

Boosting Facial Recognition Capability for Faces Wearing Masks using Attention Augmented Residual Model with Quadruplet loss



By

Muhammad Aasharib Nawshad
Fall-2020-MS-CS 00000326842

Supervisor

Dr. Muhammad Moazam Fraz

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Computer Science (MSCS)

In

Machine Vision and Intelligent Systems (MachVIS) Lab,
School of Electrical Engineering & Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(December 2022)

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Boosting Facial Recognition Capability for Faces Wearing Masks using Attention Augmented Residual Model with Quadruplet loss" written by Muhammad Aasharib Nawshad, (Registration No 00000326842), of SEECs has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: _____  _____

Name of Advisor: Dr. Muhammad Moazam Fraz

Date: 14-Dec-2022 _____

HoD/Associate Dean: _____

Date: _____

Signature (Dean/Principal): _____

Date: _____

Approval

It is certified that the contents and form of the thesis entitled "Boosting Facial Recognition Capability for Faces Wearing Masks using Attention Augmented Residual Model with Quadruplet loss" submitted by Muhammad Aasharib Nawshad have been found satisfactory for the requirement of the degree

Advisor : Dr. Muhammad Moazam
Fraz

Signature: 

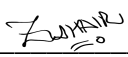
Date: 14-Dec-2022

Committee Member 1: Prof. Hasan Ali Khattak

Signature: 

15-Dec-2022

Committee Member 2: Dr Zuhair Zafar

Signature: 

Date: 14-Dec-2022

Signature: _____

Date: _____

Dedication

This thesis is dedicated to my beloved parents, family members, and teachers.

Certificate of Originality

I hereby declare that this submission titled "Boosting Facial Recognition Capability for Faces Wearing Masks using Attention Augmented Residual Model with Quadruplet loss" is my own work. To the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEecs or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEecs or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Student Name: Muhammad Aasharib Nawshad

Student Signature: 

Acknowledgments

Glory be to Allah (S.W.A), the Creator, the Sustainer of the Universe. Who only has the power to honour whom He please, and to abase whom He please. Verily no one can do anything without His will. From the day, I came to NUST till the day of my departure, He was the only one Who blessed me and opened ways for me, and showed me the path of success. There is nothing which can payback for His bounties throughout my research period to complete it successfully.

I also acknowledge the guidance of my thesis supervisor Dr. Muhammad Moazam Fraz, who taught me Computer Vision and Deep Learning from the fundamental level to the advanced level. He also provided timely advice, guidance, and high-end computational resources, without which the thesis would have been very difficult.

In addition, I also acknowledge the useful thesis and Computer Vision related discussions with my lab fellows, i.e., Muhammad Ehsan ul Haq, Muhammad Anwaar Khalid, Atif Khurshid, and Usama Aleem Shami.

Muhammad Aasharib Nawshad

Contents

1	Introduction	1
1.1	Background	1
1.1.1	Biometric Verification Systems	1
1.1.2	Face Recognition	2
1.1.3	Face Recognition vs Face Verification	3
1.1.4	COVID-19 Impact on biometric systems	3
1.2	Motivation	4
1.3	Problem Statement	4
1.4	Research Gap	5
1.5	Solution Statement	5
1.6	Contributions	6
1.6.1	Overview of proposed pipeline	6
2	Literature Review	10
2.1	Datasets	10
2.2	Face Recognition Evolution	11
2.3	Masked Face Recognition	13
2.3.1	Masked Face Recognition Methods	13
3	Methodology	20
3.1	Dataset	20

CONTENTS

3.1.1	Training Dataset	21
3.1.2	Testing Dataset	22
3.2	System Architecture workflow	22
3.3	Face Detection	22
3.3.1	Architecture	24
3.4	Mask Detection	25
3.4.1	Dataset	27
3.4.2	Architecture	27
3.5	Feature extraction using pretrained face recognition backbone	27
3.5.1	Resnet-101	28
3.5.2	MobileFaceNet	28
3.6	Masked Face Unveiling Model (MFUM)	29
3.6.1	Masked Face Unveiling Dense Residual Unit (MFU-DRU)	32
3.6.2	Masked Face Unveiling Attention Augmented Dense Residual Unit (MFU-ADRU)	33
3.6.3	Face Unveiling Conv Unit (MFU-CU)	34
3.6.4	Masked Face Unveiling Conv Unit with Attention Mechanism (MFU-CUAM1)	34
3.7	Training	35
3.7.1	Loss Functions	35
3.7.2	Model Training settings	37
3.8	Evaluation	37
3.8.1	Experimental Settings	37
3.8.2	Evaluation Metric	39
3.9	Working Demo Application	41
3.9.1	Verification Portal	41
3.9.2	Recognition Portal	42

4	Results	48
4.1	Impact of face mask on face recognition performance	48
4.1.1	UMR-MP scenario with flattened embeddings	48
4.1.2	MR-MP scenario with flattened embeddings	49
4.1.3	UMR-MP scenario with intermediate feature-map embeddings	49
4.1.4	MR-MP scenario with intermediate feature-map embeddings	50
4.1.5	Comparison of face mask effect on FNMR and FMR	50
4.1.6	Comparison of UMR-MP and MR-MP scenarios	51
4.1.7	Comparison of backbones performance	51
4.1.8	Summary	51
4.2	MFUM impact on masked face recognition performance	51
4.2.1	Impact of MFUM on the separability of genuine and imposter scores	52
4.2.2	Impact of MFUM's different architectures on masked face recognition performance	52
4.3	Comparison with State of the Art	61
4.4	Qualitative Results	90
4.5	Processing Time of MFUM Architectures	90
4.6	Mask Detector architecture	92
5	Discussion	93
5.1	Degradation in Masked Face Recognition Performance	93
5.2	Larger impact of face mask on FNMR as compared to FMR	93
5.3	Resnet-101 verification performance better as compared to MobileFacenet	94
5.4	LFW performance better as compared to MFR2	94
5.5	MFUM architectures comparison	94
5.5.1	Biometric evaluation metrics	95
5.5.2	Memory Requirement	95
5.5.3	Processing time	96

CONTENTS

5.6	Loss functions comparison	96
5.7	MFUM trained on Augmented MS1MV2 dataset vs trained on MFR2 dataset	97
5.8	The advantage of MFUM over retraining-based approach	98
5.9	Mask face detector architecture performance with transfer learning	98
6	Conclusion	99
6.1	Future Work	100
7	Publications arose from the thesis	

List of Abbreviations and Symbols

Abbreviations

ANN	Artificial Neural Network
BoF	Bag of Feature
COVID	COronaVirus Disease
CV	Computer Vision
CNN	Convolutional Neural Networks
DL	Deep Learning
FP	False Positive
FCN	Fully Convolutional Network
GAN	Generative Adversarial Networks
LFW	Labeled Faces in the Wild
NIST	National Institute of Standards and Technology
NN	Neural Networks
NMS	Non Maxima Suppression
MFUM	Masked Face Unveiling Model
MFU-DRU	Masked Face Unveiling Dense Residual Unit

CONTENTS

MFU-ADRU	Masked Face Unveiling Attention Augmented Dense Residual Unit
MFU-CU	Masked Face Unveiling Conv Unit
MFU-CUAM1	Masked Face Unveiling Conv Unit with Attention Mechanism in configuration 1
MFU-CUAM2	Masked Face Unveiling Conv Unit with Attention Mechanism in configuration 2
MFU-CUAM3	Masked Face Unveiling Conv Unit with Attention Mechanism in configuration 3
MFU-CUAM4	Masked Face Unveiling Conv Unit with Attention Mechanism in configuration 4
MLP	Multi Layer Perceptron
PCA	Principal Component Analysis
PDSN	Pairwise Differential Siamese Network
RMFRD	Real World Masked Face Recognition Dataset
SOP	Standard Operating Procedure
SoTA	State of The Art
SMFRD	Simulated Masked Face Recognition Dataset
SVM	Support Vector Machine
TP	True Positive
UMR-UMP	Unmasked Reference Unmasked Probe
UMR-MP	Unmasked Reference Masked Probe
MR-UMP	Masked Reference Masked Probe

List of Figures

1.1	Proposed pipeline for face verification.	8
3.1	Mask Augmentation methods.	21
3.2	Workflow of the complete proposed system.	23
3.3	P-Net architecture diagram.	25
3.4	R-Net architecture diagram.	26
3.5	O-Net architecture diagram.	26
3.6	Bottleneck layer architecture in MobileFaceNet.	29
3.7	Overall proposed MFUM architecture diagram workflow along with MFU-ADRU architecture.	30
3.8	Alternate MFUM architectures.	31
3.9	Verification Portal Flow Chart.	44
3.10	Face verification portal shows whether two given images are of the same entity along with mask status, cosine similarity score, and time taken by face detection, mask detection, and backbone embedding generation.	45
3.11	Recognition Portal Flow Chart.	46
3.12	Face Recognition portal showing the name of the recognized entity along with mask status and cosine similarity score.	47
3.13	Face Recognition portal showing the name of the recognized entity along with mask status and cosine similarity score.	47

LIST OF FIGURES

4.1	Qualitative Results for UMR-UMP scenario. The green box around the image shows correct prediction whereas the red box around the image show incorrect prediction using our approach.	90
4.2	Qualitative Results for MR-UMP scenario. The green box around the image shows correct prediction whereas the red box around the image show incorrect prediction using our approach.	91

List of Tables

1.1	Brief comparison of existing biometric systems	2
2.1	List of Available Datasets for face recognition	12
3.1	Architecture of Resnet101.	28
3.2	Architecture of MobileFaceNet.	32
3.3	Experimental Settings.	37
3.4	Experimental Settings Abbreviation and their elaboration.	40
4.1	The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-ADRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-ADRU unit trained with Quadruplet loss approach.	59

4.2	The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-ADRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-ADRU unit trained with the Quadruplet loss approach.	60
4.3	Comparison of our approaches with SoTA approaches for Unmasked Reference and Masked Probe experimental setting. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.	61
4.4	Comparison of our approaches with SoTA approaches for Masked Reference and Masked Probe experimental setting. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.	62
4.5	Comparison of our approaches with SoTA approaches for Unmasked Reference and Masked Probe experimental setting on MFR2 dataset. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.	62
4.6	Comparison of our approaches with SoTA approaches for Masked Reference and Masked Probe experimental setting on MFR2 dataset. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.	63

4.7 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-DRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-DRU unit trained with the Quadruplet loss approach. 64

4.8 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-DRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-DRU unit trained with Quadruplet loss approach. 65

4.9 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 66

4.10 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 67

4.11 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM1 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 68

4.12 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM1 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 69

4.13 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM2 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 70

4.14 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM2 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 71

4.15 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM3 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 72

4.16 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM3 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 73

4.17 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM4 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 74

4.18 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM4 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 75

4.19 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-DRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 76

4.20 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-DRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 77

4.21 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-ADRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 78

4.22 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-ADRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 79

4.23 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 80

4.24 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 81

4.25 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM1 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 82

4.26 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM1 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 83

4.27 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM2 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 84

4.28 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM2 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 85

4.29 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM3 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 86

4.30 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM3 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 87

4.31 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM4 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 88

4.32 The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM4 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. 89

4.33 Running time of different MFUM architectures in second for Resnet-101 backbone. 92

Abstract

With the outbreak of COVID-19, people worldwide started wearing face masks to cover their mouths and noses to avoid the negative effects of the pandemic. The conventional face biometrics systems were not designed to handle masked faces. Some facial features like the nose and mouth get hidden under the mask, resulting in performance degradation in face biometrics systems. Several studies also reported this degradation in face biometric systems performance when a mask is worn. In addition, several methods involving complex and computationally expensive deep learning models proposed for solving masked face recognition problems had downgraded performance in unmasked face recognition settings. Therefore, there was a need for a biometric face recognition system that could not only recognize faces with good performance in masked scenarios but has at least the same performance as state-of-the-art in unmasked scenarios. This thesis proposes the Masked Face Unveiling Model (MFUM) to cope with this problem. The MFUM works on top of existing face recognition models and is built on the concept that facial embeddings get corrupted for masked faces. This model makes masked facial embeddings of a person similar to unmasked facial embeddings of the same person and different from unmasked facial embeddings of other persons. Different ablation studies have been conducted using face recognition model backbones, MFUM architectures, and loss functions. Results are evaluated on the LFW dataset with synthetic masks and a real-world masked face recognition dataset, i.e., MFR2. We have reported the performance of the proposed model in different unmasked and masked settings using evaluation metrics adopted globally for reporting the performance of biometrics systems, i.e., Equal Error Rate, False Match Rate, and Fisher discriminant ratio. The results show that the MFUM having Face Unveiling Attention Augmented Dense Residual Unit architecture and trained using Quadruplet loss has outperformed state-of-the-art methods in addition to other evaluated MFUM architectures with different losses.

CHAPTER 1

Introduction

The field of Computer Vision has advanced significantly in the last decade, particularly after the use of Deep Learning based techniques. These Deep Learning based methods have outperformed the conventional Machine Learning based methods in the applications like Image Classification, Object Detection, Instance Segmentation, Semantic Segmentation, Object Tracking, Action Recognition, and 3D reconstruction. Face Recognition is one of the most prominent use cases of computer vision which is the combination of image classification and object detection techniques. It is a biometric technology that identifies a person through his/her face image. In this section, we shall look into the background, significance, and challenges of face recognition systems, particularly in the COVID-19 scenario. Further, we shall discuss the research gaps and provide a brief overview of our contribution to the challenges.

1.1 Background

1.1.1 Biometric Verification Systems

Biometrics verification systems have been in use for quite a long time. Facial Recognition, Iris Recognition, Fingerprint recognition, and Gait based verification systems are some examples of biometric verification systems. There are two major categories of biometric systems, i.e., the one which requires contact with the sensor and the other one which does not require contact with the sensor (contactless). Fingerprint verification systems are an example of biometric systems which require contact with the sensor, whereas facial recognition systems, Iris recognition systems, and Gait-based verifica-

tion systems are an example of contactless biometric systems. These biometric systems require special sensors to work; for example, fingerprint recognition systems require fingerprint sensors, Iris recognition systems require iris scanners, and Gait recognition systems require motion sensors and video cameras. These sensors are very expensive compared to the sensor required by a face recognition system, i.e., a camera. In addition to this, the accuracy of face recognition systems is higher as compared to the accuracy of other biometric systems. Therefore among all available biometric systems, face recognition is the most popular and widely used biometric technique due to its efficiency and cost-effectiveness. A brief comparison of these biometric systems is given in Table 1.1.

Table 1.1: Brief comparison of existing biometric systems

Biometric System	Nature	Accuracy	Limitations
Finger Print Recognition	Contact	98.60%	Suffer with skin cuts and other deterioration problems.
Iris recognition	Contact Less	90-99%	Suffer due to eye color and race variation. Person must be very close to sensors.
Gait Recognition	Contact Less	98.70%	Lots of initial data requirement for each person along with specialized hardware requirement.
Face Recognition	Contact Less	99.97%	Suffer with pose, illumination and occlusion.

1.1.2 Face Recognition

Face Recognition technology has been in use for more than 50 years. It is a biometric technique that identifies people with the help of facial images. Before the deep learning era, most of the algorithms were based upon mathematical approaches like Principal Component Analysis (PCA) and then local feature and shallow feature learning-based approaches. Then with the rise of Deep Learning (DL), the focus of research in facial

recognition shifted toward DL. These DL-based techniques produced state-of-the-art (SoTA) results resulting in their applications in a variety of areas, for example, surveillance systems, attendance systems, transportation ticket verification systems, border management systems, etc. The details of various Face Recognition techniques are shared in the literature review section.

1.1.3 Face Recognition vs Face Verification

The terms face recognition and face verification are often interchangeably used. But in literature, there is a difference between the two. The main difference between these two terms is the type of matching. Face recognition is termed as one-to-many matching whereas face verification is termed as one-to-one matching. In face recognition, a probe image is compared with many images to find the exact person. Whereas in face verification, a probe image is compared with a single reference image to output whether these two images are of same identity or not.

1.1.4 COVID-19 Impact on biometric systems

The first case of Coronavirus disease (COVID-19) caused by the SARS-CoV-2 virus was reported in December 2019. As a result, the globe was confronted with a catastrophic disaster in the form of a global pandemic. The pandemic caused a global socio-economic crisis that affected every single individual, organization, and country. Researchers found out that the disease spread due to close proximity to the infected patient. When people breathe, the contaminated droplets which contain the virus go to the lungs of a healthy person resulting in infection. To cope with the afflictive effect of the pandemic Standard Operating Procedures (SOPs) were issued to be followed by everyone. Covering the face, particularly the nose and mouth, was the most prominent and effective SOP of all SOPs. To comply with this SOP, people started wearing face masks. In addition, it was also mentioned in the SOPs that a person should not touch their face and facial mask. Further, it was also advised to reduce the points of common contact. Due to this, the biometrics systems like fingerprint recognition systems requiring physical contact for working were left useless due to the higher risk of the spread of the disease. At that time, there was a need for a contactless biometric verification system. Therefore, the demand for contactless biometric verification systems increased. Face Recognition technology,

the most popular, economical, and widely used contactless biometric technique, was used extensively as an alternative to fingerprint-based verification systems.

1.2 Motivation

Before COVID-19, Face Recognition techniques were very mature in non-challenging scenarios like good illumination and no occlusion. But when it came to challenging scenarios like occlusion, illumination, and pose, conventional facial recognition systems' performance got badly affected. The same situation happened during the COVID-19 pandemic when people were wearing facial masks to cover their mouths and noses to avoid the afflictive effect of the pandemic. By wearing a face mask, the important facial features of the nose and mouth region get hidden under the mask, which makes facial recognition challenging not only for machines but also humans. According to the National Institute of Standards and Technology (NIST) 2020 report, the facial recognition systems' performance has a 20-50% error due to masked faces [36]. Therefore, in order for facial recognition systems to work effectively, people must pull off the mask to get themselves verified. Still, at the same time, there is a risk of getting the infection during that period. In addition to this, touching the face and mask is highly discouraged as per the SOPs of COVID-19. Also, in the surveillance systems, each and every individual cannot be forced to remove the facial mask. Therefore, keeping these facts in view, there is a dire need to improve the existing facial recognition techniques to cater to the problems created by masked faces. Currently, facial recognition with masked faces is an under-researched area, and many research efforts are underway. Most of the solutions proposed till now have two major limitations in addition to the performance in masked face scenarios. These limitations are high computational requirements and deterioration in performance in unmasked face scenarios.

1.3 Problem Statement

"The performance of existing face recognition systems have been thwarted by the use of facial masks and the solutions presented so far are either less accurate in some unmasked scenario or are computationally expensive. Therefore, there is a dire need of an effective, robust, and computationally inexpensive facial recognition system which

recognizes people with facial masks effectively and accurately without deteriorating the performance in unmasked scenario."

1.4 Research Gap

Most face recognition algorithms are developed for working under controlled environments in which there is no occlusion on the face and good illumination. Therefore when faces are covered with the mask, the mask acts as occlusion and performance of these algorithms deteriorates. Basically, by wearing a face mask, the facial features get corrupted, and when these corrupted facial features are processed by conventional face recognition algorithms, the resulting embeddings are corrupted; hence the performance is affected. Therefore there is a need for the development of those models which handle the corrupted facial features and improve the performance of the model on masked faces.

Further, the solutions developed for improving the masked face recognition performance use combination of masked and unmasked images for training the existing models, but there are two major drawbacks of this approach. The first one is several studies have reported deterioration in the performance of the existing SoTA model in unmasked face scenarios. At the same time, there is a slight improvement in the masked face recognition scenario. And the second one is there is very high computational power, resource, and data requirement for training such large deep learning models.

In addition to this, the performance of deep learning models relies heavily upon the dataset which is used for training these models. The deep learning model tries to learn the generalization of the provided training data. Therefore, there is a requirement for a good reasonable-sized dataset that has almost all the variations of real-life data. Unfortunately, we do not have any reasonably large-sized dataset which has both unmasked and masked images of persons required for training of face recognition model. Therefore there is a need for some augmentation-based method for generating masked faces data.

1.5 Solution Statement

"To create a robust, effective and computationally inexpensive facial recognition system for masked face scenario without affecting the performance on unmasked faces."

1.6 Contributions

In this thesis, we have looked for the possible approaches which can improve the masked face recognition performance. Following are the main contributions of this thesis.

1. The Masked Face Unveiling Model (MFUM), a novel and computationally inexpensive masked face recognition technique, is suggested.
2. The MFUM increases the reusability of existing facial recognition models in masked face scenarios without the need for retraining.
3. The MFUM increases the similarity of the same identity's corrupted masked face embedding and unmasked facial embedding. Furthermore, it reduces the similarity between the unmasked face embedding of an identity and the corrupted facial embedding of another identity.
4. The proposed method does not deteriorate the performance of an existing model in an unmasked face recognition scenario and operates only on masked face images.
5. A complete face verification pipeline for masked as well as unmasked faces is developed for the qualitative evaluation of the proposed approach.
6. Extensive ablation studies have been conducted using various face recognition model backbones, MFUM architectures, and loss functions.
7. Evaluated the results upon augmented masked face dataset, i.e., LFW, and real-world masked face recognition dataset, i.e., MFR2, using the biometric evaluation metrics recognized globally for reporting the performance of biometrics systems.
8. Achieved better results compared to SoTA face recognition models.

1.6.1 Overview of proposed pipeline

Now lets, look into the brief overview of MFUM which processes the corrupted masked face embedding. The next subsections present a brief overview of MFUM which processes the corrupted masked face embedding. In addition to this, the working of the complete face verification pipeline and its key components is also discussed briefly. The overall proposed pipeline is given in Figure 1.1.

Face Detector

The first pre-requisite of our proposed pipeline is that input images must have faces. Therefore we have used the MTCNN face detector for detecting and cropping only relevant face areas to be fed to the mask detector and masked face unveiling model.

Mask Face Detector

The working of our proposed masked face unveiling model is highly dependent upon the provision of only masked face embeddings. When the user of our system provides two input images for the verification task, either none or both can be masked or unmasked. Therefore we have to find whether the person in the input image is wearing a mask or not and then process only the masked image embedding with MFUM. For this task, we have developed a mask face detector that indicates whether a person is wearing a mask. The details of mask face detector implementation are given in Chapter 3.

Masked Face Unveiling Model

Our proposed approach is constructed on top of the existing face recognition model; therefore, it does not require any modification or retraining existing face recognition model. We name our model as Masked Face Unveiling Model (MFUM). The MFUM operates upon embedding space. It takes masked face embeddings generated from the face recognition model as input and outputs new facial embeddings similar to the unmasked facial embedding of the same person. In addition, the new facial embeddings generated by our proposed MFUM are dissimilar from the facial embeddings of different persons. For training MFUM, we have used quadruplet loss which guides the MFUM to train better compared to currently available state-of-the-art loss functions like triplet loss and self restrained triplet loss. Quadruplet loss has previously been used in Person Re-identification problems. We have used it in our scenario, which is a novel contribution. Quadruplet loss takes two negative examples in addition to a positive example and anchor. This loss function tries to minimize the distance between the anchor and the positive example and maximize the distance between the anchor and negative example 1 and negative example 2. The details of the Masked Face Unveiling Model are discussed in detail in Chapter 3.

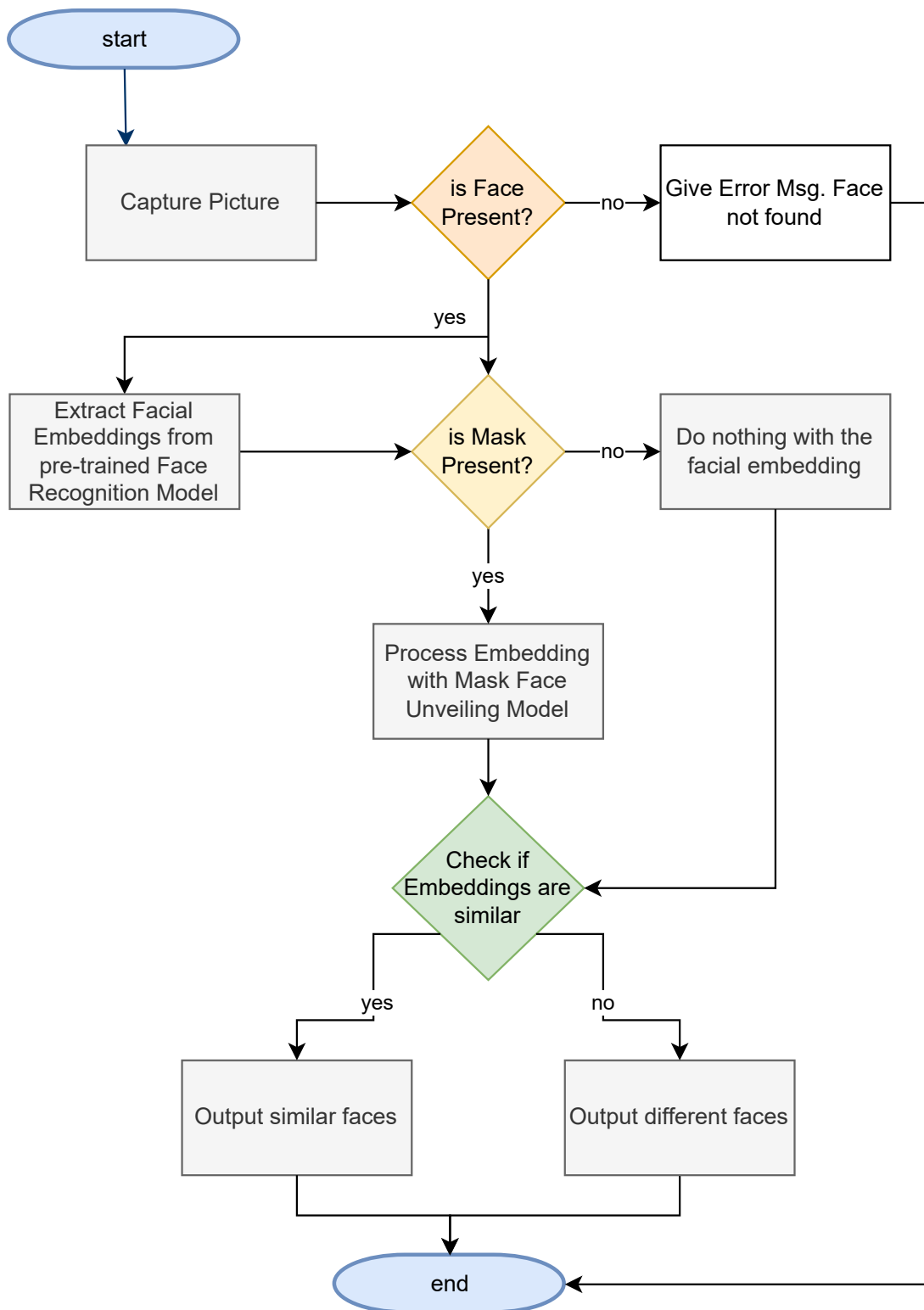


Figure 1.1: Proposed pipeline for face verification.

