dimensional images into the lower-dimensional embeddings, preserving the semantic similarities and dissimilarities among them. The L2 distance among those embeddings could then be measured and a decision could be made on the identity of the subject.

6.5 Proposed model

One more challenge to tackle is the effect of collusion due to the mask. Occlusion created by the mask affects the areas just above the nose tip and to the chin. So, the occluded area is deemed not so useful for the task at hand. It is proposed to use an attention-based mechanism to make the model give higher weightage to the non-occluded part of the face while creating the lower dimensional embedding of the input image. It is proposed to use the Squeeze-and-Excitation block (Hu et al., 2018) in the deep CNN model, as the same is proven to perform well when the task at hand needs attention to specific areas in the image. For model training, we propose to use contrastive loss first proposed by (Chopra et al., n.d.) for their work on face recognition. The block diagram for the model training and evaluation process could look as shown in Figure 2.

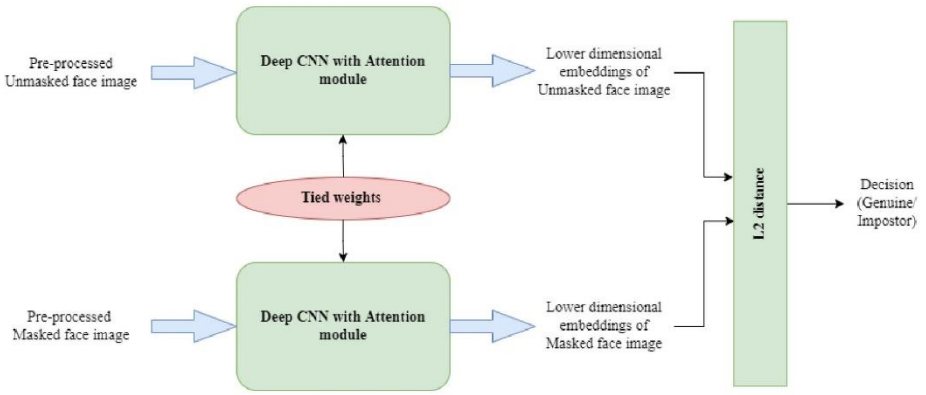


Figure 2. Model block diagram

6.6 Performance metric

To test the differentiation ability of the trained model between genuine and impostor pairs of input masked, unmasked face images, it is proposed to use Genuine Mean Distance (GMean)

and Impostors Mean Distance (IMean), as has been used by (Neto et al., 2021). GMean is the mean L2 distance among all the genuine pairs of face images while IMean is the mean L2 distance among all the impostor pairs. It is estimated that the model would be able to give face embeddings such that GMean is less than the IMean.

7. Required Resources

Upon prima facie estimation, broadly there are two main categories of required resources for the proposed research project namely, hardware and software.

7.1 Hardware Requirements

Under the hardware resources, the project would need GPU-supported computing platforms such as Jupyter Notebook, and Google Colab. Given the size of data, the hardware specifications required would be around 16 GB of RAM, 4 GB of Graphics memory, and a processor with a clock speed 2.50GHz or higher.

7.2 Software Requirements

Under the software resources, the project would need python version 3.5 or higher. For the data handling, visualization, and pre-processing, python libraries viz. Pandas (1.4.0), Numpy (1.22.1), Matplotlib (3.5.1), Seaborn (0.11.2) could come into use. The latest version of Machine Learning frameworks like TensorFlow (2.7.0), Keras (2.7), Scikit-learn (1.0.2), etc. would be required for model training and evaluation purpose. We may go ahead with transfer learning for the proposed model so pre-training model(s) trained on the face image dataset could come into use.