The rest of the thesis is organized as follows: The literature relevant to the problem under discussion is discussed in Chapter 2. The methodology is then explained in Chapter 3, with each component of the solution pipeline being detailed in greater depth. The results of our system are discussed in Chapter 4 along with ablation studies. The results of our research are discussed in Chapter 5. In this chapter, we have also analyzed why certain things work and others do not. The thesis is concluded in Chapter 6 to bring it to a close.

Literature Review

In this section, we shall first look into the details of the available dataset for the problem under consideration. Then we shall look into the evolution of the face recognition research domain over the years. Then we shall look into details of different methodologies found in the literature to solve the masked face recognition problem.

2.1 Datasets

For any deep learning problem, the dataset is extremely important and is the foundation for good training of the deep learning model. The necessary condition for a good dataset is that it covers all the real-life scenarios. But practically, it isn't easy to get such an ideal dataset. Therefore, researchers try to acquire a dataset that best represents all the variations of real-life scenarios. For the field of face recognition, many datasets, i.e., Labeled Faces in the Wild (LFW) [11], CASIA-WebFace [15], VGGFace2 [28], MSCeleb1M [19], WebFace260M [58] etc. have been developed which deal with different challenges of facial recognition like recognition in the wild, pose, illumination, occlusion, race, and gender. The summary of the list of available datasets is given in Table 2.1. But for our problem of masked face recognition, there is no large dataset that contains masked as well as unmasked images of the same entity. There are only two datasets, i.e., Real World Masked Face Dataset (RMFRD) [37] and Masked Face Recognition (MFR2) [34], which have both masked and unmasked images. The limitation of RMFRD is that it does not have a masked image for every identity. Further, the ratio of masked images to unmasked images of the same person is very small, as there are almost 90,000 un-

masked and 5,000 masked images of 525 identities. The limitation of the MFR2 dataset is that it is a very small dataset containing 53 identities and 269 masked and unmasked images. But the dataset is quite balanced between the count of unmasked and masked images. Therefore due to the limitations mentioned above, researchers have developed mask augmentation techniques for generating artificial mask datasets from unmasked face recognition datasets [34], and [36]. Anwar et al. method has produced a Simulated Masked Face Recognition Dataset (SMFRD), which contains almost 500,000 face images of 10,000 persons. They have generated this data by augmenting mask on the combination of LFW [11], and CASIA-Webface [15] images. This dataset has an almost equal number of masked and unmasked images. They have also reported 95% accuracy for facial recognition using these datasets by applying attention weights to key features in the non-occluded part. They have also published a Real-world Masked Face verification dataset containing 4015 facial images of 426 identities. Further, they have also shared 7178 unmasked and masked pairs having 3589 pairs each of the same and different identities [34]. The authors have not shared much implementation and architecture details for the evaluation of datasets.

2.2 Face Recognition Evolution

Facial recognition is a very well-researched problem. It has been in use for more than 50 years. It gained popularity after the proposal of the eigenface approach by Turk et al. [1], in the early 1990s. Then for more than 20 years, local-features approaches along with shallow feature learning based approaches were used [3], [5], [7], [8], [10] until the emergence of Deep Learning based approaches like DeepFace by Taigman et al. [56] and Deep Face Recognition by Parkhi et al. [17]. After these benchmarks setting approaches, the focus of researchers shifted towards deep learning based approaches and, more particularly, CNNs [25], [53]. The CNNs resulted in the emergence of state-of-the-art approaches like CosFace by Wang et al. [57] and ArcFace by Deng et al. [49]. These approaches have achieved over 99.5% accuracy on popular face datasets like Labeled Faces in the Wild (LFW) [11], and CASIA-WebFace [15]. In deep learning, for vision tasks, mostly CNNs have been used. Still, with the emergence of vision transformers, recently, much research has been carried out to test the performance of vision transformers in the same scenarios where CNNs were previously applied. In some

Table 2.1: List of Available Datasets for face recognition

Dataset Name	Year	# Entities	# Images	Masks
COLOR-FERET Facial image [2]	1997	994	11338	No
AR Face Database [4]	1998	126	4000	No
Yale FaceDatabase B [6]	2001	37	2516	No
Extend Yela DB [9]	2005	28	16128	No
LFW [11]	2008	5749	13,233	No
Youtube Faces [13]	2011	1,595	3425 videos	No
FaceScrub [14]	2014	695	141,130	No
CASIA-WebFace [15]	2014	10,575	494,414	No
IJB-A [16]	2015	500	25,809	No
MegaFace [21]	2016	4,030	4,400,000	No
CFP [22]	2016	500	7000	No
AgeDB [24]	2017	568	16,488	No
CALFW [27]	2017	4025	12,174	No
CPLFW [33]	2017	3968	11,652	No
IJB-B [26]	2017	1845	11,754 images and 7,011 videos	No
MS1MV2 [49]	2018	85000	5.8Million	No
VGGFace2 [28]	2018	330	9500	No
IJB-C [31]	2018	3,531	31,300 images and 11,779 videos	No
Celeb A [30]	2018	10,177	200,000	No
MFR2 [34]	2020	53	269	Yes (Real)
SMFRD [37]	2020	10000	500,000	Yes (Sim- u- lated)
RMFRD [37]	2020	525	5000 masked, 90000 unmasked	Yes (Real)
RWOC [39]	2021	180	3195 unmasked, 678 masked	Yes (Real)
WebFace42M [58]	2021	2 Million	42 Million	No

cases, the vision transformers have outperformed CNNs. Zhong et al. [43] have used vision transformers in face recognition, and the results are quite encouraging to support research in this domain. They have reported good performance of vision transformers with the condition of availability of a large amount of face data. Also, they have reported poor performance of vision transformers on occluded faces. Therefore, considering the results of Zhong et al. [43], we have kept our research direction limited to convolutional neural network based approaches due to our problem domain being specific to occlusion scenarios.

2.3 Masked Face Recognition

With the outbreak of COVID-19, people wear face masks to cover a large portion of the face, including mouth and nose regions. This creates the problem of occlusion for existing facial recognition based systems. Therefore, in these special circumstances, only an occlusion robust system shall work. According to the NIST 2020 report, the facial recognition systems' performance has 20-50% error due to masked faces [36]. Also, Damer et al. [48] have evaluated the performance of three facial recognition systems with masked faces and reported that masks have a very high impact on genuine pair decisions compared to imposter pair decisions. According to Li et al. [41], there are three major challenges associated with masked face recognition. The first one being there is a lack of masked face recognition datasets. Second, the mouth and nose features are severely damaged and useful features that can discriminate between different classes are not available to the model. Third, detecting whether a person has worn a face mask is hard due to the mask's different colors, styles, designs, and patterns.

2.3.1 Masked Face Recognition Methods

For the problem of corrupted facial features, researchers have developed many different methodologies. We shall discuss each methodology along with the dataset used, results, and limitations.

Cropping Based Methods

Hariri et al. [45] proposed removing the occluded portion of the face, i.e., nose and mouth regions, and then applying pre-trained deep convolutional neural networks like AlexNet, VGG-16, and ResNet-50 to extract the embeddings. Then the Bag of Feature (BoF) paradigm is applied to quantize the embeddings and get a representation of a fully connected layer of classical CNN. In the end, Multi-Layer Perceptron (MLP) is applied for the classification process. They have achieved the highest accuracy of 91.3% using Resnet-50 and BoF approach on the RMFRD dataset and 88.9% on SMFRD.

Mundial et al. [52] proposed a similar solution to Hariri et al. but added face detection in pre-processing steps to find relevant facial regions accurately. After this step, 128 Dimensional embeddings are extracted using convolutional neural networks (CNN), and Support Vector Machine (SVM) classifier is applied to classify the input image. They have trained the model on VGGFace2 [28] dataset and evaluated on LFW [11], and RMFRD datasets [37]. The facial recognition accuracy of 97% was reported on RMFRD and 98% on the LFW dataset.

Li et al. [41] have proposed a combination of attention mechanism and cropping-based approach to focus only on the non-occluded region of the face. In this way, the occluded region of the face is discarded and the network is trained to predict only using the non-occluded region. They have used SMFRD [34], and CASIA-Webface [15] for finding the optimal cropping size and selecting the attention module. Whereas AR[4] and Extend Yela B [9] datasets have been used for verification of the attention mechanism. The highest accuracy of 90% is achieved using this approach. Further, as the number of entities to be identified is increased above 700, the recognition accuracy falls down to almost 82.86%.

Song et al. [55] proposed using a Pairwise Differential Siamese Network (PDSN) to learn the correspondence between corrupted facial feature elements and occluded facial blocks. This PDSN act as an attention mechanism to focus on the relevant non-occluded facial region. They have trained the model on CASIA-Webface dataset [15] and evaluated the model on MegaFace [21], and AR [4] datasets. This paper reports an accuracy of 98.2% on the AR dataset and 74.4% on the MegaFace Challenge with partial occlusion. As the model is tested on different occlusion types and not particularly the masks, therefore exact accuracy for masked face recognition has yet to be discovered.

The major problem with cropping-based approaches is the loss of important facial structure information due to cropping of the occluded portion. Facial structure, for example, the shape of the face, i.e., elongated or round face and jawline, are often visible even after wearing face masks. Therefore, these parts of the face have important discriminatory features that can help facial recognition algorithms make correct decisions. But when the occluded region is cropped, the important facial structure information is lost completely, and we are left with very few discriminatory features of the eye and forehead region. Therefore, we need a solution that retains the essential facial features which could be preserved otherwise. In our research, we have adopted the same non-discarding methodology. Instead our model takes full face image corrupted embeddings and enhances its similarity to the embeddings of unmasked image of same identity and reduces the similarity from unmasked face embeddings of different identities.

Reconstruction Based Methods

Malakar et al. [51] have used Principal Component Analysis (PCA) and deep learning to solve the problem of occlusion in face recognition. PCA reconstructs the destroyed or occluded facial features, and deep learning is then used to identify the person. This paper uses Yale Face Database B [6] for training. Further, they have used a masked version of Yale Face Database B for evaluation. They have added artificial masks but have not discussed its methodology. Yale Face Database B contains 37 subjects and 68 images of each subject. Therefore, the dataset used for training and validation is not large. Also, the accuracy achieved is between 85 to 95%, which could be more impressive for a dataset of this size. However, a 15% increase in accuracy is reported compared to using deep learning based method alone.

Din et al. [35] have proposed a novel Generative Adversarial Networks (GAN) based approach for unmasking the masked facial image. They have used two-stage architecture. The first stage detects the face mask by generating the binary segmentation mask for the masked region, and the second stage performs masked region reconstruction using Generative Adversarial Networks (GAN). They have used CelebA dataset [30] for training and some real-world images collected from the internet for testing the model. They have achieved state-of-the-art performance both qualitatively and quantitatively. This method is only designed for the medical face mask and is incapable of automati-

cally detecting and removing multiple types of complex objects. Due to this reason, the network outputs a partially occluded image in case of small occlusion present in front of the face.

Jabbar et al. [46] have proposed improvements to the Din et al. method and made detecting and removing various types, sizes and colors of objects automatic. They have used the same two-stage architecture and CelebA dataset for training and achieved excellent qualitative and quantitative results upon different real-world image collections. Also, this method performs well over objects of different types, sizes and colors.

The main problem with reconstruction-based approaches is their limitation in reconstructing unknown identities, which means these approaches cannot be generalized well. These approaches cannot precisely reconstruct the facial features of identities absent in the gallery. Further, an increase in the gallery results in greater complexity of the solution. Therefore reconstruction-based approaches are computationally expensive. In addition, as most of these approaches are GAN based and these approaches are computationally expensive for both training and inference. In contrast, our proposed method is not only applicable to unknown identities and produces good results but it is computationally light-weight also.

Training with Combination of mask and unmask images methods

Montero et al. [42] have proposed training the ArcFace with ResNet-50 backbone with slight modification in the architecture. A dense layer is added parallel to the last layer, which generates the feature vector after the dropout layer. The purpose of adding this dense layer is to make the network learn whether the face mask has been worn or not. In this way, the network learns not only facial recognition but also facial mask detection, and combined output results in better facial recognition. They used a masked augmented MS1MV2 dataset for training. And for the mask augmentation, they have used Anwar et al. [34] mask augmentation technique. For result evaluation, they have used masked augmented LFW [11], CFP [22], and Agedb [24] datasets. And for evaluating real mask face scenarios, they have used MFR2 [34]. They have reported masked face recognition accuracy of 95.62%.

Sharma et al. [54] proposed the transfer learning based approach for generating the embeddings. First, the model is retrained using existing FaceNet model weights with

masked and unmasked images using the triplet loss function. During retraining, different experiments were performed with different combinations of mask and unmasked images. Then 128 Dimensional embeddings are generated using the retrained model. The training accuracy reported is 99%, and test accuracy is around 50% which clearly shows that model is overfitting.

Maharani et al. [50] proposed a transfer learning based approach for real-time masked face recognition using cosine distance. In the first stage, the face is detected using Haarcascade or MobileNet face detectors and then in the second stage face is recognized using the models, which are transfer learned either from VGG16 ImageNet or Triplet Loss FaceNet. This fine-tuning is done using a new custom dataset. This dataset has images of three entities, with 200 images for each entity. Then these two backbones, i.e. VGG16 and FaceNet, are used for finding 128D embeddings of the cropped face. Next, in parallel, Cosine distance is used to compare the last two frames to find the bounding box, and it can report whether the person has changed in the current frame compared to the previous frame. The dataset used for training is too small, i.e. just three entities. The model may suffer with large numbers of identities to be identified using this approach. Further, there is nothing particular done for handling masked faces and masked corrupted features are being used which lead to poor performance on masked faces if large number of identities are to be identified.

Muhi et al. [47] have proposed a transfer learning based deep learning method for masked face recognition. They have proposed a three-stage method. The first stage segments the mask region of the face. Then the second stage extracts the feature using Resnet50. The third stage performs classification with a softmax activation function. They have used the LFW dataset and mask-augmented version of LFW for training. Training is done using a combination of these two datasets and achieves a test accuracy of 98%. The dataset used for training is quite small, especially when we fine-tune the Resnet50 model.

Boutros et al. [44] have proposed a novel Self Restrained Triplet Loss to train an artificial neural network with 2 hidden layers to process the embeddings produced by the existing face recognition model. This neural network aims to uncorrupt the corrupted masked face embedding by learning the similarity between masked and unmasked images of the same person. They have used the MS1MV2 [49] dataset to train the artificial

neural network. They have evaluated the results on two real-world masked datasets, i.e., MFR2 [34] and MRF. They have also evaluated their model on two simulated masked face datasets, i.e., LFW [11] and IJB-C [31] datasets. They have achieved impressive performance on these four datasets.

Huber et al. [40] have proposed template-level knowledge distillation for mask-invariant face recognition. Their solutions utilize template-level knowledge distillation during the training phase to produce masked face embedding similar to the unmasked face embeddings of the same person whereas different from another person. The teacher model is trained on MS1MV2 [49] dataset. The teacher model has unmodified face images, whereas the student network has images with a 50% probability of synthetic masks of different colors and shapes. They have evaluated the model on two real-world masked face datasets, i.e., MFR2 [34] and MFRC21 [38] of MFR competition. They have also evaluated their model on five simulated mask datasets, i.e., LFW [11], CFP-FP [22], AgeDB-30 [24], CALFW [27], CPLFW [33]. They have achieved state-of-the-art performance on both real mask datasets as well as simulated mask datasets.

The limitation of the training with combination of mask and unmask images based approach category is deterioration in performance in unmasked face recognition scenarios. Basically, in these methods, existing models are trained with the combination of masked and unmasked images, which slightly improves the performance in the masked scenario. Still, on the other hand, the performance in the unmasked scenario deteriorates. Further, retraining a model is a computationally expensive task and more computational resources and time is required. Contrary to this, our approach does not impact the performance of existing models in the unmasked scenario and improves the performance in the masked scenario. Also, our model training is computationally inexpensive relative to retraining of existing face recognition models.

Miscellaneous

Anwar et al. [34] developed an open-source tool for generating the masked face dataset from the unmasked face dataset. They generated masked versions of existing popular face datasets like VGGFace2 [28], and LFW [11]. They also developed a small real-world mask dataset of 53 entities and 269 images named MFR2 to verify model performance on real-world data. They retrained state-of-the-art Facenet [18] on the custom-generated

VGGFace2-mini dataset (a subset of the VGGFace2 dataset) and its masked version. Then they evaluated the performance of the trained model on simulated LFW and MFR2 datasets. They achieved a 38% improvement in true positive rate for the existing Facenet system for unmasked as well as masked faces. In addition to this, they achieved excellent accuracy on the MFR2 dataset, proving the generalization capability of the simulated mask trained model on real-world data.

Summary

Summarizing the above mentioned masked face recognition methods there are mainly three approaches which have been used until now for masked face recognition scenarios.

1. Discarding the occluded facial region in face image and training the existing face recognition models on unoccluded regions. (Can be through use of attention mechanism)
2. Reconstructive approach for regenerating the occluded facial regions and then applying conventional face recognition algorithms.
3. Training with the combination of masked and unmasked faces. (include transfer learning based approaches).

The major shortcoming of these approaches is performance, efficiency, generalizability, and scalability. In our proposed work, we address the shortcomings of the reviewed methods and present a solution that has a good performance both in masked as well as unmasked scenarios, good efficiency in terms of computational power requirement and performance, generalization, and scalability to datasets with non-overlapping identities.

Methodology

In this section, we present our proposed methodology to improve the masked face verification performance of existing facial recognition models without the need to retrain them. We have built our solution on top of existing facial recognition models. This gives us an advantage because we do not need to retrain the existing models. We have developed our model to utilize the embeddings from existing face recognition models and process these embeddings to produce embeddings which are more similar to unmasked facial embeddings of the same person and different from unmasked facial embeddings of other persons. We shall discuss complete training and evaluation pipelines along with dataset augmentation techniques.

3.1 Dataset

The data is the first prerequisite for solving any deep learning task. As we have discussed in the literature review section, no large masked face recognition dataset is publicly available. This makes our problem harder to solve. Fortunately, we have mask augmentation techniques proposed by Anwar et al. [34], and Ngan et al. [36]. These techniques augment masks on unmasked facial images of a person to generate a corresponding masked face image. Now we shall look into the details of these techniques.

1. MaskTheFace [34]:

MaskTheFace [34] is a computer vision based tool for augmenting the mask on a person's face. It can generate over 100 different mask types. These masks are of different shapes and colors. It uses the python dlib library for detecting facial

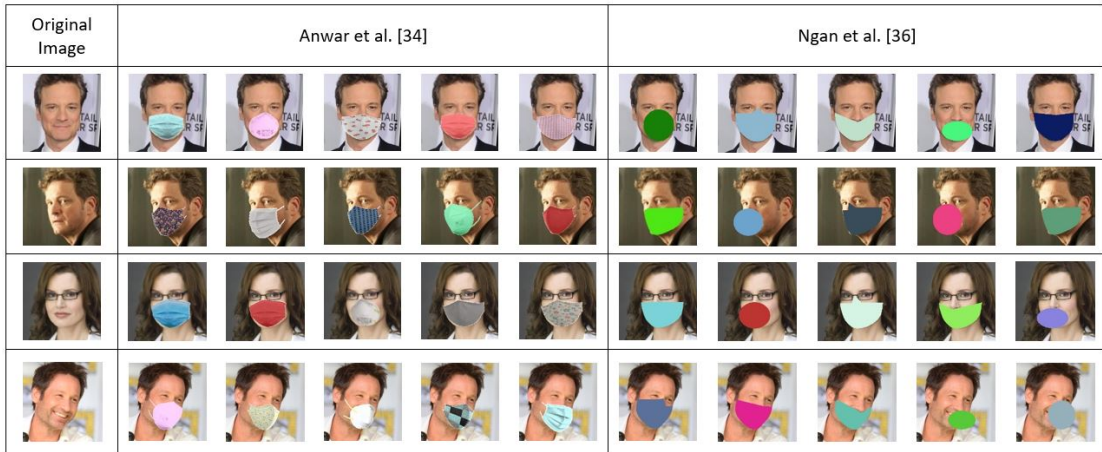


Figure 3.1: Mask Augmentation methods.

landmarks and six key features which are needed for augmenting the mask on a person’s face. The tool also handles the tilt of the mask to perfectly fit the tilted face. The results of mask augmentation are very similar to real-world scenarios. Some examples of face masks generated using this approach are given in Figure 3.1

2. Mask Augmentation Approach by Ngan et al. [36]

Ngan et al. [36] have developed an automated tool for augmenting the mask on a person’s face. The tool can generate 6 different shapes of masks with random colors. These generated masks differ in height and facial coverage. These masks are categorized as round-low coverage, wide-low coverage, round-medium coverage, wide-medium coverage, round-high coverage, and wide-high coverage. The tool uses the python dlib library to detect the face in the image and then extract 68 facial landmarks. Once facial landmarks are detected, masks of random colors are augmented on the unmasked face image. Figure 3.1 shows a few examples of masks generated using this approach.

3.1.1 Training Dataset

For training the model, we have used the MS1MV2 dataset [49]. The dataset has 5.8 million images of 85 thousand identities. We have generated a mask version of the dataset using the mask augmentation approach as proposed by Anwar et al. [34]. This augmentation approach failed to extract facial landmarks of almost 429,000 images due to the limitation of the dlib library; thus, masks could not be augmented on these

images. Therefore, these unmasked images were discarded from the training batches.

3.1.2 Testing Dataset

We have used two evaluation datasets to test our model’s performance. The first one is the simulated mask dataset developed by augmenting mask on LFW dataset [11] using mask augmentation approach by Ngan et al. [36]. The second is the real-world mask face recognition dataset, i.e. MFR2 [34].

3.2 System Architecture workflow

First, we look into the proposed methodology’s workflow to understand the complete system. First, an image is captured using a camera. Then a face detector detects whether a face is present in the image. If no face is found in the image, the system has nothing to process and exits. If a face is found, facial embeddings are extracted using pretrained face recognition models, and we also check whether the person in the image is wearing a mask or not using a mask detector. If no mask is present, we do not need to process the image with Mask Face Unveiling Model (MFUM). If a mask is worn, then we process the face embeddings using MFUM to produce enhanced embeddings of mask faces which are similar to the unmasked facial embeddings of the same person and dissimilar from unmasked facial embeddings of a different person. Once this process is complete, we compare the embeddings of one image to another to check whether these images are of the same person. This comparison is made using cosine similarity metrics. Figure 3.2 contains the workflow of the complete system.

3.3 Face Detection

Face detection is used to detect whether a face is present in the input image. It not only tells how many faces are present in the image but also gives the bounding boxes coordinates for the faces. We have used a light weight, but accurate face detector so that our system not only performs better but also the predictions are quick. This is why we have used Multi-Task Cascaded CNN (MTCNN) [23] for face detection. Further, weights of pretrained MTCNN are available publicly, and we do not need to train the

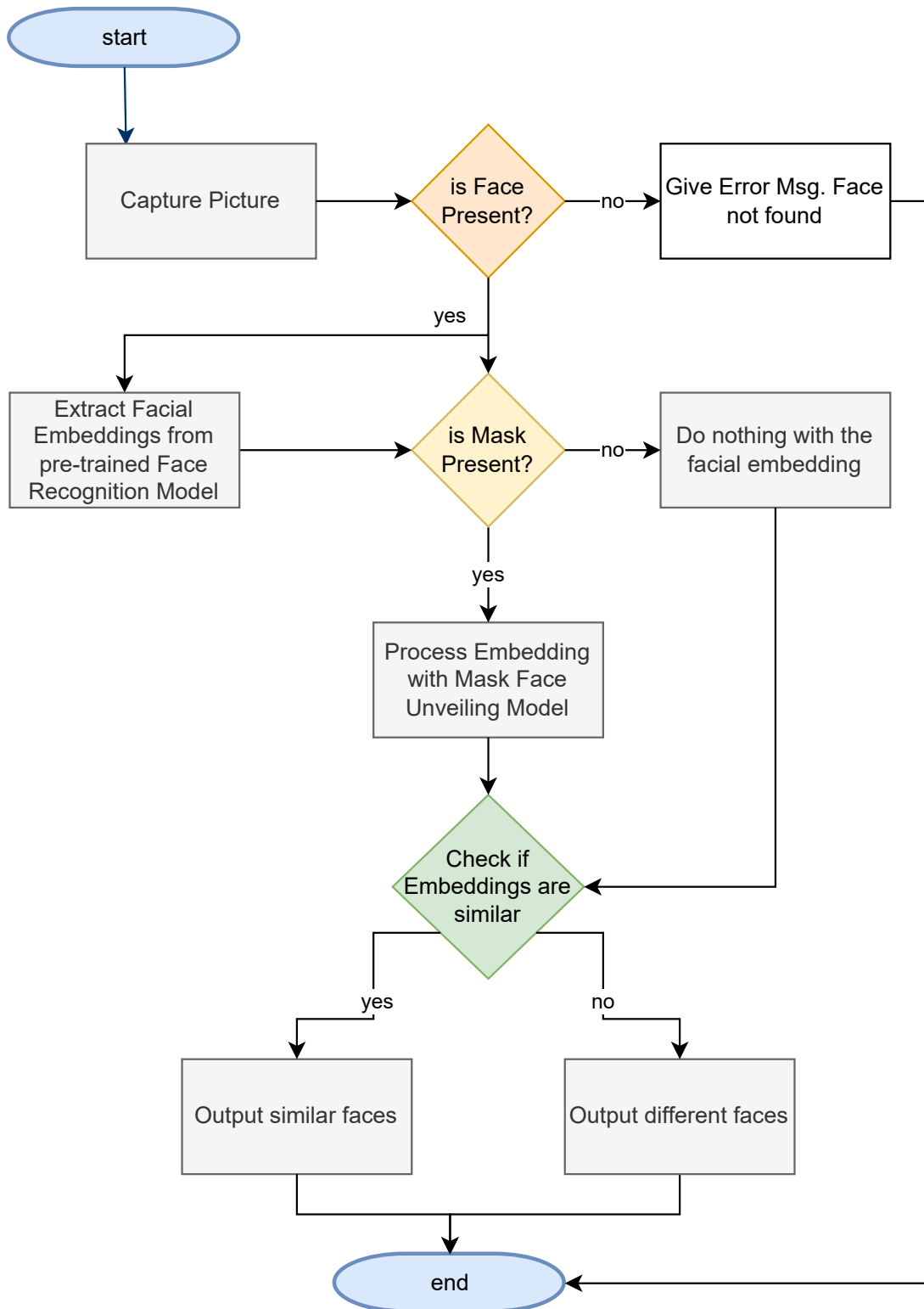


Figure 3.2: Workflow of the complete proposed system.

model. Now, we shall look into the architecture and workings of the MTCNN.

3.3.1 Architecture

MTCNN has three-stage architecture. All three stages are in series. The first stage is Proposal Network (P-Net), followed by Refine Network (R-Net) and the last stage is Output Network (O-Net). Now we shall look into the details of these networks.

1. **First Stage (P-Net)** The first stage is Proposal Network. It is a Fully Convolutional Network (FCN) which is different from CNN due to the lack of dense layers in the architecture. It is fed with an image pyramid having the same image of different scales. The output of this network is region proposals. We then apply bounding box regression to these region proposals, followed by Non-Maxima Suppression (NMS) to discard overlapping region proposals.

Now let's look into the architectural details of P-Net. It has a series of 3 convolution layers. Each layer uses a filter size of 3, but the number of filters varies in each layer, i.e. 10,16,32 for the first, second and third layers, respectively. The feature map is down-sampled using a factor of 2 after the first convolution layer using the max pool layer. Each convolution layer also has a Parametric ReLU activation function which helps in dying ReLU problem and speeds up the training. A kernel size of 1, along with the Softmax activation function, is used for probability calculation, and a linear activation function is used for bounding box regression. The architecture of P-Net is shown in Figure 3.3.

2. **Second Stage (R-Net)** The second stage is Refine Network. It is a CNN network, unlike P-Net, which was FCN. The outputs of this network are face classification which tells whether the input is a face or not. Bounding box regression which tells bounding box coordinates with vector size of 4 and facial landmarks localization using vector size of 10.

The R-Net has a series of 3 convolution layers followed by three dense layers. The first two layers of convolution layers have a kernel size of 3 and a max pooling layer after them with a downsampling factor of 2. The third layer has a filter size of 2. The number of filters for the first, second and third layers are 28, 48, and 64, respectively. In the end, a dense layer is present. The first dense layer

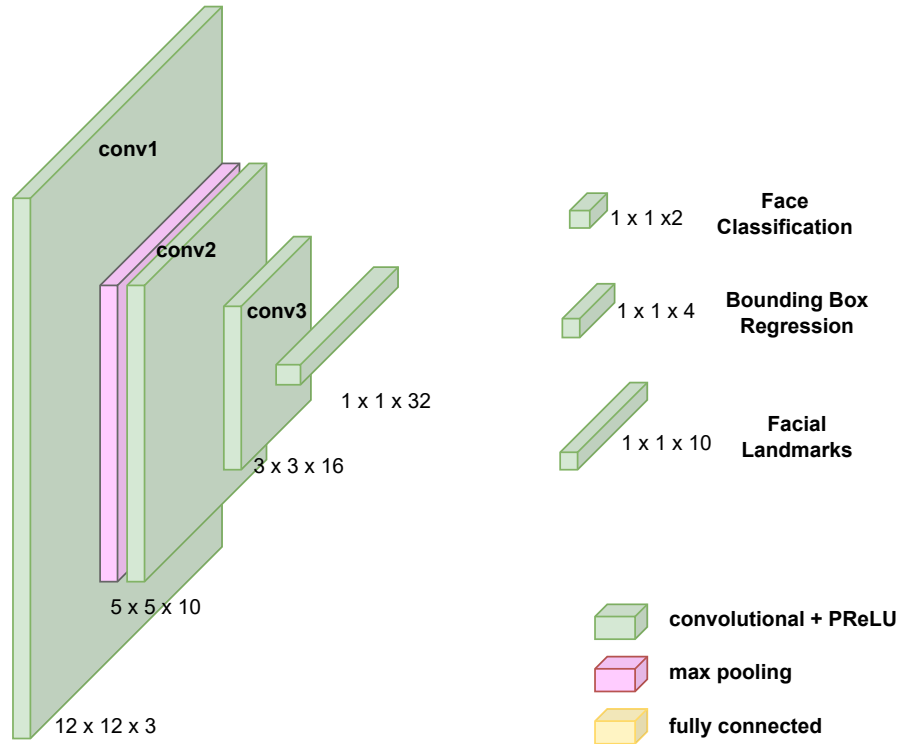


Figure 3.3: P-Net architecture diagram.

has 128 neurons, and the last three dense layers (present in parallel to each other) have 2, 4, and 10 neurons for classification, bounding box regression and landmark localization, respectively. The architecture of R-Net is shown in Figure 3.4.

- Third Stage (O-Net)** This is the final stage, and is similar to the R-Net stage. The main aim of this stage is to describe the face in more detail, and it outputs five facial landmark positions for the eyes, nose and mouth.

The O-Net has similar architecture to R-Net apart from a few differences. We shall only discuss the differences here. It has four convolution layers instead of three in case of R-Net. Every layer has filter size of 3. The number of filters are 32, 64, 64, 128 respectively for each layer. The first dense layer has 256 neurons instead of 128 neurons. The architecture of O-Net is shown in Figure 3.5.

3.4 Mask Detection

When an image is given to the system at inference time, we need to find out whether the face wears a mask in the image. Since in our proposed architecture, as shown in

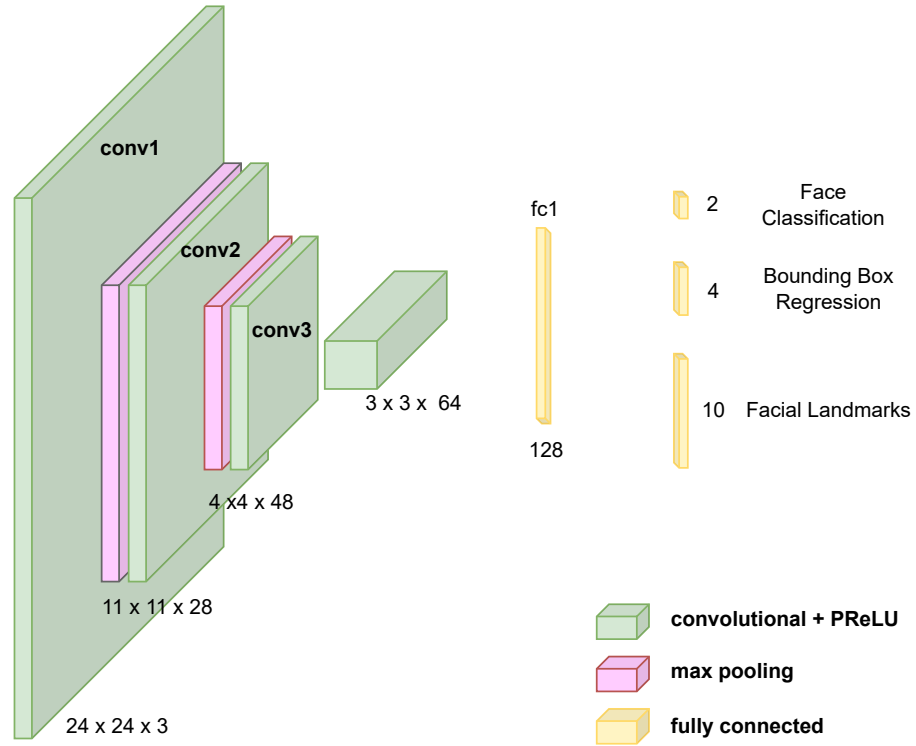


Figure 3.4: R-Net architecture diagram.

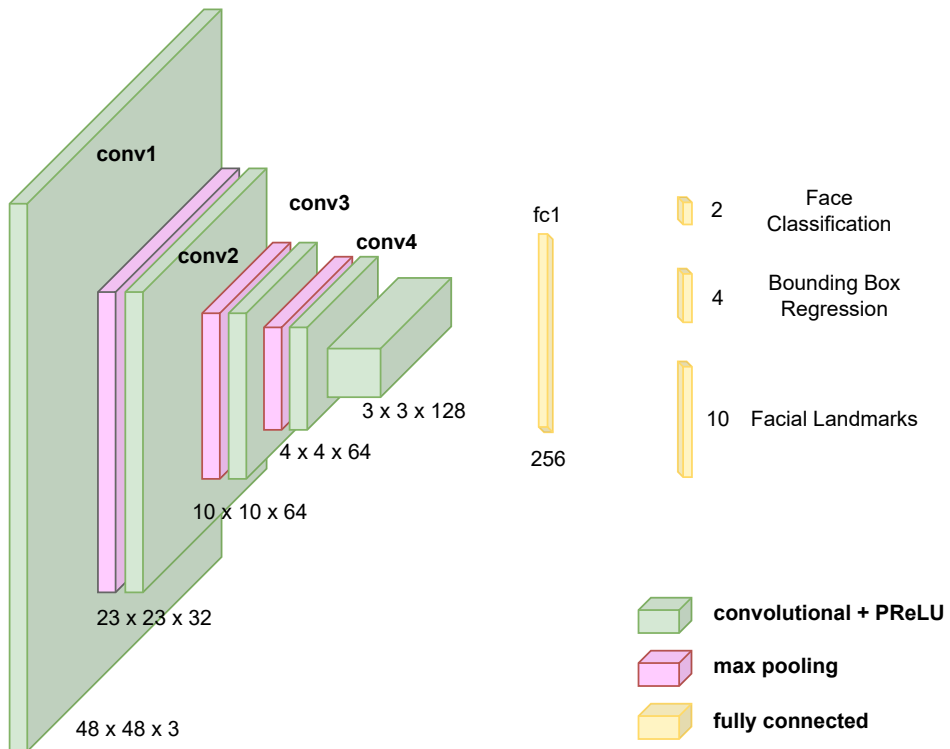


Figure 3.5: O-Net architecture diagram.

Figure 3.2, we post-process the face embeddings of only masked face images and leave the unmasked face embeddings. Therefore we need to know whether the mask is worn by the face in the input images in order to be processed by the Masked Face Unveiling Model (MFUM), which is discussed in section 3.6. This task is done by a mask detection module which outputs the presence of a mask in a facial image.

3.4.1 Dataset

Due to the absence of a large masked face dataset, we have used the combination of MFR2 [34], RWOC [39], and RMFRD [37] datasets. We have split the dataset into 80% training, 10% validation and 10% test images.

3.4.2 Architecture

Since we get cropped facial image as input, therefore we just have to check whether mask is present in face image or not. We treat this problem as a two-class classification problem. For architecture development, we have already used existing CNNs architecture like Resnet-50 due to good performance in classification scenarios. For the first experiment, we trained the model without pretrained weights. In contrast, for the second experiment, we have used a transfer learning based approach using pretrained Resnet-50 weights [20] of ImageNet dataset [12]. We have modified the last fully connected layer of the Resnet-50 model to output two classes instead of the 1000 classes present in the ImageNet dataset. These two classes tell whether a mask is worn in the facial image or not.

3.5 Feature extraction using pretrained face recognition backbone

We have used existing pretrained face recognition model backbones to extract the face embedding. For the ablation study, we have taken embeddings from two different stages of the backbone. In the first case, we take out embedding after the Global Average Pooling (GAP) or Global Depth wise Conv (GDConv) layer. This results in flat embeddings. Whereas in the second case, we take out embedding before the GAP layer, giving us a 3D feature map instead of flat embeddings. The details of where and how

these embeddings will be utilized are given in the MFUM architecture section.

3.5.1 Resnet-101

Resnet [20] is well-known and the most popular CNN architecture. We have used Resnet-101, which is trained on MS1MV2 [49] dataset. It generates 512-dimensional embeddings after the GAP layer, and before the GAP layer, the feature size is $7 \times 7 \times 512$. We save both embeddings to be processed later by MFUM architecture. The complete architecture for Resnet-101 is given in Table 3.1.

Table 3.1: Architecture of Resnet101.

layer	input size	output size	filters	repetitions	
conv1_x	112x112x3	112x112x64	7x7x64, stride 1	0	
conv2_x	112x112x64	56x56x256	3x3 max pool, stride 2		0
			$1 \times 1, 64$	3	
			$3 \times 3, 64$		
$1 \times 1, 256$					
conv3_x	56x56x256	28x28x512	$1 \times 1, 128$	4	
			$3 \times 3, 128$		
			$1 \times 1, 512$		
conv4_x	28x28x512	14x14x1024	$1 \times 1, 256$	23	
			$3 \times 3, 256$		
			$1 \times 1, 1024$		
conv5_x	14x14x1024	7x7x512	$1 \times 1, 512$	3	
			$3 \times 3, 512$		
GAP	7x7x512	1x1x512	-	0	

3.5.2 MobileFaceNet

MobileFaceNet [29] is a lightweight backbone and runs efficiently on mobile devices without drastically affecting performance. MobileNetV2 greatly inspires the architecture

of MobileFaceNet as residual bottleneck makes up most of the network. The architecture of the bottleneck layer is shown in Figure. 3.6.

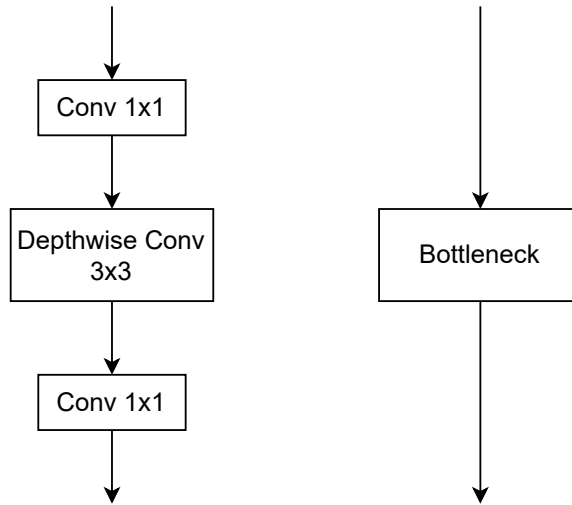


Figure 3.6: Bottleneck layer architecture in MobileFaceNet.

The major difference of this architecture from MobileNetV2 is that it has no Global Average Pooling layer, and it uses PRelu non-linearity instead of Relu. The network also uses a fast down-sampling strategy at the beginning and a linear 1x1 layer after the Global Depth wise Conv (GDConv) layer as output. The MobileFaceNet architecture is shown in Table 3.2. For flattened embedding of 1 dimension, we use the feature map after the linear 1x1 layer, whereas for 7x7x128 embedding, we use the feature map after the last bottleneck layer.

3.6 Masked Face Unveiling Model (MFUM)

The masked face embeddings generated from the pretrained face recognition backbone are processed by Masked Face Unveiling Model (MFUM) to make them similar to unmasked face embeddings of the same person and different from unmasked face embeddings of a different person. The overall context in which MFUM is trained and works are shown in Figure 3.8. We have developed seven MFUM architectures for the ablation study. These architectures are developed keeping in mind the dimension of the embedding from different backbones and attention mechanisms. These are discussed in detail below.

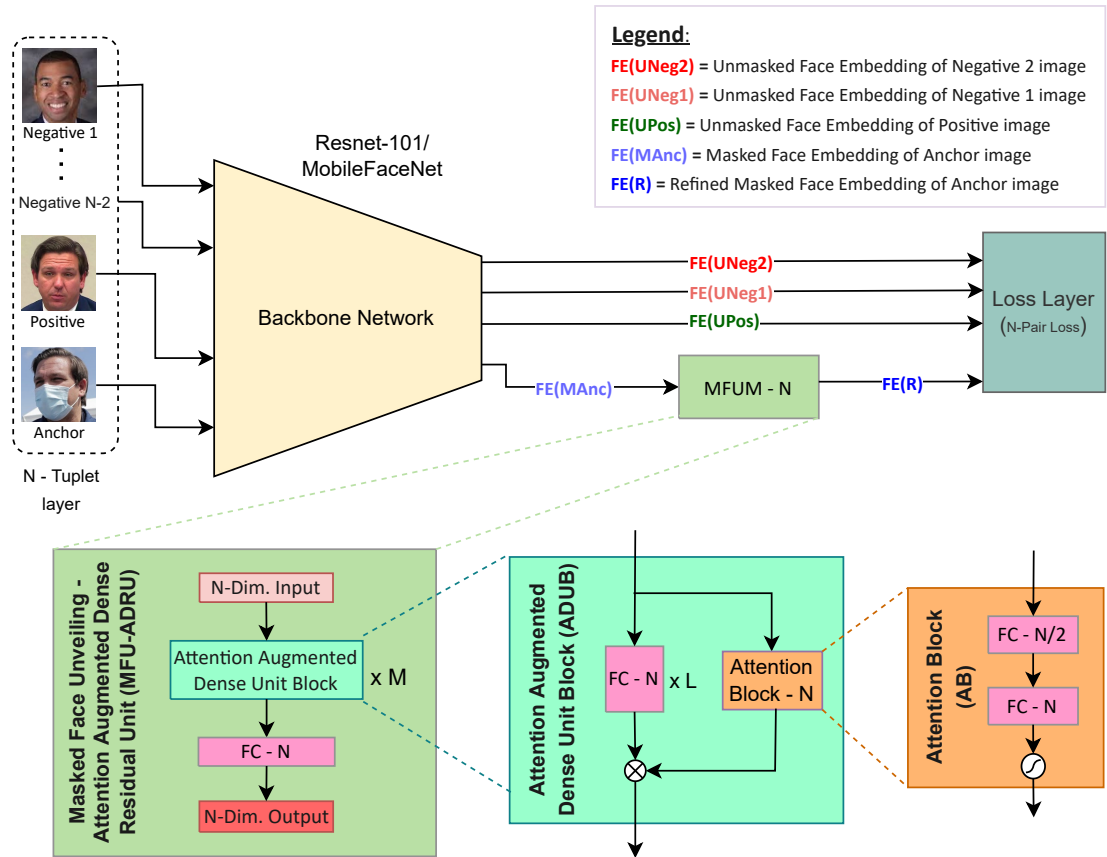


Figure 3.7: Overall proposed MFUM architecture diagram workflow along with MFU-ADRU architecture.

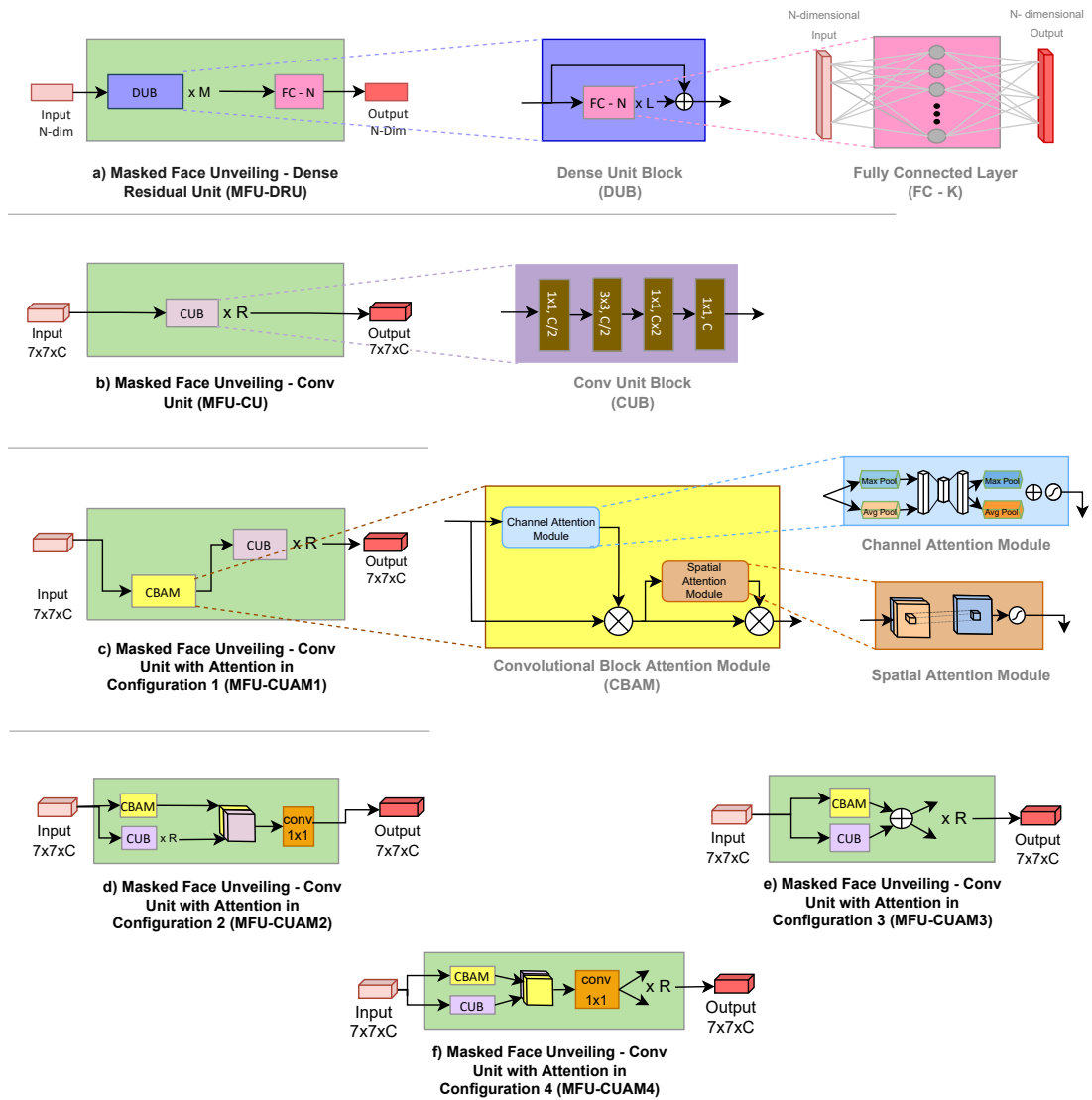


Figure 3.8: Alternate MFUM architectures.

Table 3.2: Architecture of MobileFaceNet.

Input	Operator	expansion factor	output channels	repetitions	stride
112x112x3	conv3x3	-	64	1	2
56x56x64	depthwise conv3x3	-	64	1	1
56x56x64	bottleneck	2	64	5	2
28x28x64	bottleneck	4	128	1	2
14x14x128	bottleneck	2	128	6	1
14x14x128	bottleneck	4	128	1	2
7x7x128	bottleneck	2	128	2	1
7x7x128	conv1x1	-	512	1	1
7x7x512	linear GDCConv7x7	-	512	1	1
1x1x512	linear conv1x1	-	128	1	1

3.6.1 Masked Face Unveiling Dense Residual Unit (MFU-DRU)

Masked Face Unveiling Dense Residual Unit (MFU-DRU) takes flatten embeddings from the backbone as input and does some processing on it and then outputs flatten embeddings of the same dimension as input. Its architecture is included in figure 3.8.

Architecture

MFU-DRU architecture has M Dense Unit Blocks (DUB) connected in series which are then followed by a last fully connected layer. Each DUB has L fully connected layers consisting of N number of neurons each. The values of M is 3 and L is 2 and are found after extensive experimentation. Whereas, the value of N depends upon the backbone face recognition model used and it is the size of embeddings that the backbone produces. A skip connection is introduced to the output of each DUB from the input of that DUB block. The input is passed through 3 DUB in series and in the end passed through a fully connected layer. The output is N-Dimensional embedding for the original masked face embedding which is similar to the unmasked embedding of the same person and different from the facial embedding of a different person.

Number of trainable parameters

The number of trainable parameters for MFU-DRU with Resnet-101 backbone is approximately 1.83 million, whereas approximately 0.114 million for MobileFacenet backbone.

3.6.2 Masked Face Unveiling Attention Augmented Dense Residual Unit (MFU-ADRU)

Masked Face Unveiling Attention Augmented Dense Residual Unit (MFU-ADRU) is similar to the MFU-DRU, but with an additional attention mechanism. The input of MFU-ADRU is flattened embeddings from the existing face recognition model backbone. Some processing is done on the input which results in refined embedding of the same dimension as the input. Its architecture is included in Figure 3.8.

Architecture

There is a stack of M Augmented Dense Unit Block (ADUB) in the FUADRU followed by a FC layer. Furthermore, there is a stack of L FC layers in each ADUB. The value of M and L are kept equal to 3 and 2, respectively. These values are optimal and are found after several experiments.

There is also an additional block in parallel to stack of FC layers, called the Attention Block (AB), which uses the same input as the first FC layer of the ADUB and aims to calculate attention by giving weights to skip connection from the input. The AB generates output with values between 0 to 1, i.e. mask output. The result of the stack of FC layers is combined with the output of AB by taking the Hadamard product, which generates the outcome of ADUB. There are two FC layers in AB. There is half the number of neurons as input dimension in the first layer and neurons equal to input dimension in the second layer. In the end, there is sigmoid activation which generates a mask. And this mask is applied to the output of the stack of FC layers in ADUB.