8. Research Plan

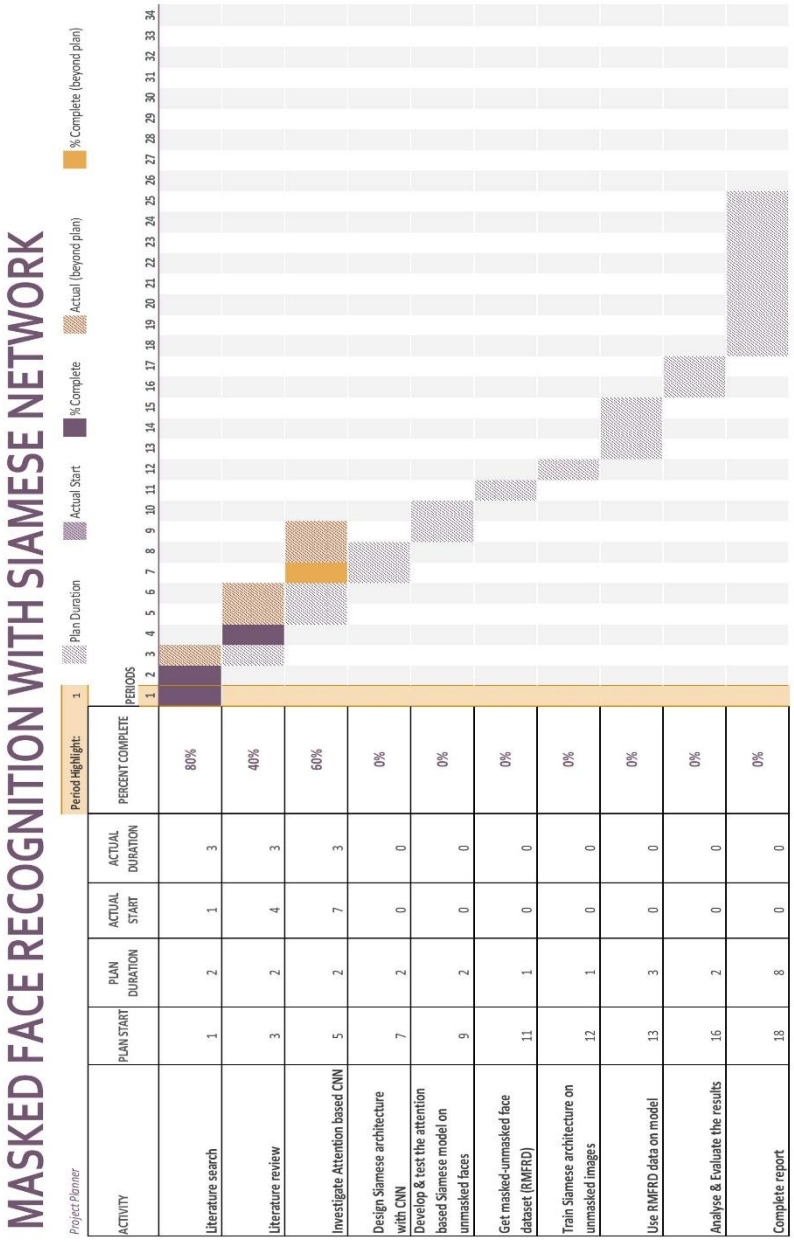


Figure 3. Project plan

References

Chopra, S., Hadsell, R. and Lecun, Y., (n.d.) *Learning a Similarity Metric Discriminatively, with Application to Face Verification*.

Deng, J., Guo, J., Xue, N. and Zafeiriou, S., (2019) Arcface: Additive angular margin loss for deep face recognition. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp.4690–4699.

Deng, J., Zhou, Y. and Zafeiriou, S., (2017) Marginal loss for deep face recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. pp.60–68.

Ding, F., Peng, P., Huang, Y., Geng, M. and Tian, Y., (2020) Masked face recognition with latent part detection. In: *Proceedings of the 28th ACM International Conference on Multimedia*. pp.2281–2289.

Hariri, W., (2021) Efficient Masked Face Recognition Method during the COVID-19 Pandemic. *Signal, Image and Video Processing*. [online] Available at: https://arxiv.org/abs/2105.03026v1 [Accessed 28 Jan. 2022].

Hu, J., Shen, L. and Sun, G., (2018) Squeeze-and-excitation networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.7132–7141.

Li, C., Ge, S., Zhang, D. and Li, J., (2020) Look through Masks: Towards Masked Face Recognition with De-Occlusion Distillation. *MM 2020 - Proceedings of the 28th ACM International Conference on Multimedia*, [online] 20, pp.3016–3024. Available at: https://doi.org/10.1145/3394171.3413960 [Accessed 28 Jan. 2022].