Number of trainable parameters

The total number of trainable parameters for MFU-ADRU with Resnet-101 backbone is approximately 2.2 million, whereas approximately 0.139 million for MobileFacenet backbone.

3.6.3 Face Unveiling Conv Unit (MFU-CU)

Masked Face Unveiling Conv Unit (MFU-CU) takes intermediate feature map embeddings from the backbone as input and pass it through stack of Conv Unit Blocks (CUB) and then outputs feature map embeddings of same dimension as input. Its architecture is included in Figure 3.8. Each CUB consist of stack of convolution layers. Its architecture is also included in Figure 3.8.

Number of trainable parameters

The total number of trainable parameters for MFU-CU with Resnet-101 backbone is approximately 4.5 million, whereas approximately 0.282 million for MobileFacenet backbone.

3.6.4 Masked Face Unveiling Conv Unit with Attention Mechanism (MFU-CUAM1)

Masked Face Unveiling Conv Unit with Attention Mechanism (MFU-CUAM) is like MFU-CU with additional attention mechanism. It takes intermediate feature map embeddings from the backbone as input and pass it through stack of Conv Unit Blocks and Convolutional Block Attention Module (CBAM) [32]. We have tried different architectural structures for Conv Unit with CBAM. They all are included in Figure 3.8 along with architecture of the CBAM.

Number of trainable parameters

The total number of trainable parameters for MFU-CUAM1 with Resnet-101 backbone is approximately 5.04 million, whereas approximately 0.31 million for MobileFacenet backbone.

The total number of trainable parameters for MFU-CUAM2 with Resnet-101 backbone is approximately 5.04 million, whereas approximately 0.31 million for MobileFacenet backbone.

The total number of trainable parameters for MFU-CUAM3 with Resnet-101 backbone is approximately 6.09 million, whereas approximately 0.38 million for MobileFacenet

backbone.

The total number of trainable parameters for MFU-CUAM4 with Resnet-101 backbone is approximately 6.09 million, whereas approximately 0.38 million for MobileFacenet backbone.

3.7 Training

In this section, we discuss the training details of MFUM. We look into the loss functions and model training settings in the following subsections.

3.7.1 Loss Functions

Loss function is used for training the MFUM. It helps MFUM to learn discriminative facial embeddings. We have used three loss functions i.e. triplet loss, self-restrained triplet and quadruplet loss for the ablation studies. Now, we shall discuss these losses in detail.

Triplet Loss

Let $i \in I$ representing batch of training images. $f(i)$ is the embedding generated from pretrained backbone. Triplet loss requires a triplet of sample in the form of $\{i_j^a, i_j^p, i_j^n\} \in I$, where i_j^a is anchor, i_j^p is positive example of same identity as anchor, i_j^n is negative example. For mini batch of size N , mathematically Triplet loss is defined as:

$$L_t = \frac{1}{N} \sum_j \max([d(f(i_j^a), f(i_j^p)) - d(f(i_j^a), f(i_j^n)) + m], 0) \quad (3.7.1)$$

where m is the margin which introduces separability between imposter and genuine pairs. d is the euclidean distance and is mathematically defined as,

$$d(a_j, b_j) = \|a_j - b_j\|^2 \quad (3.7.2)$$

Self Restrained Triplet Loss

SRT loss also requires a triplet to be defined i.e. $\{i_j^a, i_j^p, i_j^n\} \in I$ where i_j^a is anchor, i_j^p is positive example of same identity as anchor, i_j^n is negative example. For a mini batch of N size, mathematically SRT loss is defined as:

$$L_{SRT} = \begin{cases} (\frac{1}{N} \sum_j^N \max([d(f(i_j^a), f(i_j^p)) - d(f(i_j^a), f(i_j^n)) + m], 0) & \text{if } \mu(d2) < \mu(d3)) \\ (\frac{1}{N} \sum_j^N \max([d(f(i_j^a), f(i_j^p)) - \mu(d3) + m, 0) & \text{otherwise}) \end{cases} \quad (3.7.3)$$

where d is euclidean distance as described in Equation. 3.7.2, m is margin which impose separability between imposter and genuine pairs. $d1$ is anchor positive distance, $d2$ is anchor negative distance and $d3$ is positive negative distance. $\mu(d2), \mu(d3)$ are mean of anchor-negative distance and mean of positive - distance across the batch respectively. SRTL behaves as triplet loss when $\mu(d2) < \mu(d3)$ i.e. mean of anchor-negative distance is smaller than mean of positive-negative distance across the batch. In this scenario, the loss focuses on minimizing the anchor positive distance and maximizing the anchor negative distance. But when $\mu(d2) \geq \mu(d3)$, for calculating the loss, $d2$ is replaced with $d3$. By doing this, only $d1$ distance is minimized and anchor-positive examples are brought closer to each other.

Quadruplet Loss

Let $i \in I$ representing batch of training images. $f(i)$ is the embedding from pretrained face recognition model. Quadruplet loss requires a quadruplet of sample in the form of $\{i_j^a, i_j^p, i_j^{n1}, i_j^{n2}\} \in I$, where i_j^a is anchor, i_j^p is positive example of same identity as anchor, i_j^{n1} is first negative example and i_j^{n2} of is second negative example of different identity. The learning objective of quadruplet loss function is to reduce the distance of anchor $f(i_j^a)$ and positive $f(i_j^p)$ (genuine pair) and maximize the distance of anchor $f(i_j^a)$ and first negative example $f(i_j^{n1})$ i.e. first imposter pair as well as anchor $f(i_j^a)$ and second negative example $f(i_j^{n2})$ i.e. second imposter pair. Mathematically, Quadruplet loss L_q for a mini-batch of size N is defined as :

$$L_q = \frac{1}{N} \left(\sum_j^N [d(f(i_j^a), f(i_j^p)) - d(f(i_j^a), f(i_j^{n1})) + m1] + \sum_j^N [d(f(i_j^a), f(i_j^p)) - d(f(i_j^a), f(i_j^{n2})) + m2] \right) \quad (3.7.4)$$

where $m1$ is margin1 and $m2$ is margin2. These margins are applied to increase the separability between imposter and genuine pairs. d is euclidean distance which is applied

on normalized facial feature. d mathematical representation is given in 3.7.2.

3.7.2 Model Training settings

We have used Python Pytorch library for the implementation of MFUM. The MFUM is trained using already extracted feature embedding of the MS1MV2 dataset. For the purpose of comparison, MFUM is trained using different architecture and loss functions. For training MFUM with each architecture and corresponding loss, training settings are given Table 3.3.

Table 3.3: Experimental Settings.

Experimental Parameters	Settings
Epochs	20
Batch Size	512
Initial Learning rate	0.1
Optimizer	SGD
Learning Rate Decay step size	9
Learning Rate Decay factor	0.1

3.8 Evaluation

3.8.1 Experimental Settings

For each evaluation dataset, we have used nine different experimental setups. These are discussed in detail below.

- 1. Unmasked Reference Unmasked Probe (UMR-UMP):**

In this experimental setting, the unmasked reference is compared with unmasked probe. It gives us the baseline performance of pretrained face recognition model on unmasked reference and unmasked probe image.

- 2. Unmasked Reference Masked Probe (UMR-MP):**

In this experimental setting, the unmasked reference is compared with masked probe without applying any additional processing on masked probe with MFUM.

It gives us the baseline performance of pretrained face recognition model on unmasked reference but masked probe scenario.

3. Unmasked Reference and Masked Probe processed with MFUM trained with Triplet Loss (UMR-MP(T)):

In this experimental setting, the unmasked reference is compared with masked probe. The masked probe is additionally processed by MFUM trained with Triplet loss. It gives us the performance of pretrained face recognition model aided with MFUM trained with Triplet loss in unmasked reference and masked probe scenario.

4. Unmasked Reference and Masked Probe processed with MFUM trained with Self Restrained Triplet Loss (UMR-MP(SRT)):

In this experimental setting, the unmasked reference is compared with masked probe. The masked probe is additionally processed by MFUM trained with Self Restrained Triplet loss. It gives us the performance of pretrained face recognition model aided with MFUM trained with SRT loss in unmasked reference and masked probe scenario.

5. Unmasked Reference and Masked Probe processed with MFUM trained with Quadruplet Loss (UMR-MP(Q)):

In this experimental setting, the unmasked reference is compared with masked probe. The masked probe is additionally processed by MFUM trained with Quadruplet loss. It gives us the performance of pretrained face recognition model aided with MFUM trained with Quadruplet loss in unmasked reference and masked probe scenario.

6. Masked Reference Masked Probe (MR-MP):

In this experimental setting, the masked reference is compared with masked probe without applying any additional processing on masked reference and masked probe with MFUM. It gives us the baseline performance of pretrained face recognition model on masked reference and masked probe scenario.

7. Masked Reference and Masked Probe processed with MFUM trained with Triplet Loss (MR-MP(T)):

In this experimental setting, the masked reference is compared with masked probe. The masked reference and masked probe are additionally processed by MFUM

trained with Triplet loss. It gives us the performance of pretrained face recognition model aided with MFUM trained with Triplet loss in masked reference and masked probe scenario.

8. Masked Reference and Masked Probe processed with MFUM trained with Self Restrained Triplet Loss (MR-MP(SRT)):

In this experimental setting, the masked reference is compared with masked probe. The masked reference and masked probe are additionally processed by MFUM trained with SRT loss. It gives us the performance of pretrained face recognition model aided with MFUM trained with SRT loss in masked reference and masked probe scenario.

9. Masked Reference and Masked Probe processed with MFUM trained with Quadruplet Loss (MR-MP(Q)):

In this experimental setting, the masked reference is compared with masked probe. The masked reference and masked probe are additionally processed by MFUM trained with Quadruplet loss. It gives us the performance of pretrained face recognition model aided with MFUM trained with Quadruplet loss in masked reference and masked probe scenario.

The verification performance of our approaches is evaluated using the metrics which are adopted globally for the evaluation of biometric systems performance. These evaluation metrics include False Match Rate (FMR100 and FMR1000), False Non-Match Rate (FNMR) and Equal Error Rate (EER), Mean of genuine scores, Mean of imposter scores and Fischer Discriminant Ratio. Now we shall look into the details of these evaluation metrics.

3.8.2 Evaluation Metric

1. **Equal Error Rate (EER):** This metrics reports the performance of the biometric system in the verification scenario. It is the value or threshold at which the False Acceptance Rate (FAR) equals the False Rejection Rate (FRR). Generally speaking, if EER is less, the accuracy will be more.

2. **False Match Rate (FMR):**

It is the probability that two biometric signals (in our case, face images) coming

Table 3.4: Experimental Settings Abbreviation and their elaboration.

Abbreviation	Elaboration
UMR-UMP	Unmasked Reference and Unmasked Probe, both not processed additionally with MFUM
UMR - MP	Unmasked Reference and Masked Probe, with Masked Probe not processed additionally with MFUM
UMR - MP (T)	Unmasked Reference and Masked Probe, with Masked Probe processed additionally with MFUM trained with Triplet loss
UMR - MP (SRT)	Unmasked Reference and Masked Probe, with Masked Probe processed additionally with MFUM trained with Self Restrained Triplet loss
UMR - MP (Q)	Unmasked Reference and Masked Probe, with Masked Probe processed additionally with MFUM trained with Quadruplet loss
MR - MP	Masked Reference and Masked Probe, both not processed additionally with MFUM
MR - MP (T)	Masked Reference and Masked Probe, both processed additionally with MFUM trained with Triplet loss
MR - MP (SRT)	Masked Reference and Masked Probe, both processed additionally with MFUM trained with Self Restrained Triplet loss
MR - MP (Q)	Masked Reference and Masked Probe, both processed additionally with MFUM trained with Quadruplet loss

from different identities are misclassified as coming from the same identity.

3. False Non Match Rate (FNMR):

It is the probability that two biometric signals (in our case, face images) coming from the same identities are misclassified as coming from different identities.

4. FMR100:

It is the lowest FNMR for which $FMR \leq 1.0\%$.

5. FMR1000:

It is the lowest FNMR for which $FMR \leq 0.1\%$.

6. FMR100_Th and FMR1000_Th:

To estimate the real-life scenario in which operational thresholds are decided based on UMR-UMP performance, we have reported FMR100_Th and FMR1000_Th for every experimental setting, e.g. UMP-MP(Q), MR-MP(Q) etc.

7. Mean of Genuine Score:

It is the average of all genuine pair similarity scores.

8. Mean of Imposter Score:

It is the average of all imposter pair similarity scores.

9. Fisher Discriminant Ratio (FDR):

It is the ratio of the difference between the mean of the genuine score and the mean of the imposter score to the sum of squares of the standard deviation of genuine and imposter scores. Mathematically, FDR is given as :

$$FDR = \frac{\mu_G - \mu_I}{(\sigma_G)^2 + (\sigma_I)^2}$$

where μ_G is mean of genuine score, μ_I is mean of imposter score, σ_G is standard deviation of genuine score values and σ_I is standard deviation of imposter score values. It helps in studying the separability of imposter and genuine scores. A greater value of FDR means better separation between genuine and imposter scores.

3.9 Working Demo Application

For the proof of concept, we have developed two working demo applications for verification as well as recognition scenario.

3.9.1 Verification Portal

Face verification is the task of comparing a face image to another face image and checking if both images are of the same person or not. This task is a one-to-one mapping task. For the purpose of the verification scenario demo, we have developed a web application using Python Flask API. The intuition behind developing this web application was to make the demo run on devices of different platforms by just using a web link. Now, let's look into the working of this application.

Input

This application takes two images as input. The input can be provided using an existing file stored on the device or webcam. For the demo purpose, we have given the user option to check various models which we have developed, i.e. MFU-DRU, MFU-ADRU, MFU-CU and MFU-CUAM etc., as discussed in section. 3.6. Additionally, these models can be evaluated using different loss functions, i.e. Triplet, SRT, and Quadruplet loss, as discussed in section. 3.7.1. These options can be selected using the dropdown menu.

Processing

We first detect whether the face is present in the images or not using the face detector discussed in section. 3.3. If a face is not found, we exit the pipeline after giving an error message to the user. Once a face is detected, we take out face embedding using backbones discussed in the section. 3.5. In addition to this, after face detection, we detect whether the person in the image is wearing a mask or not. If the person wears a mask in the image, then we process the embedding using MFUM architecture as discussed in section. 3.6. If a mask is not worn, then we do not process the embedding using MFUM. Once we have embeddings, we compute the cosine similarity of these embeddings. If the similarity is above the threshold, then we predict both images to be of the same person, otherwise different.

Output

This application outputs whether the image is of the same person or not. Additionally, for each image, we also predict whether the person in the image is wearing a mask or not. Figure 3.9 shows the verification portal flowchart.

3.9.2 Recognition Portal

Face recognition is the task of matching digital images or frames of videos against the database of faces. It is a one-to-many mapping scenario in which a single image is compared to many images. For the purpose of the demo, we have developed a python desktop application that takes a webcam video stream as input and performs recognition on video frames. Now, we shall look into the details of this application.

Input

The application takes video stream as input from a webcam or pre-recorded video in mp4 format.

Processing

The first step in this application is generating the face embedding database for the faces to be recognized. We take out the embeddings of face images. Then, we process the video stream frame-wise. Once we have the frame, we first detect the presence of a face in the frame. If no frame is present, then we take the next frame for processing. If a face is found, then we take out embeddings using a pretrained backbone as discussed in section 3.5. In addition to this, for the detected face, we detect whether the person in the image is wearing a mask or not. If the person wears a mask in the image, then we process the embedding using MFUM architecture as discussed in section 3.6. If a mask is not worn, then we do not process the embedding using MFUM. Once we have embeddings, we compute the cosine similarity of the embedding against the face embedding database. We assign that label to the bounding box whose cosine similarity with the processed face is maximum.

Output

The application outputs a real-time video stream with a named label bounding box. Additionally, we also output mask is worn or not status along with the name. Figure 3.11 shows the recognition portal.

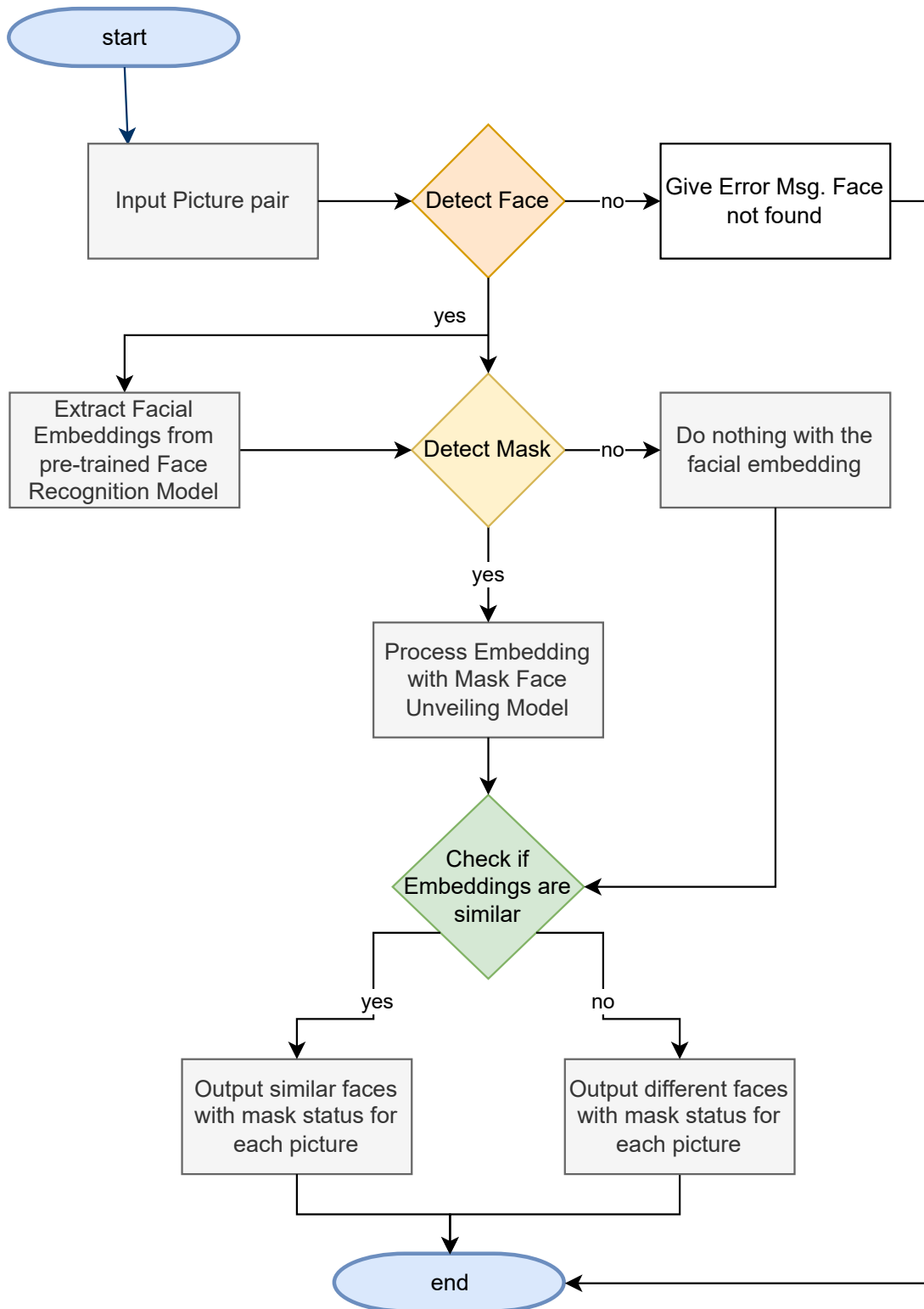


Figure 3.9: Verification Portal Flow Chart.

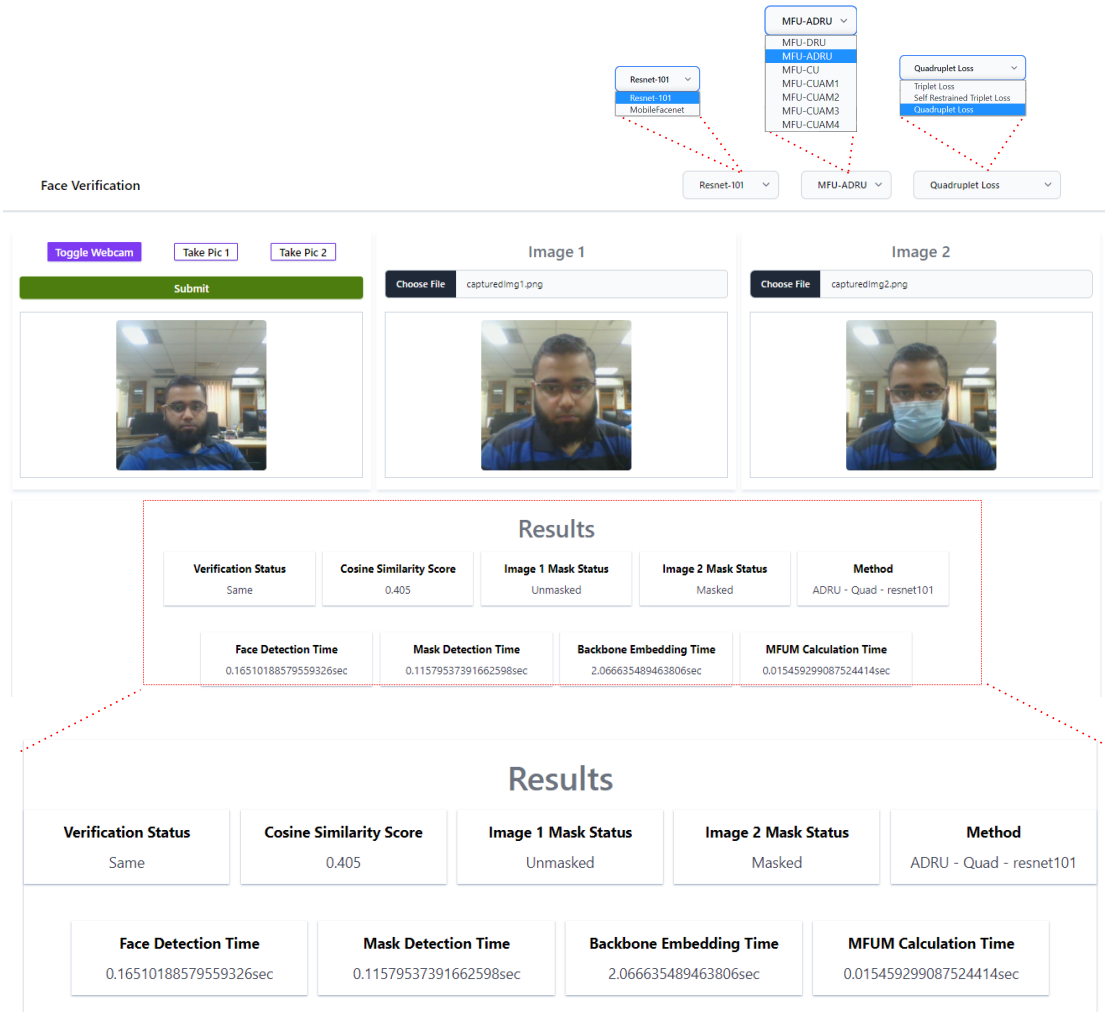


Figure 3.10: Face verification portal shows whether two given images are of the same entity along with mask status, cosine similarity score, and time taken by face detection, mask detection, and backbone embedding generation.

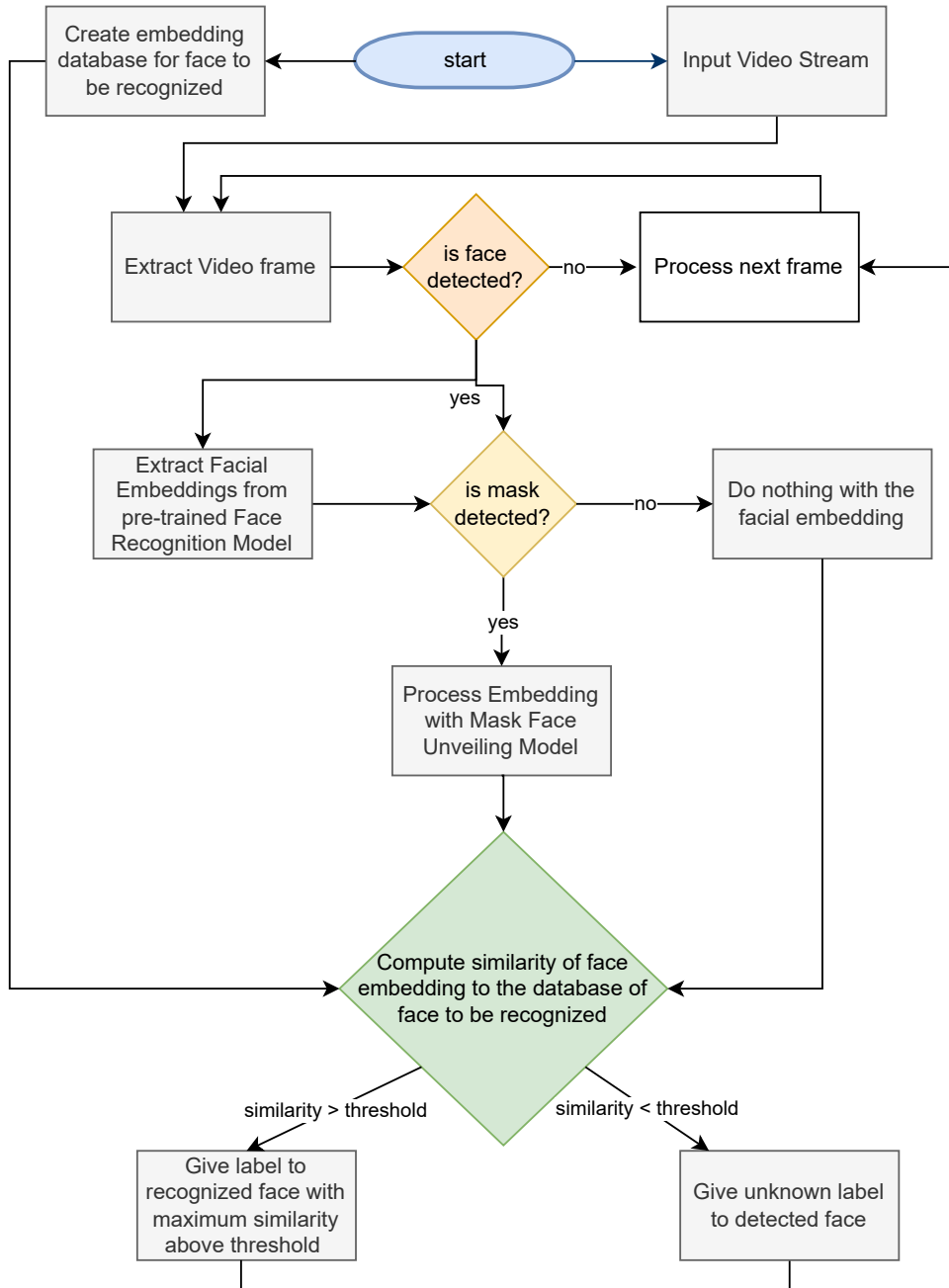


Figure 3.11: Recognition Portal Flow Chart.

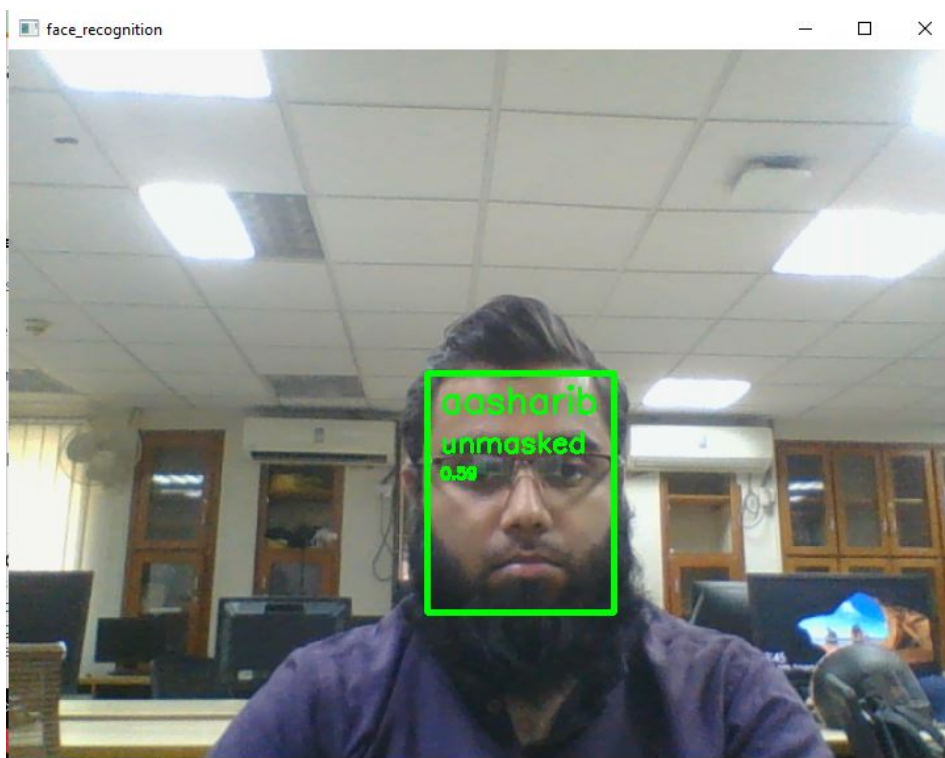


Figure 3.12: Face Recognition portal showing the name of the recognized entity along with mask status and cosine similarity score.



Figure 3.13: Face Recognition portal showing the name of the recognized entity along with mask status and cosine similarity score.

Results

In this section, we present the qualitative as well as quantitative results for our proposed methodology. For the quantitative results, we have defined evaluation metrics in section. 3.8.2 to evaluate the nine experimental settings discussed in section. 3.8.1. For qualitative results, we have created two demo applications as discussed in section. 3.9. First of all, we present the impact on face recognition performance due to wearing of face mask. Then, we present the MFUM’s impact on the performance of masked face verification and the separability of genuine and imposter scores. Then, we present the ablation study on different MFUM architectures as well as loss functions by comparing their performances. These results are shown in Table 4.1-4.2 and 4.7-4.18. Then, we compare our results with state-of-the-art approaches. The comparison with the state-of-the-art is shown in Table 4.3-4.6. We have also conducted another ablation study in which we trained the MFUM architectures using the MFR2 dataset which is a small real-world masked face dataset and evaluated on LFW and MFR2 datasets to check the performance of MFUM. These results are shown in Table 4.19-4.32. In the end, we present the scenarios in which our approach suffers degradation in performance or needs improvements.

4.1 Impact of face mask on face recognition performance

4.1.1 UMR-MP scenario with flattened embeddings

Table 4.1-4.2 and Table 4.7-4.8 show the comparison of baseline experimental setting, i.e. unmasked reference unmasked probe (UMR-UMP) and unmasked reference masked

probe (UMR-MP) in which probe is wearing a mask. We can observe clearly that the UMR-UMP case is achieving better verification performance than the UMR-MP case for Resnet101 and MobileFaceNet backbone. For the LFW dataset, the UMR-UMP case has achieved 0.3%, 0.7% EER% for Resnet-101 and MobileFaceNet backbones, respectively, compared to 0.7%, 3.8% EER% for Resnet-101 and MobileFaceNet backbones respectively in UMR-MP scenario (Table 4.1, 4.7). Similarly, for the MFR2 dataset, the UMR-UMP case has achieved 0.0%, 0.0% EER% for Resnet-101 and MobileFaceNet backbones, respectively, compared to 9.2224%, 9.5927% EER% for Resnet-101 and MobileFaceNet backbones respectively in UMR-MP scenario (Table 4.2, 4.8). These results clearly show the degradation in masked face recognition performance due to the presence of a mask in the probe image.

4.1.2 MR-MP scenario with flattened embeddings

Table 4.1-4.2 and Table 4.7-4.8 show the comparison of baseline experimental setting i.e. unmasked reference unmasked probe (UMR-UMP) and masked reference masked probe (MR-MP) in which both reference and probe are wearing masks. We can observe clearly that the UMR-UMP case is achieving better verification performance than the MR-MP case for Resnet101 and MobileFaceNet backbone. For the LFW dataset, the UMR-UMP case has achieved 0.3%, 0.7% EER% for Resnet-101 and MobileFaceNet backbones as compared to 0.7%, 3.6% EER% for Resnet-101 and MobileFaceNet backbones in MR-MP scenario (Table 4.1, 4.7). Similarly for MFR2 dataset, the UMR-UMP case has achieved 0.0%, 0.0% EER% for Resnet-101 and MobileFaceNet backbones as compared to 4.6434%, 7.730% EER% for Resnet-101 and MobileFaceNet backbones in UMR-MP scenario (Table 4.2, 4.8). These results clearly show the degradation in masked face recognition performance due to the presence of masks in reference and probe images.

4.1.3 UMR-MP scenario with intermediate feature-map embeddings

Table 4.9-4.18 show the comparison of baseline experimental setting i.e. unmasked reference unmasked probe (UMR-UMP) and unmasked reference masked probe (UMR-MP) in which probe is wearing a mask. We can observe clearly that the UMR-UMP case is achieving better verification performance than the UMR-MP case for Resnet101 and MobileFaceNet backbone. For the LFW dataset, the UMR-UMP case has achieved

0.3%, 6.9% EER% for Resnet-101 and MobileFaceNet backbones respectively as compared to 0.5%, 10.7% EER% for Resnet-101 and MobileFaceNet backbones respectively in UMR-MP scenario (Table 4.9, 4.11, 4.13, 4.15, 4.17). Similarly for MFR2 dataset, the UMR-UMP case has achieved 0.0%, 2.4214% EER% for Resnet-101 and MobileFaceNet backbones respectively as compared to 4.9534%, 16.0969% EER% for Resnet-101 and MobileFaceNet backbones respectively in UMR-MP scenario (Table 4.10, 4.12, 4.14, 4.16, 4.18). These results clearly show the degradation in masked face recognition performance due to the presence of a mask in the probe image.

4.1.4 MR-MP scenario with intermediate feature-map embeddings

Table 4.9-4.18 show the comparison of baseline experimental setting i.e. unmasked reference unmasked probe (UMR-UMP) and masked reference and masked probe (MR-MP) in which both reference and probe are wearing mask. We can observe clearly that the UMR-UMP case is achieving better verification performance than the MR-MP case for Resnet101 and MobileFaceNet backbone. For LFW dataset, the UMR-UMP case has achieved 0.3%, 6.9% EER% for Resnet-101 and MobileFaceNet backbones as compared to 0.8%, 11.5% EER% for Resnet-101 and MobileFaceNet backbones in MR-MP scenario (Table 4.9, 4.11, 4.13, 4.15, 4.17). Similarly for the MFR2 dataset, the UMR-UMP case has achieved 0.0%, 2.4214% EER% for Resnet-101 and MobileFaceNet backbones as compared to 8.8556%, 20.7086% EER% for Resnet-101 and MobileFaceNet backbones in MR-MP scenario (Table 4.10, 4.12, 4.14, 4.16, 4.18). These results clearly show the degradation in masked face recognition performance due to the presence of masks in reference and probe images.

4.1.5 Comparison of face mask effect on FNMR and FMR

Table 4.1-4.2 and Table 4.7-4.18 shows a greater negative impact on FNMR as compared to FMR. For example, for the LFW dataset, if we compare the baseline performance for UMR-UMP and UMR-MP cases on the Resnet-101 backbone, we find out that FNMR has increased to 0.8% from 0.3% as compared to FMR which has decreased from 1.0% to 0.5% (Table 4.7). Another example is, for MFR2 dataset, if we compare the baseline performance for UMR-UMP and UMR-MP cases on the Resnet-101 backbone, we find out that FMR has increased from 0.9983% to 1.0284%. Whereas FNMR has

increased from 0.0% to 8.6687%, which is a larger negative impact as compared to FMR (Table 4.8).

4.1.6 Comparison of UMR-MP and MR-MP scenarios

Table 4.2 and Table 4.8-4.18 shows lower verification performance achieved by MR-MP case compared to UMR-MP on Resnet-101 and MobileFacenet backbones on LFW dataset as well as MFR2 dataset. There is a slight exception in LFW dataset results (Table 4.1 and 4.7), where MR-MP verification performance is slightly better than the UMR-MP case for the Resnet-101 backbone. But overall UMR-MP performance is better than MR-MP case.

4.1.7 Comparison of backbones performance

Table 4.7-4.18 shows the performance of Resnet-101 and MobileFacenet in different experimental settings. It is quite evident that the verification performance of Resnet-101 is better than MobileFacenet in all experimental settings.

4.1.8 Summary

Summarizing the above findings, we can conclude that the performance of face recognition models deteriorates with the introduction of face masks. These results are very much in accordance with results presented in previous studies like Ngan et al. [36], Damer et al. [48], who have evaluated the negative impact of masks on the performance of face recognition models.

4.2 MFUM impact on masked face recognition performance

Table 4.1-4.2 and Table 4.7-4.18 shows the improvement in verification performance of existing face recognition performance due to the addition of MFUM trained with Quadruplet and Self Restrained Triplet loss.

4.2.1 Impact of MFUM on the separability of genuine and imposter scores

Our proposed approach utilizing MFUM has improved the separability of genuine and imposter scores as compared to the baseline face recognition models. Table 4.1-4.2 and Table 4.7-4.18 shows improvement in FDR. The improvement in FDR shows better verification performance by the model as the separability of genuine and imposter scores increases. For example, Table 4.7 shows LFW evaluation results for the UMR-MP case using Resnet-101 and MobileFacenet backbone and our proposed MFU-DRU approach. We can clearly observe that FDR for the baseline model is 13.9739 (UMR-MP) and 21.8056 using our proposed MFU-DRU trained with Quadruplet loss (UMR-MP(Q)). We can observe, a similar improvement in FDR for the MR-MP cases too. This improvement in FDR can be observed in almost all evaluated backbone networks, scenarios, and datasets.

4.2.2 Impact of MFUM’s different architectures on masked face recognition performance

MFUM has improved the verification performance of masked face recognition models. Since we have developed different architectures of the MFUM for the ablation studies, therefore we have results for each architecture.

Masked Face Unveiling Dense Residual Unit (MFU-DRU)

When we consider the Resnet-101 backbone’s UMR-MP case for the LFW dataset processed by MFU-DRU (Table 4.7), we find out that, $FMR_{1000}\%$, $FMR_{100_Th}^{UMR-UMP}$ have improved from 1.0 to 0.7, 0.65 to 0.6 respectively. Whereas $EER\%$, $FMR_{100}\%$ has remained unchanged. When we consider the MR-MP case for the LFW dataset processed by MFU-DRU (Table 4.7), we find out that $EER\%$, $FMR_{100}\%$, $FMR_{1000}\%$, $FMR_{100_Th}^{UMR-UMP}$, $FMR_{1000_Th}^{UMR-UMP}$ have improved from 0.7 to 0.5, 0.7 to 0.5, 1.0 to 0.7, 1.2 to 1.15, and 0.7 to 0.65 respectively. Similarly, we can see observe improvements in these evaluation metrics on the MFR2 dataset as well as given in Table 4.8.

When we consider the MobileFacenet backbone’s UMR-MP case for the LFW dataset

processed by MFU-DRU (Table 4.7), we find out that, EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$. $FMR1000_Th^{UMR-UMP}$ have improved from 3.8 to 2.0, 6.4 to 2.6, 15.1 to 4.0, 3.75 to 1.8, and 6.0 to 2.1 respectively. When we consider the MR-MP case for the LFW dataset processed by MFU-DRU (Table 4.7), we find out that EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$. $FMR1000_Th^{UMR-UMP}$ have improved from 0.7 to 0.5, 0.7 to 0.5, 1.0 to 0.7, 1.2 to 1.15, and 0.7 to 0.65 respectively. Similarly, we can see observe improvements in these evaluation metrics on the MFR2 dataset as well as given in Table 4.8.

Masked Face Unveiling Attention Augmented Dense Residual Unit (MFU-ADRU)

When we consider the Resnet-101 backbone’s UMR-MP case for the LFW dataset processed by MFU-ADRU (Table 4.1), we find out that, EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$ and $FMR1000_Th^{UMR-UMP}$ have improved from 0.7 to 0.5, 0.7 to 0.3, 1.0 to 0.7, 0.65 to 0.6, and 0.6 to 0.4 respectively. There is improvement in all evaluation metrics. Further, when we consider MR-MP cases for the LFW dataset processed by MFU-ADRU (Table 4.1), we find out that EER%, FMR100%, and FMR1000% have improved from 0.7 to 0.5, 0.7 to 0.4, 1.0 to 0.9 to respectively. Whereas $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have not improved. Likewise, we can see observe similar improvements in evaluation metrics on the MFR2 dataset as given in Table 4.2.

When we consider the MobileFacenet backbone’s UMR-MP case for the LFW dataset processed by MFU-ADRU (Table 4.1), we find out that, EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$. $FMR1000_Th^{UMR-UMP}$ have improved from 3.8 to 2.1, 6.4 to 3.2, 15.1 to 7.6, 3.75 to 2.0, and 6.0 to 2.9 respectively. When we consider the MR-MP case for the LFW dataset processed by MFU-ADRU (Table 4.1), we find out that EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$. $FMR1000_Th^{UMR-UMP}$ have improved from 3.6 to 2.0, 6.6 to 2.5, 17.6 to 5.9, 4.7 to 3.6, and 3.9 to 1.85 respectively. Similarly, we can see observe improvements in evaluation metrics on the MFR2 dataset as well. The results are given in Table 4.2.

Masked Face Unveiling Conv Unit (MFU-CU)

When we consider the Resnet-101 backbone’s UMR-MP case for the LFW dataset processed by MFU-CU (Table 4.9), we find out that, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR100_Th^{UMR-UMP}$ have improved from 0.9 to 0.7, 0.55 to 0.5, 0.8 to 0.6 respectively. Whereas EER% and FMR100% have remained unchanged. When we consider the MR-MP case for the LFW dataset processed by MFU-CU (Table 4.9), we find out that EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 0.8 to 0.6, 0.8 to 0.6, 1.4 to 0.9, 2.1 to 1.8 and 0.75 to 0.65 respectively. Now let’s consider the MFR2 dataset as given in Table 4.10. For the UMR-MP case, we find out that EER%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 4.9534 to 4.3334, 4.2499 to 4.1955 and 5.4423 to 4.9931 respectively. Whereas FMR100% and FMR1000% remain unchanged. For the MR-MP case EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$ have improved from 8.8556 to 8.4923, 14.3911 to 14.0221, 22.5092 to 19.1882, 7.598 to 7.2541 respectively. Whereas $FMR1000_Th^{UMR-UMP}$ deteriorates from 7.3257 to 7.5404.

When we consider the MobileFacenet backbone’s UMR-MP case for the LFW dataset processed by MFU-CU (Table 4.9), we find out that, EER%, FMR100% have improved from 10.7 to 8.3, 29.6 to 24.4 respectively Whereas FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have not improved rather performance has deteriorated. When we consider the MR-MP case for the LFW dataset processed by MFU-CU (Table 4.9), we find out that EER%, FMR100% have improved from 11.5 to 8.2, 30.5 to 22.3 respectively whereas $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. Now let’s consider the MFR2 dataset as given in Table 4.10. For the UMR-MP case, we find out that EER% has improved from 16.0969 to 14.5501 whereas FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. For the MR-MP case, EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 20.7086 to 17.7112, 54.6125 to 52.0295, 30.6777 to 28.0889 and 36.5489 to 32.79 respectively whereas FMR1000% has deteriorated.

Masked Face Unveiling Conv Unit with Attention Mechanism in Configuration 1 (MFU-CUAM1)

When we consider the Resnet-101 backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM1 (Table 4.11), we find out that, FMR1000% and $FMR100_Th^{UMR-UMP}$ have improved from 0.9 to 0.7, 0.8 to 0.5 respectively. Whereas EER%, FMR100%, $FMR100_Th^{UMR-UMP}$ have remained unchanged. When we consider the MR-MP case for the LFW dataset processed by MFU-CUAM1 (Table 4.11), we find out that EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, and $FMR1000_Th^{UMR-UMP}$ have improved from 0.8 to 0.6, 0.8 to 0.6, 1.4 to 0.9, 2.1 to 1.95 and 0.75 to 0.7 respectively. Now let’s consider the MFR2 dataset as given in Table 4.12. For the UMR-MP case, we find out that EER%, $FMR1000_Th^{UMR-UMP}$ have improved from 4.9534 to 4.6434 and 5.4423 to 5.2997 respectively. Whereas FMR100% and FMR1000% remain unchanged. For MR-MP case EER%, $FMR100_Th^{UMR-UMP}$ have improved from 8.8556 to 8.4853 and 7.598 to 7.2343 respectively whereas FMR100% remains unchanged and FMR1000% and $FMR1000_Th^{UMR-UMP}$ deteriorate.

When we consider the MobileFacenet backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM1 (Table 4.11), we find out that, EER%, FMR100% have improved from 10.7 to 8.3, 29.6 to 24.4 respectively, whereas FMR1000%, $FMR100_Th^{UMR-UMP}$. $FMR1000_Th^{UMR-UMP}$ have not improved rather performance has deteriorated. When we consider the MR-MP case for the LFW dataset processed by MFU-CU (Table 4.11), we find out that EER% and FMR100% have improved from 11.5 to 8.2 and 30.5 to 22.3 respectively whereas FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. Now let’s consider the MFR2 dataset as given in Table 4.12. For the UMR-MP case, we find out that EER% has improved from 16.0969 to 14.8601 whereas FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. For the MR-MP case, EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 20.7086 to 18.4483, 54.6125 to 53.1365, 30.6777 to 23.7247 and 36.5489 to 27.0941 respectively whereas FMR1000% has deteriorated.

Masked Face Unveiling Conv Unit with Attention Mechanism in Configuration 2 (MFU-CUAM2)

When we consider the Resnet-101 backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM2 (Table 4.13), we find out that, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 0.9 to 0.5, 0.55 to 0.5 and 0.8 to 0.45 respectively. Whereas EER% and FMR100% have remained unchanged. When we consider the MR-MP case for the LFW dataset processed by MFU-CUAM2 (Table 4.13), we find out that EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 0.8 to 0.7, 0.8 to 0.6, 1.4 to 0.9, 2.1 to 1.6 and 0.75 to 0.7 respectively. Now let’s consider the MFR2 dataset as given in Table 4.14. For the UMR-MP case, we find out that EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 4.9534 to 4.446, 6.8111 to 6.5015, 4.2499 to 3.8707 and 5.4423 to 5.0175 respectively whereas FMR1000% deteriorates. For MR-MP case EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$ and $FMR1000_Th^{UMR-UMP}$ have improved from 8.8556 to 7.7482, 14.3911 to 12.1771, 22.5092 to 18.0812, 7.598 to 6.2917 and 7.3257 to 6.8923 respectively.

When we consider the MobileFacenet backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM2 (Table 4.13), we find out that, EER%, FMR100%, FMR1000% have improved from 10.7 to 7.6, 29.6 to 22.0, 43.1 to 42.1 respectively Whereas $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have not improved rather performance has deteriorated. When we consider the MR-MP case for the LFW dataset processed by MFU-CUAM2 (Table 4.13), we find out that EER% and FMR100% have improved from 11.5 to 8.0 and 30.5 to 20.7 respectively whereas FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. Now let’s consider the MFR2 dataset as given in Table 4.14. For the UMR-MP case, we find out that EER%, FMR100%, and FMR1000% have improved from 16.0969 to 14.5501, 42.7245 to 39.3189, 62.2291 to 60.6811 whereas $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. For the MR-MP case, EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 20.7086 to 16.6353, 54.6125 to 50.9225, 30.6777 to 25.9626 and 36.5489 to 32.7795 respectively whereas FMR1000% has deteriorated.

Masked Face Unveiling Conv Unit with Attention Mechanism in Configuration 3 (MFU-CUAM3)

When we consider the Resnet-101 backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM3 (Table 4.15), we find out that, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 0.9 to 0.5, 0.55 to 0.5 and 0.8 to 0.5 respectively. Whereas EER% and FMR100% have remained unchanged. When we consider the MR-MP case for the LFW dataset processed by MFU-CUAM3 (Table 4.15), we find out that EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 0.8 to 0.7, 0.8 to 0.6, 1.4 to 0.8, 2.1 to 1.75 and 0.75 to 0.6 respectively. Now let’s consider the MFR2 dataset as given in Table 4.16. For the UMR-MP case, we find out that EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 4.9534 to 4.4217, 6.8111 to 6.5015, 4.2499 to 3.8585 and 5.4423 to 4.8505 respectively whereas FMR1000% remains unchanged. For MR-MP case EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$ and $FMR1000_Th^{UMR-UMP}$ have improved from 8.8556 to 8.115, 14.3911 to 11.8081, 22.5092 to 18.4502, 7.598 to 6.5546 and 7.3257 to 6.5583 respectively.

When we consider the MobileFacenet backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM3 (Table 4.15), we find out that, EER%, FMR100%, and FMR1000% have improved from 10.7 to 7.1, 29.6 to 19.5, 43.1 to 34.8 respectively Whereas $FMR100_Th^{UMR-UMP}$. $FMR1000_Th^{UMR-UMP}$ have not improved rather performance has deteriorated. When we consider the MR-MP case for the LFW dataset processed by MFU-CUAM3 (Table 4.15), we find out that EER% and FMR100% have improved from 11.5 to 7.2 and 30.5 to 22.5 respectively. Whereas FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. Now let’s consider the MFR2 dataset as given in Table 4.16. For the UMR-MP case, we find out that EER%, FMR100%, and FMR1000% have improved from 16.0969 to 13.9302, 42.7245 to 37.1517, 62.2291 to 60.0619 respectively, whereas $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. For the MR-MP case, EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 20.7086 to 14.1, 54.6125 to 49.0775, 30.6777 to 25.3052 and 36.5489 to 31.5406 respectively whereas FMR1000% has deteriorated.

Masked Face Unveiling Conv Unit with Attention Mechanism in Configuration 4 (MFU-CUAM4)

When we consider the Resnet-101 backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM4 (Table 4.17), we find out that, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 0.9 to 0.5, 0.55 to 0.5 and 0.8 to 0.45 respectively. Whereas EER% and FMR100% have remained unchanged. When we consider the MR-MP case for the LFW dataset processed by MFU-CUAM4 (Table 4.17), we find out that EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 0.8 to 0.7, 0.8 to 0.6, 1.4 to 0.8, 2.1 to 1.75 and 0.75 to 0.6 respectively. Now let’s consider MFR2 dataset as given in Table 4.18. For the UMR-MP case, we find out that EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 4.9534 to 4.3334, 6.8111 to 6.5015, 4.2499 to 3.8403 and 5.4423 to 4.8566 respectively whereas FMR1000% deteriorates. For MR-MP case EER%, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$ and $FMR1000_Th^{UMR-UMP}$ have improved from 8.8556 to 6.8196, 14.3911 to 11.8081, 22.5092 to 18.0812, 7.598 to 6.474 and 7.3257 to 6.3633 respectively.

When we consider the MobileFacenet backbone’s UMR-MP case for the LFW dataset processed by MFU-CUAM4 (Table 4.17), we find out that, EER%, FMR100%, and FMR1000% have improved from 10.7 to 6.5, 29.6 to 23.0, 43.1 to 35.4 respectively. Whereas $FMR100_Th^{UMR-UMP}$. $FMR1000_Th^{UMR-UMP}$ have not improved rather performance has deteriorated. When we consider the MR-MP case for the LFW dataset processed by MFU-CUAM4 (Table 4.17), we find out that EER% and FMR100% have improved from 11.5 to 7.3 and 30.5 to 23.5 respectively whereas FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. Now let’s consider MFR2 dataset as given in Table 4.18. For the UMR-MP case, we find out that EER%, FMR100%, and FMR1000% have improved from 16.0969 to 13.7054, 42.7245 to 39.0093, 62.2291 to 58.8235 whereas $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have deteriorated. For the MR-MP case, EER%, FMR100%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$ have improved from 20.7086 to 14.0993, 54.6125 to 51.2915, 30.6777 to 25.9661 and 36.5489 to 32.0661 respectively whereas FMR1000% has deteriorated.