Li, Y., Guo, K., Lu, Y. and Liu, L., (2021) Cropping and attention based approach for masked face recognition. *Applied Intelligence*, [online] 515, pp.3012–3025. Available at: https://link.springer.com/article/10.1007/s10489-020-02100-9 [Accessed 28 Jan. 2022].

Liu, J., Deng, Y., Bai, T., Wei, Z. and Huang, C., (2015) Targeting ultimate accuracy: Face recognition via deep embedding. *arXiv preprint arXiv:1506.07310*.

Mishra, S., Majumdar, P., Singh, R. and Vatsa, M., (2021) Indian Masked Faces in the Wild Dataset. *arXiv preprint arXiv:2106.09670*.

Neto, P.C., Boutros, F., Pinto, J.R., Saffari, M., Damer, N., Sequeira, A.F. and Cardoso, J.S., (2021) My Eyes Are Up Here: Promoting Focus on Uncovered Regions in Masked Face Recognition. [online] Available at: http://arxiv.org/abs/2108.00996.

Song, L., Gong, D., Li, Z., Liu, C. and Liu, W., (2019) *Occlusion Robust Face Recognition Based on Mask Learning With Pairwise Differential Siamese Network*.

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L., (2014) Deepface: Closing the gap to human-level performance in face verification. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.1701–1708.

Ud Din, N., Javed, K., Bae, S. and Yi, J., (2020) A Novel GAN-Based Network for Unmasking of Masked Face. *IEEE Access*, 8, pp.44276–44287.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z. and Liu, W., (2018) Cosface: Large margin cosine loss for deep face recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp.5265–5274.

Wang, M., Neurocomputing, W.D.- and 2021, undefined, (n.d.) Deep face recognition: A survey. *Elsevier*. [online] Available at: https://www.sciencedirect.com/science/article/pii/S0925231220316945 [Accessed 29 Jan. 2022].

APPENDIX B: PYTHON CODE - DATA PARTITIONING AND PROCESSING

```
# Importing libraries
import glob
from PIL import Image
import glob
import pandas as pd
import random
from sklearn.model_selection import train_test_split
from sklearn.utils import shuffle


# Getting list of masked subjects
masked_name_list = []
for name in glob.glob('/content/sample_data/self-built-masked-face-
recognition-dataset/AFDB_masked_face_dataset/*'):
  masked_name_list.append(name.split('/')[-1])

# Getting list of unmasked subjects
unmasked_name_list = []
for name in glob.glob('/content/sample_data/self-built-masked-face-
recognition-dataset/AFDB_face_dataset/*'):
  unmasked_name_list.append(name.split('/')[-1])

# Getting the common subjects
masked_name_list = set(masked_name_list)
unmasked_name_list = set(unmasked_name_list)
common_names = masked_name_list.intersection(unmasked_name_list)
common_names = list(common_names)

# Sorting list for proper data division
common_names.sort()

# Setting ceiling to the number of picks (masked and unmasked) per
subject
max_pick = 20

# Choosing 80% of the total nubmer of subjects for creation of train
and validation datasets
no_of_subjects =  int(len(common_names)*0.8)

masked_image_list = []
for name in common_names[:no_of_subjects]:
  count=0
  for filename in glob.glob('/content/sample_data/self-built-masked-
face-recognition-dataset/AFDB_masked_face_dataset/'+name+'/*.jpg'):
      masked_image_list.append(filename)
      count+=1
```

```
      if count==max_pick:
        break

unmasked_image_list = []
for name in common_names[:no_of_subjects]:
  count=0
  for filename in glob.glob('/content/sample_data/self-built-masked-
face-recognition-dataset/AFDB_face_dataset/'+name+'/*.jpg'):
      unmasked_image_list.append(filename)
      count+=1
      if count==max_pick:
        break


# Shuffling the masked, unmasked images list
random.shuffle(masked_image_list)
random.shuffle(unmasked_image_list)

# Combining
fin_series = []
label_series = []
for mi1 in unmasked_image_list:
  for mi2 in masked_image_list:
    if mi1.split('/')[-2]==mi2.split('/')[-2]:
      lab=1.  # Genuine pair
    else:
      lab=0. # Impostor pair
    fin_series.append([mi1, mi2, lab])

# Creating DataFrame with masked face path, unmasked face path and
label
comb_ = pd.DataFrame(fin_series, columns = ['mi1', 'mi2', 'label'])
comb_same = comb_[comb_.label==1].reset_index(drop=True)
comb_diff = comb_[comb_.label==0].reset_index(drop=True)


# Splitting the created DataFrame into train and validation sets
comb_same_train, comb_same_val = train_test_split(comb_same,
                                          random_state=2021,
                                          test_size = 0.2)

comb_diff_train, comb_diff_val = train_test_split(comb_diff,
                                          random_state=2021,
                                          test_size = 0.2)



# Extracting the data such that no. of Genuine and Impostor pairs is
equal in strength
```

```
comb_train =
comb_same_train.append(comb_diff_train[:len(comb_same_train)]).reset
_index(drop=True)
comb_val =
comb_same_val.append(comb_diff_val[:len(comb_same_val)]).reset_index
(drop=True)


# Shuffling the obtained dataset
comb_train = shuffle(comb_train,
random_state=2021).reset_index(drop=True)
comb_val = shuffle(comb_val,
random_state=2021).reset_index(drop=True)


# Creating the test dataset
masked_image_list_test = []
for name in common_names[no_of_subjects:]:
  count=0
  for filename in glob.glob('/content/sample_data/self-built-masked-
face-recognition-dataset/AFDB_masked_face_dataset/'+name+'/*.jpg'):
      masked_image_list_test.append(filename)
      count+=1
      if count==max_pick:
        break

unmasked_image_list_test = []
for name in common_names[no_of_subjects:]:
  count=0
  for filename in glob.glob('/content/sample_data/self-built-masked-
face-recognition-dataset/AFDB_face_dataset/'+name+'/*.jpg'):
      unmasked_image_list_test.append(filename)
      count+=1
      if count==max_pick:
        break


# Combining
fin_series_test = []
for mi1 in unmasked_image_list_test:
  for mi2 in masked_image_list_test:
    if mi1.split('/')[-2]==mi2.split('/')[-2]:
      lab=1.  # Genuine pair
    else:
      lab=0. # Impostor pair

    fin_series_test.append([mi1, mi2, lab])
```

```
comb_test = pd.DataFrame(fin_series_test, columns = ['mi1', 'mi2',
'label'])
comb_same_test =
comb_test[comb_test.label==1].reset_index(drop=True)
comb_diff_test =
comb_test[comb_test.label==0].reset_index(drop=True)
comb_test_final =
comb_same_test.append(comb_diff_test[:len(comb_same_test)]).reset_in
dex(drop=True)



def _parse_function(filename_m, filename_um, label):
  '''
    - This is a parser function to create a data generator function
using the TensorFlow data API.

    - Here, we are not using preprocess_input(samples, version=2)
here as the same is just centering the face,
      which may be specific to the dataset.

  '''
  image_string_M = tf.io.read_file(filename_m)
  image_string_UM = tf.io.read_file(filename_um)

  image_decoded_M = tf.image.decode_jpeg(image_string_M, channels=3)
  image_decoded_UM = tf.image.decode_jpeg(image_string_UM,
channels=3)

  image_decoded_M = tf.image.resize(image_decoded_M, (224, 224))
  image_decoded_UM = tf.image.resize(image_decoded_UM, (224, 224))

  image_M = tf.cast(image_decoded_M, tf.float32)
  image_UM = tf.cast(image_decoded_UM, tf.float32)
  label = tf.cast(label, tf.float32)

  # model specific pre-processing [utils.preprocess_input(x,
version=2) for RESNET50 or SENET50]
  xx = tf.constant([91.4953,103.8827,131.0912], dtype=tf.float32)
  yy = tf.broadcast_to(xx, shape=(224,224,3))
  image_M -= yy
  image_UM -= yy

  return (image_M, image_UM), label

train_dataset = train_dataset.map(_parse_function,
                                  num_parallel_calls=tf.data.experim
ental.AUTOTUNE).batch(32)
```

```
# Creating Tensorflow batch Train dataset object

comb_train_m = tf.Variable(comb_train.mi1)
comb_train_um = tf.Variable(comb_train.mi2)
comb_train_label = tf.Variable(comb_train.label.astype('float32'))
train_dataset = tf.data.Dataset.from_tensor_slices((comb_train_m,
                                                    comb_train_um,
                                                    comb_train_labe
l))



# Creating tf batch Validation dataset object

comb_val_m = tf.Variable(comb_val.mi1)
comb_val_um = tf.Variable(comb_val.mi2)
comb_val_label = tf.Variable(comb_val.label.astype('float32'))
val_dataset = tf.data.Dataset.from_tensor_slices((comb_val_m,
                                                  comb_val_um,
                                                  comb_val_label)
)


val_dataset = val_dataset.map(_parse_function,
                              num_parallel_calls=tf.data.experim
ental.AUTOTUNE).batch(32)



# Creating tf batch Test dataset object

comb_test_m = tf.Variable(comb_test_final.mi1)
comb_test_um = tf.Variable(comb_test_final.mi2)
comb_test_label =
tf.Variable(comb_test_final.label.astype('float32'))
test_dataset = tf.data.Dataset.from_tensor_slices((comb_test_m,
                                                   comb_test_um,
                                                   comb_test_label
))


test_dataset = test_dataset.map(_parse_function,
                                num_parallel_calls=tf.data.experim
ental.AUTOTUNE).batch(32)
```