Table 4.1: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-ADRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-ADRU unit trained with Quadruplet loss approach.

LFW/ Back-bone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.7192	-0.0004	34.5764
	UMR-MP	0.7	0.7	1.0	0.5	0.8	0.65	0.1	1.1	0.6	0.5247	-0.0003	13.9739
	UMR-MP(T)	0.9	0.8	1.5	1.6	0.8	1.2	0.5	1.3	0.9	0.4574	0.002	14.775
	UMR-MP(SRT)	0.7	0.7	0.8	0.6	0.7	0.65	0.3	0.8	0.55	0.538	0.0025	17.0285
	UMR-MP(Q)	0.5	0.3	0.7	0.8	0.4	0.6	0.2	0.6	0.4	0.5638	-0.0009	20.3516
	MR-MP	0.7	0.7	1.0	1.8	0.6	1.2	0.7	0.7	0.7	0.606	0.0089	15.5591
	MR-MP(T)	1.1	1.2	1.8	5.0	0.7	2.85	3.1	0.7	1.9	0.6378	0.0063	14.1559
	MR-MP(SRT)	0.7	0.6	0.8	2.7	0.6	1.65	1.6	0.6	1.1	0.6506	0.0039	17.5434
MR-MP(Q)	0.5	0.4	0.9	3.3	0.4	1.85	1.2	0.4	0.8	0.6651	0.0007	20.7604	
Mobile-Facenet	UMR-UMP	0.7	0.7	1.0	1.0	0.7	0.85	0.1	1.0	0.55	0.6838	0.0062	19.218
	UMR-MP	3.8	6.4	15.1	1.1	6.4	3.75	0.3	11.7	6.0	0.4464	0.0035	6.4567
	UMR-MP(T)	2.6	3.4	9.3	1.1	3.2	2.15	0.3	6.5	3.4	0.4558	0.0027	9.3343
	UMR-MP(SRT)	2.2	3.1	19.1	1.4	2.9	2.15	0.6	6.5	3.55	0.455	0.0049	8.9363
	UMR-MP(Q)	2.1	3.2	7.6	1.4	2.6	2.0	0.6	5.2	2.9	0.4741	0.0037	9.6284
	MR-MP	3.6	6.6	17.6	7.6	1.8	4.7	3.1	4.7	3.9	0.5595	0.0508	7.3776
	MR-MP(T)	2.8	4.6	7.1	6.7	1.3	4.0	2.8	2.6	2.7	0.629	0.0039	9.1261
	MR-MP(SRT)	2.6	4.1	9.9	5.3	1.3	3.3	2.7	2.2	2.45	0.634	0.003	9.3476
MR-MP(Q)	2.0	2.5	5.9	5.8	1.4	3.6	1.7	2.0	1.85	0.635	0.0016	10.038	

Table 4.2: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-ADRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-ADRU unit trained with the Quadruplet loss approach.

MFR2/ Back-bone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.7585	0.0018	45.0473
	UMR-MP	4.6434	7.1207	8.6687	1.0284	7.1207	4.0745	0.1339	8.0495	4.0917	0.4415	0.0005	7.4826
	UMR-MP(T)	4.0904	6.192	10.2167	1.2048	5.8824	3.5436	0.2312	8.6687	4.45	0.4071	-0.0002	8.3382
	UMR-MP(SRT)	3.7135	6.8111	7.4303	1.2535	6.5015	3.8775	0.2738	7.4303	3.8521	0.4694	-0.0004	8.5411
	UMR-MP(Q)	2.5832	4.0248	6.5015	1.3326	3.7152	2.5239	0.2738	5.5728	2.9233	0.511	-0.0008	10.9242
	MR-MP	9.2224	12.5461	15.8672	4.5152	9.5941	7.0546	1.4303	11.8081	6.6192	0.546	0.0177	5.0717
	MR-MP(T)	8.4853	12.9151	20.6642	6.6466	10.3321	8.4893	2.8956	11.8081	7.3519	0.5593	0.0042	4.5812
	MR-MP(SRT)	9.2224	12.1771	14.7601	4.9148	9.5941	7.2545	1.7949	10.7011	6.248	0.5698	0.0063	5.1735
	MR-MP(Q)	8.115	11.0701	13.6531	4.2558	8.4871	6.3714	1.3812	10.3321	5.8566	0.5845	0.0072	5.837
Mobile-Facenet	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.7512	0.0057	30.2234
	UMR-MP	7.737	15.7895	29.7214	0.6876	17.3375	9.0125	0.0791	30.031	15.055	0.3788	-0.0029	4.5255
	UMR-MP(T)	7.737	12.6935	26.0062	1.0344	12.3839	6.7092	0.1156	26.0062	13.0609	0.3928	-0.0032	4.9623
	UMR-MP(SRT)	4.9534	7.1207	11.4551	1.077	6.8111	3.9441	0.0974	12.6935	6.3954	0.4354	0.001	7.8671
	UMR-MP(Q)	4.3334	6.192	13.0031	1.217	5.5728	3.3949	0.146	11.4551	5.8006	0.4565	0.0009	8.2587
	MR-MP	9.5927	21.4022	31.7343	6.268	11.8081	9.038	1.4934	17.7122	9.6028	0.5005	0.0597	3.7232
	MR-MP(T)	9.2224	21.4022	34.3173	5.7001	10.3321	8.0161	1.9491	15.4982	8.7236	0.5271	0.0105	3.8452
	MR-MP(SRT)	8.115	16.9742	29.5203	5.672	8.4871	7.0796	1.914	13.6531	7.7836	0.5422	0.0057	4.8195
	MR-MP(Q)	7.3779	16.6052	30.9963	5.3004	8.4871	6.8938	1.7458	14.0221	7.884	0.5356	0.0073	4.8552

4.3 Comparison with State of the Art

We have compared the performance of our proposed models with state-of-the-art models’ performance on LFW and MFR2 datasets against UMR-MP and MR-MP experimental settings. The comparisons are shown in Table 4.3-4.6.

Table 4.3: Comparison of our approaches with SoTA approaches for Unmasked Reference and Masked Probe experimental setting. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.

UMR-MP LFW	EER%	FMR100%	FMR1000%	FDR
ArcFace [49]	0.7	0.7	1	13.9739
EUM [44]	0.8667	0.8667	1.6	15.0505
MFU-DRU (Ours)	0.7	0.7	0.7	21.8056
MFU-CU (Ours)	0.5	0.5	0.7	14.2756
MFU-CUAM1 (Ours)	0.5	0.5	0.7	14.3179
MFU-CUAM2 (Ours)	0.5	0.5	0.5	13.8307
MFU-CUAM3 (Ours)	0.5	0.5	0.5	13.5788
MFU-CUAM4 (Ours)	0.5	0.5	0.5	13.54
MFU-ADRU (Ours)	0.5	0.3	0.7	20.3516

Table 4.4: Comparison of our approaches with SoTA approaches for Masked Reference and Masked Probe experimental setting. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.

MR-MP	EER%	FMR100%	FMR1000%	FDR
ArcFace [49]	0.7	0.7	1	15.5591
EUM [44]	0.9667	0.9667	2.0667	14.6018
MFU-DRU (Ours)	0.5	0.5	0.7	19.8189
MFU-CU (Ours)	0.6	0.6	0.9	13.7653
MFU-CUAM1 (Ours)	0.6	0.6	0.9	13.6669
MFU-CUAM2 (Ours)	0.7	0.6	0.9	13.864
MFU-CUAM3 (Ours)	0.7	0.6	0.8	13.8477
MFU-CUAM4 (Ours)	0.7	0.6	0.8	13.7269
MFU-ADRU (Ours)	0.5	0.4	0.9	20.7604

Table 4.5: Comparison of our approaches with SoTA approaches for Unmasked Reference and Masked Probe experimental setting on MFR2 dataset. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.

UMR-MP MFR2	EER%	FMR100%	FMR1000%	FDR
ArcFace [49]	6.6434	7.1207	8.6687	7.4826
EUM [44]	5.0507	6.5015	8.9783	7.2588
MFU-DRU (Ours)	3.7135	4.644	6.192	8.5917
MFU-CU (Ours)	4.3334	6.8111	9.9071	6.8436
MFU-CUAM1 (Ours)	4.6434	8.0495	10.5263	6.7136
MFU-CUAM2 (Ours)	4.446	6.5015	10.5263	6.9833
MFU-CUAM3 (Ours)	4.4217	6.5015	9.5975	6.9414
MFU-CUAM4 (Ours)	4.3334	6.5015	10.8359	6.9691
MFU-ADRU (Ours)	2.5832	4.0248	6.5015	10.9242

Table 4.6: Comparison of our approaches with SoTA approaches for Masked Reference and Masked Probe experimental setting on MFR2 dataset. We can clearly observe our proposed MFUM-based approaches with Resnet-101 backbone are performing better than existing SoTA approaches.

MR-MP	EER%	FMR100%	FMR1000%	FDR
ArcFace [49]	9.2224	12.5461	15.8672	5.0717
EUM [44]	9.296	13.2841	14.7601	4.9276
MFU-DRU (Ours)	7.4831	12.5461	15.4982	5.1486
MFU-CU (Ours)	8.4923	14.0221	19.9262	4.42
MFU-CUAM1 (Ours)	8.8453	15.1292	22.8782	4.384
MFU-CUAM2 (Ours)	7.7482	12.1771	18.0812	4.7578
MFU-CUAM3 (Ours)	8.115	11.8081	18.8192	4.7094
MFU-CUAM4 (Ours)	8.115	11.8081	18.8192	4.717
MFU-ADRU (Ours)	8.115	11.0701	13.6531	5.837

Table 4.7: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-DRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-DRU unit trained with the Quadruplet loss approach.

LFW/ Back-bone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.7192	-0.0004	34.5764
	UMR-MP(T)	1.6	1.8	3.8	0.7	2.4	1.55	0.2	3.6	1.9	0.3958	0.0004	10.751
	UMR-MP	0.7	0.7	1.0	0.5	0.8	0.65	0.1	1.1	0.6	0.5247	-0.0003	13.9739
	UMR-MP(SRT)	0.7	0.7	0.7	1.1	0.6	0.85	0.4	0.7	0.55	0.5922	0.0027	19.873
	UMR-MP(Q)	0.7	0.7	0.7	0.5	0.7	0.6	0.6	0.7	0.65	0.6078	0.0002	21.8056
	MR-MP(T)	1.9	3.2	9.1	27.7	0.4	14.05	21.3	0.4	10.85	0.6314	0.1026	9.8238
	MR-MP	0.7	0.7	1.0	1.8	0.6	1.2	0.7	0.7	0.7	0.606	0.0089	15.5591
	MR-MP(SRT)	0.6	0.5	0.7	3.7	0.5	2.1	1.9	0.5	1.2	0.6426	0.017	18.3048
MR-MP(Q)	0.5	0.5	0.7	1.8	0.5	1.15	0.8	0.5	0.65	0.6471	0.0142	19.8189	
Mobile-Facenet	UMR-UMP	0.7	0.7	1.0	1.0	0.7	0.85	0.1	1.0	0.55	0.6838	0.0062	19.218
	UMR-MP(T)	6.1	19.0	36.0	1.0	19.5	10.25	0.0	36.7	18.35	0.2989	-0.0016	4.929
	UMR-MP	3.8	6.4	15.1	1.1	6.4	3.75	0.3	11.7	6.0	0.4464	0.0035	6.4567
	UMR-MP(SRT)	2.0	2.6	4.0	1.6	2.3	1.95	0.3	3.9	2.1	0.5293	0.0064	10.8491
	UMR-MP(Q)	2.0	2.8	4.4	1.5	2.1	1.8	0.2	4.3	2.25	0.522	0.0047	10.4526
	MR-MP(T)	7.6	28.9	66.6	92.5	0.1	46.3	84.0	0.2	42.1	0.7253	0.3937	4.1969
	MR-MP	3.6	6.6	17.6	7.6	1.8	4.7	3.1	4.7	3.9	0.5595	0.0508	7.3776
	MR-MP(SRT)	2.2	2.8	3.7	3.1	1.5	2.3	0.8	2.8	1.8	0.5918	0.0066	10.1154
MR-MP(Q)	2.1	3.2	4.5	3.5	1.4	2.45	1.3	2.6	1.95	0.5953	0.0195	9.8732	

Table 4.8: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-DRU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP. Significant improvement in verification performance can be observed clearly by our proposed MFU-DRU unit trained with Quadruplet loss approach.

MFR2/ Back-bone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR100_{Th}^{UMR-UMP}$			$FMR1000_{Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.7585	0.0018	45.0473
	UMR-MP(T)	5.9137	9.5975	11.7647	0.8945	9.5975	5.246	0.1156	10.8359	5.4758	0.3571	0.0031	6.3172
	UMR-MP	4.6434	7.1207	8.6687	1.0284	7.1207	4.0745	0.1339	8.0495	4.0917	0.4415	0.0005	7.4826
	UMR-MP(SRT)	3.7135	4.9536	8.3591	1.3083	4.644	2.9761	0.1765	6.5015	3.339	0.4622	0.0048	8.3242
	UMR-MP(Q)	4.0235	4.644	6.192	1.3143	4.644	2.9792	0.1825	6.192	3.1872	0.4746	0.0026	8.5917
	MR-MP(T)	9.963	17.7122	23.2472	25.9412	6.2731	16.1072	13.4754	8.4871	10.9813	0.5697	0.1023	4.2514
	MR-MP	9.2224	12.5461	15.8672	4.5152	9.5941	7.0546	1.4303	11.8081	6.6192	0.546	0.0177	5.0717
	MR-MP(SRT)	7.4831	12.5461	15.4982	6.4503	8.4871	7.4687	2.0823	11.4391	6.7607	0.5569	0.0374	5.287
MR-MP(Q)	8.115	12.1771	15.4982	5.4617	8.8561	7.1589	1.6546	10.3321	5.9934	0.545	0.0313	5.1486	
Mobile-Facenet	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.7512	0.0057	30.2234
	UMR-MP(T)	8.9768	25.0774	48.9164	0.4442	34.3653	17.4048	0.0061	65.9443	32.9752	0.274	0.0022	3.3391
	UMR-MP	7.730	15.7895	29.7214	0.6876	17.3375	9.0125	0.0791	30.031	15.055	0.3788	-0.0029	4.5255
	UMR-MP(SRT)	8.0469	13.0031	23.2198	0.8458	13.0031	6.9245	0.1095	23.2198	11.6647	0.4121	-0.0013	4.9884
	UMR-MP(Q)	7.721	12.3839	21.0526	0.8397	13.0031	6.9214	0.0974	21.3622	10.7298	0.4104	0.0003	4.9714
	MR-MP(T)	15.1962	55.7196	90.7749	90.8294	1.476	46.1527	75.2296	2.952	39.0908	0.6777	0.413	1.9174
	MR-MP	9.5927	21.4022	31.7343	6.268	11.8081	9.038	1.4934	17.7122	9.6028	0.5005	0.0597	3.7232
	MR-MP(SRT)	9.963	17.7122	29.1513	3.6598	13.2841	8.472	0.6871	19.9262	10.3066	0.4921	0.0243	3.8141
MR-MP(Q)	9.0168	16.9742	31.7343	4.4871	11.4391	7.9631	0.9816	17.3432	9.1624	0.499	0.0386	3.7944	

Table 4.9: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ Conv Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G-mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	0.8	0.8	1.2	0.1	1.5	0.8	0.0	3.8	1.9	0.3337	0.0004	11.4474
	UMR-MP(SRT)	0.5	0.5	0.7	0.6	0.5	0.55	0.0	1.0	0.5	0.4322	0.011	14.2613
	UMR-MP(Q)	0.5	0.5	0.7	0.5	0.5	0.5	0.0	1.2	0.6	0.4321	0.0061	14.2756
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	0.8	0.8	1.3	86.7	0.0	43.35	70.9	0.1	35.5	0.6039	0.2017	11.1982
	MR-MP(SRT)	0.7	0.6	1.0	3.0	0.6	1.8	0.6	0.7	0.65	0.5109	0.0168	13.7779
	MR-MP(Q)	0.6	0.6	0.9	3.4	0.6	2.0	0.6	0.7	0.65	0.512	0.0197	13.7653
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	8.7	28.0	49.2	0.0	99.6	49.8	0.0	100.0	50.0	0.3133	0.0342	4.2071
	UMR-MP(SRT)	8.4	26.4	48.1	0.0	99.4	49.7	0.0	100.0	50.0	0.324	0.0393	4.2478
	UMR-MP(Q)	8.3	24.4	53.3	0.0	99.0	49.5	0.0	100.0	50.0	0.3282	0.0322	4.3024
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	9.3	27.9	54.6	1.9	21.2	11.55	0.4	33.8	17.1	0.6098	0.1531	4.1542
	MR-MP(SRT)	9.1	25.7	55.1	1.4	23.2	12.3	0.4	37.1	18.75	0.5994	0.1264	4.2166
	MR-MP(Q)	8.2	22.3	59.7	0.5	30.1	15.3	0.2	43.7	21.95	0.5747	0.0697	4.3012

Table 4.10: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CU trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ Back-bone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	5.5702	7.7399	15.4799	0.2799	10.8359	5.5579	0.0304	17.6471	8.8387	0.2714	0.0003	6.3106
	UMR-MP(Q)	4.6434	7.1207	9.5975	0.5537	8.0495	4.3016	0.0791	9.9071	4.9931	0.348	0.0075	6.8436
	UMR-MP(SRT)	4.3334	6.8111	9.9071	0.6511	7.7399	4.1955	0.0913	9.9071	4.9992	0.3519	0.0117	6.8807
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	7.8499	15.4982	21.7712	95.2535	0.0	47.6267	76.4916	0.0	38.2458	0.561	0.2255	4.6749
	MR-MP(Q)	8.8556	14.0221	19.9262	4.2137	10.7011	7.4574	1.0587	14.0221	7.5404	0.4402	0.0308	4.4297
	MR-MP(SRT)	8.4923	14.0221	19.1882	3.8071	10.7011	7.2541	0.9185	14.3911	7.6548	0.4398	0.0267	4.5067
	Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264
UMR-MP		16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
UMR-MP(T)		15.2065	43.9628	68.7307	0.0	100.0	50.0	0.0	100.0	50.0	0.2686	0.033	2.1528
UMR-MP(SRT)		14.5501	43.9628	67.4923	0.0	100.0	50.0	0.0	100.0	50.0	0.2807	0.0326	2.3004
UMR-MP(Q)		15.5013	43.0341	65.0155	0.0	100.0	50.0	0.0	100.0	50.0	0.2789	0.0393	2.1697
MR-MP		20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
MR-MP(T)		17.3409	52.3985	73.8007	1.1428	51.6605	26.4017	0.3365	62.7306	31.5336	0.5359	0.1774	2.1144
MR-MP(SRT)		18.078	53.5055	72.6937	0.3646	62.3616	31.3631	0.0771	73.0627	36.5699	0.4794	0.1005	2.1518
MR-MP(Q)	17.7112	52.0295	71.9557	0.8273	55.3506	28.0889	0.2664	65.3137	32.79	0.5215	0.1551	2.1264	

Table 4.11: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM1 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ Conv with CBAM A1 Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	1.2	1.3	2.3	0.2	1.9	1.05	0.0	4.8	2.4	0.3103	0.0003	10.3382
	UMR-MP(SRT)	0.5	0.5	0.7	0.5	0.6	0.55	0.0	1.3	0.65	0.4283	0.0067	14.3817
	UMR-MP(Q)	0.5	0.5	0.7	0.7	0.5	0.6	0.0	1.0	0.5	0.4299	0.0121	14.3179
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	1.3	1.6	5.7	97.0	0.0	48.5	90.0	0.0	45.0	0.6165	0.2634	8.7256
	MR-MP(SRT)	0.6	0.6	0.9	3.6	0.6	2.1	0.8	0.6	0.7	0.5023	0.0233	13.7266
MR-MP(Q)	0.6	0.6	0.9	3.3	0.6	1.95	0.8	0.6	0.7	0.5009	0.0206	13.6669	
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	8.7	28.0	49.2	0.0	99.6	49.8	0.0	100.0	50.0	0.3133	0.0342	4.2071
	UMR-MP(SRT)	8.4	26.4	48.1	0.0	99.4	49.7	0.0	100.0	50.0	0.324	0.0393	4.2478
	UMR-MP(Q)	8.3	24.4	53.3	0.0	99.0	49.5	0.0	100.0	50.0	0.3282	0.0322	4.3024
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	9.3	27.9	54.6	1.9	21.2	11.55	0.4	33.8	17.1	0.6098	0.1531	4.1542
	MR-MP(SRT)	9.1	25.7	55.1	1.4	23.2	12.3	0.4	37.1	18.75	0.5994	0.1264	4.2166
MR-MP(Q)	8.2	22.3	59.7	0.5	30.1	15.3	0.2	43.7	21.95	0.5747	0.0697	4.3012	

Table 4.12: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM1 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ Conv with CBAM A1 / Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	5.5702	8.6687	17.3375	0.2495	15.1703	7.7099	0.0243	19.8142	9.9193	0.2455	0.0006	5.6123
	UMR-MP(SRT)	4.9534	8.0495	10.2167	0.5963	8.3591	4.4777	0.073	10.5263	5.2997	0.3409	0.0084	6.7055
	UMR-MP(Q)	4.6434	8.0495	10.5263	0.7119	8.0495	4.3807	0.1034	10.5263	5.3149	0.3469	0.0131	6.7136
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	7.7482	16.9742	27.6753	99.9439	0.0	49.972	98.3664	0.0	49.1832	0.5805	0.2964	4.4168
	MR-MP(SRT)	8.8556	14.3911	23.2472	4.4941	10.3321	7.4131	1.227	14.0221	7.6245	0.428	0.0353	4.3564
MR-MP(Q)	8.4853	15.1292	22.8782	4.1366	10.3321	7.2343	1.0657	14.7601	7.9129	0.4278	0.0318	4.384	
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	17.0268	52.0124	75.8514	0.0	100.0	50.0	0.0	100.0	50.0	0.2721	0.047	1.8725
	UMR-MP(SRT)	16.7168	46.7492	69.6594	0.0	100.0	50.0	0.0	100.0	50.0	0.2836	0.0497	2.0096
	UMR-MP(Q)	14.8601	46.7492	68.7307	0.0	100.0	50.0	0.0	100.0	50.0	0.2717	0.0269	2.2151
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	19.5556	54.2435	75.6458	3.169	44.2804	23.7247	1.0517	53.1365	27.0941	0.5678	0.24	1.6879
	MR-MP(SRT)	18.8186	53.1365	73.4317	1.3391	49.0775	25.2083	0.4137	63.0996	31.7566	0.5341	0.1845	1.8602
	MR-MP(Q)	18.4483	54.9815	74.9077	0.3996	63.0996	31.7496	0.0911	74.9077	37.4994	0.4691	0.0988	1.9701

Table 4.13: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM2 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ Conv with CBAM A2 Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	0.9	0.8	1.6	0.4	1.3	0.85	0.0	2.7	1.35	0.3652	0.0027	10.8084
	UMR-MP(SRT)	0.5	0.5	0.6	0.6	0.5	0.55	0.0	1.4	0.7	0.4278	0.0084	12.6388
	UMR-MP(Q)	0.5	0.5	0.5	0.5	0.5	0.5	0.0	0.9	0.45	0.44	0.0106	13.8307
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	1.0	1.0	2.6	78.2	0.1	39.15	57.5	0.1	28.8	0.5753	0.177	10.8332
	MR-MP(SRT)	0.8	0.8	1.1	3.2	0.6	1.9	0.7	0.8	0.75	0.5027	0.0237	13.204
MR-MP(Q)	0.7	0.6	0.9	2.6	0.6	1.6	0.7	0.7	0.7	0.5199	0.0137	13.864	
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	8.7	29.4	56.8	0.0	95.8	47.9	0.0	99.6	49.8	0.3443	0.0521	4.0412
	UMR-MP(SRT)	8.8	29.4	53.6	0.0	94.9	47.45	0.0	99.3	49.65	0.3537	0.0563	4.1857
	UMR-MP(Q)	7.6	22.0	42.1	0.0	93.7	46.85	0.0	98.9	49.45	0.3583	0.0405	4.5965
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	9.3	28.6	56.9	1.3	25.2	13.25	0.4	39.8	20.1	0.5897	0.1914	3.8694
	MR-MP(SRT)	9.0	26.4	56.2	0.8	27.9	14.35	0.2	44.4	22.3	0.5762	0.1658	4.0642
MR-MP(Q)	8.0	20.7	47.8	0.2	39.7	19.95	0.1	55.6	27.85	0.5349	0.079	4.5034	

Table 4.14: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM2 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ Conv with CBAM A2 / Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	4.9534	7.7399	14.5511	0.359	10.5263	5.4427	0.0669	16.4087	8.2378	0.295	0.0025	6.2059
	UMR-MP(SRT)	4.6434	6.8111	9.9071	0.5781	7.4303	4.0042	0.0791	10.2167	5.1479	0.354	0.01	6.8005
	UMR-MP(Q)	4.446	6.5015	10.5263	0.6207	7.1207	3.8707	0.1278	9.9071	5.0175	0.3656	0.0111	6.9833
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	7.913	15.1292	22.1402	90.6822	0.0	45.3411	62.0767	0.738	31.4074	0.533	0.2034	4.6476
	MR-MP(SRT)	8.4853	13.6531	19.9262	4.417	9.2251	6.8211	1.0517	13.6531	7.3524	0.4405	0.0366	4.5365
MR-MP(Q)	7.7482	12.1771	18.0812	3.3583	9.2251	6.2917	0.8694	12.9151	6.8923	0.4504	0.0213	4.7578	
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	15.7869	45.5108	62.2291	0.0	100.0	50.0	0.0	100.0	50.0	0.2988	0.0493	2.158
	UMR-MP(SRT)	15.17	43.6533	61.6099	0.0	100.0	50.0	0.0	100.0	50.0	0.3078	0.0539	2.2196
	UMR-MP(Q)	14.5501	39.3189	60.6811	0.0	100.0	50.0	0.0	100.0	50.0	0.3032	0.0367	2.4534
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	17.4986	50.9225	70.8487	1.0026	50.9225	25.9626	0.2454	65.3137	32.7795	0.5334	0.2136	1.945
	MR-MP(SRT)	16.6353	50.9225	69.0037	0.617	55.7196	28.1683	0.1052	69.0037	34.5544	0.5163	0.188	2.0299
	MR-MP(Q)	16.751	53.5055	70.1107	0.0982	70.4797	35.2889	0.0	81.9188	40.9594	0.4492	0.0974	2.2349

Table 4.15: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM3 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ Conv with CBAM A3 Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	0.9	0.8	1.4	0.4	1.3	0.85	0.0	2.7	1.35	0.3593	0.0023	11.1187
	UMR-MP(SRT)	0.5	0.5	0.8	0.8	0.5	0.65	0.0	1.0	0.5	0.4391	0.0102	13.6606
	UMR-MP(Q)	0.5	0.5	0.5	0.5	0.5	0.5	0.0	1.3	0.65	0.437	0.0073	13.5788
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	0.9	0.9	1.6	81.0	0.1	40.55	61.5	0.1	30.8	0.5791	0.1819	10.9918
	MR-MP(SRT)	0.7	0.7	1.2	2.9	0.6	1.75	0.5	0.7	0.6	0.5158	0.0128	13.3722
	MR-MP(Q)	0.7	0.6	0.8	2.9	0.6	1.75	0.6	0.7	0.65	0.5169	0.0175	13.8477
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	8.0	28.9	54.1	0.0	96.7	48.35	0.0	99.8	49.9	0.3382	0.045	4.3615
	UMR-MP(SRT)	7.7	24.3	48.1	0.0	95.0	47.5	0.0	99.5	49.75	0.357	0.054	4.6482
	UMR-MP(Q)	7.1	19.5	34.8	0.0	96.0	48.0	0.0	99.5	49.75	0.3462	0.0293	4.9704
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	8.6	31.0	55.2	1.7	24.8	13.25	0.7	37.7	19.2	0.5993	0.1681	4.2025
	MR-MP(SRT)	8.3	26.5	52.7	1.2	25.9	13.55	0.3	41.9	21.1	0.5848	0.1331	4.5824
	MR-MP(Q)	7.2	22.5	45.3	0.3	36.0	18.15	0.1	50.9	25.5	0.5517	0.0648	5.0597

Table 4.16: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM3 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ Conv with CBAM A3 / Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	4.9534	7.4303	14.8607	0.3286	10.8359	5.5823	0.0487	17.0279	8.5383	0.2909	0.002	6.1844
	UMR-MP(SRT)	4.6434	6.5015	9.5975	0.6146	7.1207	3.8677	0.1034	9.5975	4.8505	0.3643	0.0105	6.9058
	UMR-MP(Q)	4.4217	6.5015	10.2167	0.5963	7.1207	3.8585	0.1156	10.2167	5.1662	0.3608	0.0085	6.9414
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	8.115	15.4982	22.8782	91.9933	0.0	45.9966	64.2081	0.369	32.2885	0.5358	0.2063	4.6025
	MR-MP(SRT)	8.8556	14.0221	18.4502	3.2812	10.7011	6.9912	0.8133	14.0221	7.4177	0.4477	0.0205	4.557
MR-MP(Q)	8.115	11.8081	18.8192	3.8842	9.2251	6.5546	0.9395	12.1771	6.5583	0.4492	0.028	4.7094	
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	14.8601	43.9628	62.5387	0.0	100.0	50.0	0.0	100.0	50.0	0.295	0.0444	2.2913
	UMR-MP(SRT)	13.9302	40.5573	61.6099	0.0	100.0	50.0	0.0	100.0	50.0	0.3136	0.0549	2.4565
	UMR-MP(Q)	14.2401	37.1517	60.0619	0.0	100.0	50.0	0.0	100.0	50.0	0.2907	0.0256	2.567
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	16.6038	51.6605	74.1697	1.1639	49.4465	25.3052	0.3506	62.7306	31.5406	0.5455	0.2023	2.1583
	MR-MP(SRT)	15.4964	50.5535	72.3247	0.7221	51.6605	26.1913	0.1823	67.1587	33.6705	0.5277	0.1681	2.3467
MR-MP(Q)	14.0187	49.0775	70.8487	0.1683	67.8967	34.0325	0.021	78.2288	39.1249	0.4707	0.0934	2.6505	

Table 4.17: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM4 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ Conv with CBAM A4 Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	0.8	0.7	1.3	0.0	1.3	0.65	0.0	2.6	1.3	0.3582	0.0017	11.6027
	UMR-MP(SRT)	0.5	0.5	0.7	0.7	0.5	0.6	0.0	0.9	0.45	0.4445	0.0105	14.0422
	UMR-MP(Q)	0.5	0.5	0.5	0.5	0.5	0.5	0.0	1.3	0.65	0.4352	0.0072	13.5478
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	0.7	0.7	2.1	80.0	0.1	40.05	59.7	0.1	29.9	0.5823	0.1789	11.6389
	MR-MP(SRT)	0.7	0.6	1.1	2.9	0.6	1.75	0.5	0.7	0.6	0.5182	0.0131	13.781
MR-MP(Q)	0.7	0.6	0.8	2.9	0.6	1.75	0.6	0.7	0.65	0.5194	0.016	13.7269	
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	8.5	27.5	56.2	0.0	96.7	48.35	0.0	99.9	49.95	0.3346	0.043	4.3446
	UMR-MP(SRT)	7.6	23.0	47.7	0.0	96.3	48.15	0.0	99.7	49.85	0.3477	0.0438	4.7906
	UMR-MP(Q)	6.5	24.2	35.4	0.0	96.3	48.15	0.0	99.7	49.85	0.3414	0.0283	4.9505
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	9.1	25.6	57.9	1.0	25.7	13.35	0.4	40.4	20.4	0.5896	0.1633	4.1876
	MR-MP(SRT)	8.3	23.5	56.3	0.5	32.9	16.7	0.2	48.8	24.5	0.5618	0.1043	4.7481
MR-MP(Q)	7.3	23.5	49.4	0.2	41.4	20.8	0.1	57.0	28.55	0.5333	0.057	4.6642	

Table 4.18: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM4 trained with Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ Conv with CBAM A4 / Backbone	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	4.7438	7.4303	15.4799	0.3408	10.5263	5.4335	0.0487	17.0279	8.5383	0.2896	0.0019	6.2469
	UMR-MP(SRT)	4.4764	6.5015	9.9071	0.6328	7.1207	3.8768	0.1156	9.5975	4.8566	0.3673	0.0106	6.9394
	UMR-MP(Q)	4.3334	6.5015	10.8359	0.5598	7.1207	3.8403	0.1034	10.2167	5.1601	0.3605	0.0081	6.9691
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	6.8196	15.4982	22.8782	91.4744	0.0	45.7372	62.6586	0.369	31.5138	0.5365	0.2048	4.7343
	MR-MP(SRT)	8.4853	13.6531	18.0812	3.2672	9.9631	6.6151	0.7642	14.0221	7.3932	0.4471	0.0206	4.6107
MR-MP(Q)	8.115	11.8081	18.8192	3.7229	9.2251	6.474	0.9185	11.8081	6.3633	0.4508	0.0256	4.717	
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	15.17	45.5108	63.1579	0.0	100.0	50.0	0.0	100.0	50.0	0.2934	0.0417	2.304
	UMR-MP(SRT)	14.0762	40.5573	58.8235	0.0	100.0	50.0	0.0	100.0	50.0	0.3049	0.0445	2.4958
	UMR-MP(Q)	13.7054	39.0093	61.3003	0.0	100.0	50.0	0.0	100.0	50.0	0.2897	0.0246	2.5829
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	17.3409	51.2915	73.4317	1.0096	50.9225	25.9661	0.2945	63.8376	32.0661	0.5383	0.1952	2.1652
	MR-MP(SRT)	14.7593	52.3985	72.3247	0.4627	60.5166	30.4897	0.0982	72.3247	36.2114	0.504	0.1379	2.4603
MR-MP(Q)	14.0993	55.3506	74.1697	0.1613	71.2177	35.6895	0.021	81.5498	40.7854	0.4583	0.0854	2.5408	

Table 4.19: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-DRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-DRU trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	0.7	0.7	1.0	0.5	0.8	0.65	0.1	1.1	0.6	0.5247	-0.0003	13.9739
	UMR-MP(T)	42.0	96.5	99.5	0.0	100.0	50.0	0.0	100.0	50.0	0.0247	0.0061	0.0736
	UMR-MP(SRT)	41.3	97.8	99.1	0.0	99.5	49.75	0.0	99.9	49.95	0.0214	-0.0033	0.1038
	UMR-MP(Q)	37.1	94.3	99.0	0.3	98.9	49.6	0.0	99.7	49.85	0.0383	0.0007	0.2128
	MR-MP	0.7	0.7	1.0	1.8	0.6	1.2	0.7	0.7	0.7	0.606	0.0089	15.5591
	MR-MP(T)	5.1	13.0	24.9	100.0	0.0	50.0	100.0	0.0	50.0	0.8884	0.769	5.0668
	MR-MP(SRT)	5.2	12.9	36.2	100.0	0.0	50.0	100.0	0.0	50.0	0.8889	0.7724	4.8849
MR-MP(Q)	5.1	11.3	16.3	100.0	0.0	50.0	100.0	0.0	50.0	0.9052	0.7902	5.1355	
Mobile-Facenet	UMR-UMP	3.1	5.5	13.7	1.0	5.5	3.25	0.1	13.7	6.9	0.5543	0.0494	7.415
	UMR-MP	9.1	24.3	37.0	1.0	25.2	13.1	0.0	51.4	25.7	0.3723	0.0495	3.6
	UMR-MP(T)	40.7	97.6	99.4	0.0	99.8	49.9	0.0	100.0	50.0	0.043	0.0056	0.09
	UMR-MP(SRT)	38.7	96.6	99.6	0.2	99.5	49.85	0.0	100.0	50.0	0.0518	0.0002	0.1621
	UMR-MP(Q)	34.7	92.6	98.8	0.2	98.8	49.5	0.0	99.9	49.95	0.0754	0.0011	0.3117
	MR-MP	9.4	29.6	54.1	8.0	9.9	8.95	1.7	24.6	13.15	0.4822	0.1181	3.4275
	MR-MP(T)	26.6	77.1	92.2	75.9	3.3	39.6	59.0	9.5	34.25	0.6019	0.4046	0.8014
	MR-MP(SRT)	25.0	76.6	85.4	84.1	1.8	42.95	65.7	5.3	35.5	0.6222	0.4308	0.8749
MR-MP(Q)	22.9	74.7	86.9	75.5	2.2	38.85	54.5	7.5	31.0	0.6086	0.3872	1.0771	

Table 4.20: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-DRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-DRU trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.6434	7.1207	8.6687	1.0284	7.1207	4.0745	0.1339	8.0495	4.0917	0.4415	0.0005	7.4826
	UMR-MP(T)	5.2603	12.3839	37.1517	1.1075	12.3839	6.7457	0.1704	31.8885	16.0295	0.2561	-0.0017	5.9415
	UMR-MP(SRT)	3.5344	11.7647	39.6285	2.1967	5.2632	3.7299	0.6937	13.6223	7.158	0.311	-0.0059	6.3394
	UMR-MP(Q)	2.4767	3.4056	7.7399	2.9451	2.4768	2.7109	0.7971	3.7152	2.2561	0.453	-0.008	9.9115
	MR-MP	9.2224	12.5461	15.8672	4.5152	9.5941	7.0546	1.4303	11.8081	6.6192	0.546	0.0177	5.0717
	MR-MP(T)	3.0605	6.6421	11.0701	100.0	0.0	50.0	100.0	0.0	50.0	0.8828	0.647	7.2443
	MR-MP(SRT)	2.9519	4.059	8.1181	100.0	0.0	50.0	100.0	0.0	50.0	0.9063	0.6824	9.6126
	MR-MP(Q)	2.9519	3.69	5.1661	100.0	0.0	50.0	100.0	0.0	50.0	0.9183	0.5795	11.3275
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	7.737	15.7895	29.7214	0.6876	17.3375	9.0125	0.0791	30.031	15.055	0.3788	-0.0029	4.5255
	UMR-MP(T)	0.7112	0.6192	9.5975	2.2088	0.6192	1.414	0.2921	3.096	1.694	0.4693	0.0014	12.1784
	UMR-MP(SRT)	0.6169	0.6192	2.1672	3.4684	0.6192	2.0438	0.7728	0.6192	0.696	0.5959	0.0026	15.7958
	UMR-MP(Q)	0.4347	0.3096	0.9288	2.6591	0.0	1.3296	0.5416	0.6192	0.5804	0.7126	-0.0066	24.0354
	MR-MP	9.5927	21.4022	31.7343	6.268	11.8081	9.038	1.4934	17.7122	9.6028	0.5005	0.0597	3.7232
	MR-MP(T)	3.1096	7.0111	20.6642	46.9256	0.0	23.4628	28.6966	0.0	14.3483	0.781	0.2289	7.866
	MR-MP(SRT)	2.2113	2.214	8.8561	44.0651	0.0	22.0325	25.7519	1.107	13.4295	0.8202	0.2148	9.5204
	MR-MP(Q)	1.4742	1.845	2.952	22.2323	0.0	11.1162	10.061	0.0	5.0305	0.8775	0.1218	17.1006

Table 4.21: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-ADRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-DRU trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.7192	-0.0004	34.5764
	UMR-MP	0.7	0.7	1.0	0.5	0.8	0.65	0.1	1.1	0.6	0.5247	-0.0003	13.9739
	UMR-MP(T)	38.0	95.9	98.2	0.0	99.9	49.95	0.0	99.9	49.95	0.0282	-0.0001	0.1674
	UMR-MP(SRT)	38.9	95.1	98.3	0.0	99.7	49.85	0.0	100.0	50.0	0.0275	0.001	0.1487
	UMR-MP(Q)	36.0	94.1	98.6	0.2	98.5	49.35	0.1	99.4	49.75	0.0438	0.0013	0.2892
	MR-MP	0.7	0.7	1.0	1.8	0.6	1.2	0.7	0.7	0.7	0.606	0.0089	15.5591
	MR-MP(T)	6.8	13.1	30.2	100.0	0.0	50.0	100.0	0.0	50.0	0.8838	0.7481	4.5142
	MR-MP(SRT)	5.3	12.9	25.7	100.0	0.0	50.0	100.0	0.0	50.0	0.8974	0.7799	4.7076
MR-MP(Q)	6.6	17.3	42.2	100.0	0.0	50.0	100.0	0.0	50.0	0.9209	0.8139	4.3459	
Mobile-Facenet	UMR-UMP	0.7	0.7	1.0	1.0	0.7	0.85	0.1	1.0	0.55	0.6838	0.0062	19.218
	UMR-MP	3.8	6.4	15.1	1.1	6.4	3.75	0.3	11.7	6.0	0.4464	0.0035	6.4567
	UMR-MP(T)	38.9	96.0	99.0	1.0	96.0	48.5	0.0	99.2	49.6	0.0605	0.0078	0.1696
	UMR-MP(SRT)	36.5	93.5	97.9	1.1	93.5	47.3	0.0	98.4	49.2	0.0663	0.0039	0.2217
	UMR-MP(Q)	34.6	92.7	97.8	2.2	89.6	45.9	0.3	95.9	48.1	0.0822	0.0071	0.2813
	MR-MP	3.6	6.6	17.6	7.6	1.8	4.7	3.1	4.7	3.9	0.5595	0.0508	7.3776
	MR-MP(T)	18.6	66.8	82.4	93.9	0.6	47.25	89.7	0.8	45.25	0.7308	0.4857	1.362
	MR-MP(SRT)	18.0	59.2	70.8	90.4	0.1	45.25	84.2	0.6	42.4	0.6917	0.4268	1.6467
MR-MP(Q)	16.3	58.2	74.2	93.1	0.0	46.55	87.0	0.3	43.65	0.727	0.4577	1.8288	

Table 4.22: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-ADRU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-ADRU trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.6434	7.1207	8.6687	1.0284	7.1207	4.0745	0.1339	8.0495	4.0917	0.4415	0.0005	7.4826
	UMR-MP(T)	3.7135	6.192	12.3839	0.6633	8.0495	4.3564	0.0487	14.5511	7.2999	0.3118	-0.0014	7.4124
	UMR-MP(SRT)	4.0235	6.192	10.5263	0.6572	6.5015	3.5794	0.1156	10.2167	5.1662	0.3636	-0.005	8.0399
	UMR-MP(Q)	3.0996	4.3344	7.7399	3.5779	3.096	3.337	1.2352	4.0248	2.63	0.5009	0.0007	10.9618
	MR-MP	9.2224	12.5461	15.8672	4.5152	9.5941	7.0546	1.4303	11.8081	6.6192	0.546	0.0177	5.0717
	MR-MP(T)	2.9519	4.797	11.8081	100.0	0.0	50.0	99.986	0.0	49.993	0.8886	0.5786	9.1291
	MR-MP(SRT)	3.8081	5.9041	7.7491	100.0	0.0	50.0	100.0	0.0	50.0	0.9119	0.6441	9.0699
	MR-MP(Q)	3.3187	4.797	7.3801	100.0	0.0	50.0	99.986	0.0	49.993	0.927	0.537	9.6938
	Mobile-Facenet	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.7512	0.0057
UMR-MP		7.737	15.7895	29.7214	0.6876	17.3375	9.0125	0.0791	30.031	15.055	0.3788	-0.0029	4.5255
UMR-MP(T)		0.6321	0.6192	6.8111	2.0871	0.3096	1.1984	0.3103	2.1672	1.2388	0.4732	-0.0109	12.2255
UMR-MP(SRT)		0.9269	0.9288	2.4768	3.8274	0.0	1.9137	0.8032	0.9288	0.866	0.621	0.004	16.9013
UMR-MP(Q)		0.1491	0.0	0.3096	3.4562	0.0	1.7281	0.858	0.0	0.429	0.7287	0.0024	25.1028
MR-MP		9.5927	21.4022	31.7343	6.268	11.8081	9.038	1.4934	17.7122	9.6028	0.5005	0.0597	3.7232
MR-MP(T)		2.7463	5.9041	19.5572	44.0581	0.0	22.029	28.5564	0.0	14.2782	0.81	0.2125	7.6643
MR-MP(SRT)		0.7546	0.738	6.6421	39.8584	0.0	19.9292	23.1578	0.0	11.5789	0.8606	0.1947	12.2521
MR-MP(Q)		0.3668	0.369	1.845	26.9859	0.0	13.493	13.833	0.0	6.9165	0.9088	0.1462	18.5329

Table 4.23: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-CU trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	2.4	3.7	5.2	0.0	38.3	19.15	0.0	77.4	38.7	0.134	0.0008	8.0196
	UMR-MP(SRT)	3.3	5.3	14.0	0.0	28.2	14.1	0.0	66.0	33.0	0.1448	0.0054	7.3063
	UMR-MP(Q)	2.7	4.3	7.3	0.0	20.8	10.4	0.0	52.4	26.2	0.1595	0.0042	7.8381
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	1.4	1.6	2.8	9.0	0.8	4.9	3.4	0.9	2.15	0.5313	0.0127	11.1802
	MR-MP(SRT)	1.3	1.3	2.9	12.0	0.5	6.25	3.9	0.9	2.4	0.534	0.024	10.9478
MR-MP(Q)	1.0	1.0	1.5	8.1	0.5	4.3	3.0	0.8	1.9	0.5317	0.0152	11.8634	
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	14.5	70.5	92.9	0.0	100.0	50.0	0.0	100.0	50.0	0.238	0.0253	2.0769
	UMR-MP(SRT)	21.9	85.2	95.3	1.4	82.8	42.1	0.0	99.5	49.75	0.4497	0.2981	1.1646
	UMR-MP(Q)	20.2	78.7	91.3	1.6	70.9	36.25	0.0	96.7	48.35	0.4627	0.2784	1.3565
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	11.7	43.4	71.1	17.9	6.0	11.95	13.7	10.0	11.85	0.7817	0.0746	2.8656
	MR-MP(SRT)	12.4	46.4	70.9	55.4	0.4	27.9	48.7	0.9	24.8	0.8843	0.5017	1.7504
MR-MP(Q)	11.7	44.8	60.9	44.1	0.6	22.35	35.3	1.3	18.3	0.8567	0.408	2.0811	

Table 4.24: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CU trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-U MP, UMR-MP, MR-MP.

MFR2/ MFU-CU trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR100_Th^{UMR-UMP}$			$FMR1000_Th^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-U MP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	0.6929	0.3096	1.8576	0.2677	1.2384	0.7531	0.0365	2.7864	1.4114	0.3179	-0.0051	22.0664
	UMR-MP(SRT)	0.6291	0.6192	2.4768	0.5598	0.9288	0.7443	0.1156	2.4768	1.2962	0.3447	0.0119	20.9295
	UMR-MP(Q)	0.3343	0.3096	1.8576	0.8945	0.3096	0.602	0.2434	1.2384	0.7409	0.4285	0.0043	26.3863
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	2.6657	3.69	11.0701	7.9927	0.738	4.3654	3.2111	2.583	2.8971	0.6036	-0.0021	10.9948
	MR-MP(SRT)	2.5886	5.1661	12.1771	12.4518	0.0	6.2259	5.5178	0.738	3.1279	0.6163	0.0226	10.9089
	MR-MP(Q)	1.8445	2.952	9.2251	8.4344	0.369	4.4017	3.6037	0.738	2.1709	0.6494	0.0058	14.4675
Mobile-Facenet	UMR-U MP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	11.9156	70.8978	93.808	0.0	100.0	50.0	0.0	100.0	50.0	0.2975	0.0366	2.4815
	UMR-MP(SRT)	20.8924	83.9009	95.0464	0.3529	90.7121	45.5325	0.0	100.0	50.0	0.5068	0.3198	1.2976
	UMR-MP(Q)	15.8995	72.1362	91.9505	1.3813	65.3251	33.3532	0.0	99.3808	49.6904	0.5402	0.3006	1.7345
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	11.8074	57.9336	87.4539	15.2352	8.4871	11.8612	11.7998	11.8081	11.8039	0.7892	0.0624	2.6824
	MR-MP(SRT)	14.0187	70.4797	89.2989	47.781	0.738	24.2595	41.5411	1.845	21.693	0.8791	0.4771	1.5504
	MR-MP(Q)	11.4371	61.2546	81.9188	37.4255	0.0	18.7128	31.1646	0.0	15.5823	0.8665	0.3806	1.964

Table 4.25: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM1 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-CUAM1 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	2.8	4.3	8.0	0.0	38.8	19.4	0.0	78.2	39.1	0.1322	0.0013	7.6872
	UMR-MP(SRT)	2.6	3.9	10.5	0.0	26.7	13.35	0.0	61.6	30.8	0.1487	0.0055	7.4128
	UMR-MP(Q)	2.4	3.0	8.2	0.0	19.1	9.55	0.0	51.1	25.55	0.1604	0.0038	7.9872
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	1.4	1.5	2.6	9.8	0.6	5.2	3.6	0.8	2.2	0.5327	0.0185	11.1465
	MR-MP(SRT)	1.2	1.2	2.6	10.3	0.8	5.55	4.0	0.8	2.4	0.5351	0.0266	11.2754
MR-MP(Q)	1.3	1.4	2.2	9.6	0.7	5.15	2.8	0.7	1.75	0.5331	0.0176	11.6922	
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	16.5	78.3	95.8	0.0	100.0	50.0	0.0	100.0	50.0	0.231	0.0232	1.7971
	UMR-MP(SRT)	22.9	85.5	94.3	2.0	79.2	40.6	0.0	99.5	49.75	0.4508	0.2963	1.1581
	UMR-MP(Q)	20.9	81.9	89.4	2.5	73.6	38.05	0.0	97.6	48.8	0.4624	0.2906	1.305
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	12.7	57.1	75.4	18.8	7.6	13.2	14.2	11.0	12.6	0.7718	0.065	2.3795
	MR-MP(SRT)	13.0	52.5	67.8	51.8	0.6	26.2	43.2	1.6	22.4	0.8739	0.4791	1.8377
MR-MP(Q)	11.6	47.0	70.2	47.7	0.6	24.15	39.5	1.1	20.3	0.8617	0.4448	2.029	

Table 4.26: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM1 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ MFU-CUAM1 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	0.313	0.3096	1.8576	0.3225	0.3096	0.316	0.0365	3.096	1.5662	0.3187	-0.0046	21.5349
	UMR-MP(SRT)	1.5468	1.548	3.4056	0.6024	1.8576	1.23	0.1278	3.096	1.6119	0.3474	0.0129	18.1883
	UMR-MP(Q)	0.9269	0.9288	1.548	0.791	0.9288	0.8599	0.2191	1.548	0.8835	0.431	0.0036	26.5117
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	2.3375	4.059	13.2841	8.827	0.0	4.4135	3.8	1.107	2.4535	0.604	-0.0008	10.893
	MR-MP(SRT)	3.689	4.428	14.3911	12.5219	2.214	7.368	5.4617	2.952	4.2069	0.6097	0.0275	9.7265
	MR-MP(Q)	2.7603	3.321	7.3801	8.0698	1.107	4.5884	3.3093	2.214	2.7616	0.6482	0.0038	13.5384
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	13.9302	76.161	94.7368	0.0	100.0	50.0	0.0	100.0	50.0	0.2907	0.0332	2.2488
	UMR-MP(SRT)	22.6001	85.7585	94.1176	0.4077	91.6409	46.0243	0.0	100.0	50.0	0.5098	0.3248	1.264
	UMR-MP(Q)	17.3398	73.9938	93.1889	1.1501	73.065	37.1075	0.0	100.0	50.0	0.5348	0.317	1.6751
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	12.5445	67.1587	88.1919	18.1939	8.4871	13.3405	14.3027	10.7011	12.5019	0.8005	0.0569	2.3738
	MR-MP(SRT)	15.2874	64.2066	82.2878	47.2972	0.369	23.8331	41.1554	1.476	21.3157	0.878	0.474	1.5081
	MR-MP(Q)	11.6159	53.5055	80.0738	40.8329	0.738	20.7855	33.7026	0.738	17.2203	0.8709	0.4305	1.9433