APPENDIX C: PYTHON CODE - SENet50 AND SIAMESE MODEL

```
# downloading the SENet50 pre-trained on vggface dataset
!sudo pip install git+https://github.com/rcmalli/keras-vggface.git

# check installation
!pip show keras-vggface
!pip install keras_applications

# importing the required libraries
import Keras.applications
from keras_vggface.vggface import VGGFace
from matplotlib import pyplot
from PIL import Image
from numpy import asarray
from scipy.spatial.distance import cosine
from keras_vggface.vggface import VGGFace
from keras_vggface.utils import preprocess_input
import tensorflow as tf


# loading the SENet50 model
model = VGGFace(model='senet50', include_top=False,
input_shape=(224, 224, 3), pooling='avg')

# setting the model layers non-trainable
for layer in model.layers:
  layer.trainable = False

'''
freezing the BatchNormalization layer weights, as per practice in
transfer learning
refer: https://keras.io/guides/transfer_learning/
'''
for layer in model.layers[-70:]:
  if 'BatchNormalization' in str(layer):
    layer.trainable = False
  else:
    layer.trainable = True

# defining input layer of model
inp = Input(shape=(224, 224,3))
y = model(inp, training=False)    # training=False inorder to keep bn
layer in inference mode

#getting core of siamese network
core_model = Model(inputs=inp, outputs=y)
```

```
# defining the complete Siamese network
input_shape = (224,224,3)
input_image_1 = Input(input_shape)
input_image_2 = Input(input_shape)

y1 = core_model(input_image_1, training=False)    # training=False
inorder to keep bn layer in inference mode
y2 = core_model(input_image_2, training=False)    # training=False
inorder to keep bn layer in inference mode

# adding the distance layer at the end
l2_distance_layer = Lambda(
            lambda tensors:
tf.math.reduce_sum(tf.math.squared_difference(tf.math.l2_normalize(t
ensors[0], axis=1) ,
                                                        tf.math.l2_no
rmalize(tensors[1], axis=1)), axis=1)
            )
l2_distance = l2_distance_layer([y1, y2])

# defining the overall Siamese Architecture
siamese_model = Model(
            inputs=[input_image_1, input_image_2],
            outputs=l2_distance
            )

# creating the custom loss function
def my_loss_fn(y_true, y_pred):
    y_true = tf.cast(y_true,tf.float32)
    y_pred = tf.cast(y_pred,tf.float32)
    return y_true*y_pred**2 + (1-y_true)*tf.math.maximum(0.,(5.0-
y_pred))**2

# model compilation
siamese_model.compile(loss= my_loss_fn,
                metrics= None,
                optimizer=tf.keras.optimizers.SGD(1e-5))
```