Table 4.27: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM2 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-CUAM2 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	1.1	1.1	1.7	0.0	11.1	5.55	0.0	33.5	16.75	0.1832	0.0015	10.9026
	UMR-MP(SRT)	1.2	1.3	2.0	0.0	9.4	4.7	0.0	26.6	13.3	0.1928	0.0048	10.5437
	UMR-MP(Q)	0.9	0.8	1.9	0.0	3.2	1.6	0.0	8.3	4.15	0.2575	0.0047	11.4926
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	0.8	0.8	1.6	4.6	0.5	2.55	0.7	0.8	0.75	0.4969	0.0269	12.3627
	MR-MP(SRT)	1.0	1.0	1.3	7.2	0.4	3.8	1.5	0.9	1.2	0.4987	0.0383	12.003
MR-MP(Q)	0.9	0.8	1.5	6.4	0.6	3.5	0.9	0.9	0.9	0.5044	0.0299	12.116	
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	13.4	44.1	79.0	0.0	100.0	50.0	0.0	100.0	50.0	0.1996	0.065	2.5654
	UMR-MP(SRT)	18.5	64.3	79.3	0.0	87.2	43.6	0.0	99.7	49.85	0.4389	0.311	1.6066
	UMR-MP(Q)	24.5	78.4	86.6	0.0	100.0	50.0	0.0	100.0	50.0	0.3267	0.2618	0.9004
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	14.7	43.1	63.1	3.6	30.9	17.25	0.9	43.7	22.3	0.5734	0.2405	2.374
	MR-MP(SRT)	15.1	42.1	63.0	68.9	0.7	34.8	47.5	2.8	25.15	0.7729	0.56	2.3078
MR-MP(Q)	23.7	68.2	80.9	95.8	0.5	48.15	87.1	2.3	44.7	0.7846	0.6713	0.9068	

Table 4.28: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM2 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ MFU-CUAM2 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR100_Th^{UMR-UMP}$			$FMR1000_Th^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	3.0936	5.2632	8.0495	0.0061	15.4799	7.743	0.0	39.6285	19.8142	0.1908	0.0018	8.4545
	UMR-MP(SRT)	3.4035	5.2632	7.7399	0.0061	12.0743	6.0402	0.0	30.3406	15.1703	0.2049	0.0126	8.2804
	UMR-MP(Q)	3.0936	4.9536	6.5015	0.1582	6.192	3.1751	0.0	13.6223	6.8111	0.2697	0.0104	8.772
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	7.7482	13.6531	21.0332	5.8824	8.8561	7.3692	1.5495	12.1771	6.8633	0.446	0.0368	4.5446
	MR-MP(SRT)	8.115	13.6531	21.0332	9.9418	7.7491	8.8454	2.6362	11.0701	6.8532	0.4542	0.0589	4.5279
MR-MP(Q)	8.115	13.6531	20.6642	8.0698	8.1181	8.094	2.2506	11.0701	6.6603	0.4614	0.0428	4.6767	
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	12.3895	59.1331	86.3777	0.0	100.0	50.0	0.0	100.0	50.0	0.2117	0.0647	2.6706
	UMR-MP(SRT)	20.7403	69.0402	90.7121	0.0	98.7616	49.3808	0.0	100.0	50.0	0.4698	0.3314	1.4061
	UMR-MP(Q)	23.8916	84.8297	94.4272	0.0	100.0	50.0	0.0	100.0	50.0	0.3439	0.2727	0.9847
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	18.4483	61.9926	77.4908	0.8904	63.4686	32.1795	0.1963	74.1697	37.183	0.5055	0.1991	1.7783
	MR-MP(SRT)	20.0627	60.8856	75.2768	40.0477	7.3801	23.7139	23.5364	16.6052	20.0708	0.73	0.5282	1.5528
MR-MP(Q)	29.1483	74.9077	86.3469	80.5371	1.476	41.0065	61.4317	7.7491	34.5904	0.7544	0.6466	0.7663	

Table 4.29: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM3 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at the pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-CUAM3 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	0.6	0.5	1.0	0.1	1.2	0.65	0.0	3.0	1.5	0.3514	0.0063	12.1177
	UMR-MP(SRT)	0.5	0.5	1.1	0.2	0.9	0.55	0.0	2.7	1.35	0.3643	0.0083	12.219
	UMR-MP(Q)	0.5	0.5	0.8	0.2	0.8	0.5	0.0	2.5	1.25	0.3659	0.0087	12.1053
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	0.8	0.8	1.5	3.6	0.6	2.1	0.8	0.8	0.8	0.4966	0.0261	12.6608
	MR-MP(SRT)	0.9	0.9	1.7	4.0	0.7	2.35	1.1	0.9	1.0	0.5016	0.0297	12.7182
MR-MP(Q)	0.9	0.8	1.5	4.7	0.6	2.65	1.1	0.8	0.95	0.5072	0.028	12.8554	
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	10.0	33.0	70.5	0.0	96.0	48.0	0.0	99.8	49.9	0.38	0.1061	3.3078
	UMR-MP(SRT)	11.9	37.5	43.1	1.3	35.9	18.6	0.0	66.7	33.35	0.5304	0.2967	2.9591
	UMR-MP(Q)	11.3	36.9	52.7	1.2	35.2	18.2	0.0	65.8	32.9	0.5314	0.2814	3.022
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	10.3	46.0	69.8	9.2	11.7	10.45	6.3	19.4	12.85	0.679	0.1184	3.1255
	MR-MP(SRT)	10.6	34.2	55.9	37.7	1.2	19.45	23.6	3.8	13.7	0.76	0.4287	2.8144
MR-MP(Q)	9.5	34.7	44.4	29.2	1.5	15.35	17.2	6.4	11.8	0.7556	0.3731	2.9843	

Table 4.30: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM3 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ MFU-CUAM3 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR100_Th^{UMR-UMP}$			$FMR1000_Th^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	2.8505	4.644	7.7399	0.3408	5.8824	3.1116	0.0304	8.6687	4.3496	0.3315	0.0046	9.0209
	UMR-MP(SRT)	3.0936	4.9536	8.0495	0.5111	4.9536	2.7323	0.0913	8.0495	4.0704	0.3539	0.0125	8.8449
	UMR-MP(Q)	2.1667	3.4056	7.4303	0.8215	3.7152	2.2683	0.1825	5.5728	2.8777	0.3924	0.0126	10.6018
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	8.115	13.2841	20.2952	4.1857	8.8561	6.5209	1.1428	12.5461	6.8445	0.446	0.0315	4.8981
	MR-MP(SRT)	8.115	12.5461	21.0332	5.3285	8.4871	6.9078	1.4653	11.4391	6.4522	0.4555	0.0389	4.9431
	MR-MP(Q)	6.3687	11.4391	20.2952	5.2934	7.0111	6.1522	1.6546	9.5941	5.6244	0.477	0.0311	5.7333
	Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264
UMR-MP		16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
UMR-MP(T)		10.2167	37.7709	61.3003	0.0	99.6904	49.8452	0.0	100.0	50.0	0.386	0.0973	3.0374
UMR-MP(SRT)		13.0824	44.8916	59.7523	0.3164	52.6316	26.474	0.0	84.2105	42.1053	0.5491	0.3113	2.4137
UMR-MP(Q)		11.7635	35.9133	57.5851	0.432	42.7245	21.5782	0.0	75.5418	37.7709	0.5669	0.2968	2.7608
MR-MP		20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
MR-MP(T)		12.9148	57.5646	80.4428	3.772	29.8893	16.8306	1.9771	43.1734	22.5753	0.6262	0.0832	2.7515
MR-MP(SRT)		15.4964	53.1365	70.1107	20.7109	11.8081	16.2595	11.2248	21.7712	16.498	0.7138	0.3936	2.0682
MR-MP(Q)		13.7501	50.5535	69.0037	18.1308	9.9631	14.047	9.4861	18.4502	13.9681	0.7212	0.3477	2.3306

Table 4.31: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on LFW dataset with and without MFU-CUAM4 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold, and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

LFW/ MFU-CUAM4 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR_{100_Th}^{UMR-UMP}$			$FMR_{1000_Th}^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.3	0.3	0.3	1.0	0.3	0.65	0.1	0.3	0.2	0.6017	0.0088	25.9332
	UMR-MP	0.5	0.5	0.9	0.5	0.6	0.55	0.0	1.6	0.8	0.4214	0.0076	12.1078
	UMR-MP(T)	1.0	1.0	1.6	0.0	5.8	2.9	0.0	14.9	7.45	0.224	0.0023	11.166
	UMR-MP(SRT)	0.9	0.9	2.0	0.0	5.0	2.5	0.0	11.7	5.85	0.2377	0.005	11.2706
	UMR-MP(Q)	0.9	0.7	1.8	0.0	3.4	1.7	0.0	8.2	4.1	0.2605	0.0063	11.2741
	MR-MP	0.8	0.8	1.4	3.7	0.5	2.1	0.7	0.8	0.75	0.4942	0.0319	12.6383
	MR-MP(T)	1.0	1.0	2.0	5.3	0.5	2.9	1.3	0.9	1.1	0.518	0.0189	12.6003
	MR-MP(SRT)	0.9	0.8	1.6	6.6	0.5	3.55	1.5	0.8	1.15	0.5202	0.025	12.5434
	MR-MP(Q)	0.9	0.8	1.8	6.4	0.6	3.5	1.6	0.7	1.15	0.5256	0.0219	12.7129
Mobile-Facenet	UMR-UMP	6.9	15.9	28.9	1.0	15.9	8.45	0.1	28.9	14.5	0.6303	0.283	5.1135
	UMR-MP	10.7	29.6	43.1	0.0	58.6	29.3	0.0	79.0	39.5	0.4824	0.227	3.5865
	UMR-MP(T)	12.1	50.7	90.6	0.0	100.0	50.0	0.0	100.0	50.0	0.267	0.0345	2.6956
	UMR-MP(SRT)	18.1	71.1	82.3	0.6	75.0	37.8	0.0	98.6	49.3	0.4608	0.2853	1.6378
	UMR-MP(Q)	15.7	62.2	80.4	1.4	59.7	30.55	0.0	91.5	45.75	0.4865	0.2879	1.8757
	MR-MP	11.5	30.5	39.9	3.6	20.8	12.2	0.2	36.4	18.3	0.6032	0.345	3.1764
	MR-MP(T)	10.3	39.6	71.4	11.1	10.0	10.55	7.4	14.9	11.15	0.7174	0.0748	3.391
	MR-MP(SRT)	11.0	45.7	52.3	44.5	0.3	22.4	35.4	1.6	18.5	0.8269	0.4375	2.2746
	MR-MP(Q)	10.3	38.2	55.8	42.1	0.5	21.3	32.0	1.5	16.75	0.8204	0.413	2.3629

Table 4.32: The achieved verification performance of Resnet-101 and MobileFaceNet backbones on MFR2 dataset with and without MFU-CUAM4 trained with MFR2 dataset and Triplet loss (T), Self Restrained Triplet Loss (SRT) and Quadruplet loss (Q). The best evaluation metrics i.e. lowest EER%, lowest average error of FMR100 and FMR1000 at pre-defined threshold and highest FDR are written in bold for each evaluation experiment i.e. UMR-UMP, UMR-MP, MR-MP.

MFR2/ MFU-CUAM4 trained with MFR2	Experiments	EER%	FMR 100%	FMR 1000%	$FMR100_Th^{UMR-UMP}$			$FMR1000_Th^{UMR-UMP}$			G- mean	I-mean	FDR
					FMR%	FNMR%	Avg%	FMR%	FNMR%	Avg%			
Resnet-101	UMR-UMP	0.0	0.0	0.0	0.9983	0.0	0.4992	0.1062	0.0	0.0531	0.6526	0.0134	45.5058
	UMR-MP	4.9534	6.8111	9.5975	0.4503	8.0495	4.2499	0.0487	10.8359	5.4423	0.3454	0.0084	6.595
	UMR-MP(T)	1.8567	2.4768	6.192	0.2373	4.644	2.4406	0.0304	7.7399	3.8852	0.2927	-0.0016	12.5953
	UMR-MP(SRT)	2.1667	3.096	5.2632	0.3834	4.0248	2.2041	0.0608	5.8824	2.9716	0.3107	0.0108	11.9636
	UMR-MP(Q)	1.9176	2.1672	4.644	0.718	2.1672	1.4426	0.1643	4.644	2.4041	0.3765	0.0099	14.5577
	MR-MP	8.8556	14.3911	22.5092	5.6019	9.5941	7.598	1.3672	13.2841	7.3257	0.4364	0.0482	4.39
	MR-MP(T)	6.6408	10.3321	19.9262	6.0156	7.0111	6.5133	2.0473	8.4871	5.2672	0.5016	0.013	6.0362
	MR-MP(SRT)	6.0685	10.3321	19.1882	7.9787	4.797	6.3879	2.7414	8.8561	5.7987	0.511	0.0271	6.0858
MR-MP(Q)	5.9037	8.8561	17.7122	6.8499	5.1661	6.008	2.503	6.6421	4.5725	0.5322	0.0193	6.9103	
Mobile-Facenet	UMR-UMP	2.4214	6.8182	15.9091	0.9983	6.8182	3.9082	0.1062	15.9091	8.0076	0.7081	0.3264	7.1104
	UMR-MP	16.0969	42.7245	62.2291	0.0	91.3313	45.6656	0.0	98.452	49.226	0.4266	0.2096	2.1091
	UMR-MP(T)	8.6668	47.0588	83.9009	0.0	100.0	50.0	0.0	100.0	50.0	0.3164	0.0414	3.1001
	UMR-MP(SRT)	16.4951	64.3963	88.2353	0.1521	86.0681	43.1101	0.0	100.0	50.0	0.5141	0.3096	1.7013
	UMR-MP(Q)	12.6933	49.226	78.9474	0.9188	49.8452	25.382	0.0061	94.7368	47.3715	0.556	0.3079	2.2416
	MR-MP	20.7086	54.6125	67.8967	0.4697	60.8856	30.6777	0.0351	73.0627	36.5489	0.5214	0.3062	1.5377
	MR-MP(T)	11.4371	55.3506	78.9668	5.9525	19.9262	12.9393	3.3163	26.5683	14.9423	0.689	0.0475	3.0987
	MR-MP(SRT)	15.4964	63.4686	78.2288	36.8155	2.583	19.6993	28.3531	5.9041	17.1286	0.8061	0.4235	1.7836
MR-MP(Q)	11.5038	50.9225	74.5387	31.9007	1.107	16.5039	23.4523	2.583	13.0177	0.8092	0.384	2.2461	

4.4 Qualitative Results

Figure 4.1 and Figure 4.2 show the qualitative results of our approach in UMR-MP and MR-MP scenarios respectively. It can be observed that our proposed approach performs fairly well qualitatively as well also.






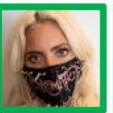


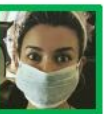






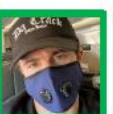


Reference Image	Genuine Pair Match	Imposter Pair Match
	   	   
	   	   
	   	   
	   	   
	   	   
	   	   
	   	   

Figure 4.1: Qualitative Results for UMR-UMP scenario. The green box around the image shows correct prediction whereas the red box around the image show incorrect prediction using our approach.

4.5 Processing Time of MFUM Architectures

We have experimented with different MFUM architectures having different types and numbers of neural network layers, therefore their running time varies. We have run














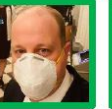














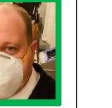









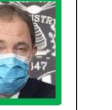





















Reference Image	Geneuine Pair Match					Imposter Pair Match			
									
									
									
									
									
									

Figure 4.2: Qualitative Results for MR-UMP scenario. The green box around the image shows correct prediction whereas the red box around the image show incorrect prediction using our approach.

the inference on Intel Core i7 6500U CPU and reported the running time of these architectures in Table 4.33. For each architecture we have taken 20 readings and taken an average of these readings for reporting the performance. MFU-DRU takes the least time to run followed by MFU-ADRU, and then MFU-CU, MFU-CUAM1, MFU-CUAM2, MFU-CUAM3 and MFU-CUAM4.

Table 4.33: Running time of different MFUM architectures in second for Resnet-101 backbone.

Architecture	Running time (sec)
MFU-DRU	0.0015
MFU-CU	0.006
MFU-CUAM1	0.01
MFU-CUAM2	0.011
MFU-CUAM3	0.012
MFU-CUAM4	0.024
MFU-ADRU	0.0021

4.6 Mask Detector architecture

In the chapter 3, we have mentioned that we have used two approaches for training the Resnet-50 architecture. The second approach in which we transfer learned from ImageNet [12] dataset, performed better than the first approach in which we have not used transfer learning. We have achieved 97.11% accuracy with non transfer learning based approach and 99.2% accuracy with transfer learning based approach. Which means over 2% increase in accuracy with the transfer learning based approach is achieved.

Discussion

In this section, we discuss the results and the intuitions behind the achieved results.

5.1 Degradation in Masked Face Recognition Performance

We have seen in UMR-MP and MR-MP cases in Table 4.7-4.18 that the masks have a negative impact on face recognition models' performance. It is due to critical facial features that play a vital role in decision-making by face recognition models that get hidden under the mask. Masked faces result in corrupted face embeddings, which ultimately leads to an error by the system.

5.2 Larger impact of face mask on FNMR as compared to FMR

We have seen in UMR-MP, MR-MP cases in Table 4.7-4.18 that FNMR is affected more severely as compared to FMR due to the presence of face masks. This is because the face masks hide the important facial features, which ultimately results in difficulty in matching the masked face to the corresponding unmasked face of the same person. When a masked face is compared with an unmasked face of a different person, the similarity is already low, and with masks, the similarity between these images doesn't increase. That's why the chances of a false match remain the same. This is the reason why FMR is not impacted much as compared to FNMR.

5.3 Resnet-101 verification performance better as compared to MobileFacenet

The verification performance of Resnet-101 is better than MobileFacenet. It is due to the fact that Resnet-101 is a relatively very complex model as compared to MobileFacenet and has 66 million trainable parameters as compared to MobileFacenet's 1 million trainable parameters. In addition to this, Resnet-101 produces 512 dimensional embeddings as compared to MobileFacenet's 128 dimensional embeddings and carries more information than MobileFacenet.

5.4 LFW performance better as compared to MFR2

The verification performance of MFUM is better upon the LFW dataset as compared to MFR2. The reason behind this difference is the fact that MFUM is trained using a masked augmented MS1MV2 dataset. Therefore it has better performance on the masked augmented test dataset. This does not mean our model has poor performance on real-world masked face datasets. We still achieve better performance on the real-world masked face dataset using MFUM compared to the scenario when no MFUM is used. This problem can be solved if we have a large real-world masked face recognition dataset for training.

5.5 MFUM architectures comparison

After looking at the results of Table 4.7-4.18, it can be inferred that the addition of MFUM on top of the existing face recognition model has improved the performance of these models. It is due to the fact that MFUM has learned to make the mask face embedding of a person similar to unmasked face embedding of the same person and different from the unmasked face embedding of a different person. Since we have tried different MFUM architectures, i.e., MFU-DRU, MFU-ADRU, MFU-CU, MFU-CUAM, so now let's compare the performance of these architectures on the basis of biometric evaluation metrics (i.e. EER, FMR100%, FMR1000%, $FMR100_Th^{UMR-UMP}$, $FMR1000_Th^{UMR-UMP}$), memory, processing time.

5.5.1 Biometric evaluation metrics

Table 4.7-4.18 shows biometric evaluation metrics results for LFW and MFR2 dataset. First lets consider the EER. MFU-CU and all MFU-CUAM architectures have better performance for UMR-MP case on LFW dataset. But at the same time, MFU-DRU and MFU-ADRU performance is better for MFR2 dataset additionally. Further, MFU-DRU and MFU-ADRU have better performance for MR-MP for both LFW and MFR2 datasets.

Now considering FMR100%, MFU-CU and all MFU-CUAM architectures have better performance for UMR-MP case on LFW dataset. But at the same time, MFU-DRU and MFU-ADRU performance is better for MFR2 dataset as well. In addition to this, MFU-DRU and MFU-ADRU have better performance for MR-MP for both LFW and MFR2 datasets whereas MFU-CUAM3 and MFU-CUAM4 have better performance on MFR2 dataset.

Now considering FMR1000%, MFU-CUAM2, MFU-CUAM3, MFU-CUAM4 architectures have better performance for UMR-MP case on LFW dataset but at the same time MFU-DRU and MFU-ADRU performances are better for MFR2 dataset as well. In addition to this, MFU-DRU and MFU-ADRU have better performance for MR-MP for both LFW and MFR2 whereas MFU-CUAM3 and MFU-CUAM4 has better performance on MFR2 dataset.

5.5.2 Memory Requirement

The MFU-DRU and MFU-ADRU need 512, 128 dimension embedding to be processed (for Resnet-101 and MobileFacenet backbone respectively). whereas MFU-CU and MFU-CUAM require 7x7x512, 7x7x128 dimension embeddings (if Resnet-101 and MobileFacenet respectively). In addition to this, the number of learnable parameters are less for MFU-DRU and MFU-ADRU architectures compared to MFU-CU and MFU-CUAM architectures. Therefore if we compare the memory requirement for each model, we can say that MFU-DRU and MFU-ADRU are the most efficient of the all proposed MFUM architectures. This is not only true for training but also for validation scenario as well.

5.5.3 Processing time

The processing time of each deep learning model is dependent on the number of learnable parameters. A greater number of learnable parameters means a larger number of floating point operations. Therefore, the MFUM architectures having less number of learnable parameters require less amount of time. The MFU-DRU takes the least amount of time to run due to the fact that it has a minimum number of trainable parameters. In addition to this, it takes flattened embeddings and then processes them through lightweight Dense Unit Blocks (DUB). The MFU-ADRU takes a little bit more time than MFU-DRU as an attention mechanism is introduced to each Attention Augmented Dense Unit Block (ADUB) which increases the number of trainable parameters.

Further, MFU-CU takes feature map embedding and then processes them through a series of Conv Unit Blocks (CUB) and has a greater number of trainable parameters. Due to these reasons, MFU-CU is slower than MFU-DRU and MFU-ADRU architectures. Further, different MFU-CUAM architectures have additional CBAM blocks for processing which increases the number of learnable parameters and makes them more slower than MFU-DRU, MFU-ADRU as well as MFU-CU architectures.

5.6 Loss functions comparison

We have seen in Chapter 4 that genuine pair similarity is affected more as compared to imposter pair similarity due to the presence of a mask on the face. This means intra-class distance is increased more as compared to inter-class distance. The main learning objective of MFUM is to minimize this effect. The main function of loss functions used was to train MFUM in such a way that MFUM learns to increase inter-class distance (i.e., increase the distance between embeddings of unmasked and masked images of different persons) and decrease the intra-class distance (i.e., decrease the distance between embeddings of unmasked and masked images of same persons). The loss function, which better trains the MFUM, produces better results upon evaluation. From Chapter 4, We can observe that the naive Triplet loss has proved to be ineffective in training the MFUM. In comparison, the Quadruplet loss is performing better than the Triplet as well as SRT Loss upon the MFR2 dataset. However, Quadruplet and SRT loss have almost identical performances upon the LFW dataset. Both these losses have better performance than Triplet loss. The reason behind the poor performance by Triplet loss is that the Triplet loss requires input triplet, which violates the condition that the anchor-negative

distance should be greater than the anchor-positive distance, which is mathematically written as $d(f(x_i^a), f(x_i^n)) > d(f(x_i^a), f(x_i^p)) + m$. If the condition is not violated, the loss is close to zero, and the model cannot update the distance between the imposter and genuine pairs. Through experimentation, we have seen that the G-mean score is reduced for masked faces, which means unmasked and masked faces have less similarity for genuine pairs. In addition to this, in a real-world training scenario, it is difficult to find the input triplet that violates the condition; therefore, there are very few triplets available for training. Therefore the performance is poor for Triplet loss.

Further, the reason behind the better performance of quadruplet loss is its better generalization capability as compared to triplet loss and self restrained triplet loss. Triplet loss ability is limited to good accuracy on training datasets as its focus is on obtaining correct results on the training dataset. However, triplet loss suffers with generalization capability from the training set to the test set. In contrast, Quadruplet loss generates output with a smaller intra-class variation and higher inter-class variation, which is better than triplet loss. This is why quadruplet loss performs better on the test dataset. On the other hand, Self Restrained Triplet loss performs better than Triplet loss due to its ability to minimize the distance of genuine pairs while maintaining the distance between imposter pairs with minor variation as of the backbone face recognition model.

5.7 MFUM trained on Augmented MS1MV2 dataset vs trained on MFR2 dataset

we have done an ablation study by training the MFUM using MS1MV2 and MFR2 datasets. For the case when MFUM is trained using MS1MV2 datasets, as shown in Table 4.1-4.2 and Table 4.7-4.18, we can observe satisfactory performance. But for the case when MFUM is trained using MFR2 datasets, the results show severe deterioration in performance on the LFW dataset for both backbones and improvement in performance on the MFR2 dataset. This trend can be observed in Tables 4.19-4.32. The performance deteriorates to such a level that the performance on the LFW dataset is even worse than the case when no MFUM is used. Therefore, we can argue that the model overfits the MFR2 dataset and is not able to generalize well upon the LFW dataset mainly due the fact that MFR2 dataset is small in size and deep learning models require a large amount of data to be trained for better generalization. Therefore, this is the reason, we could not train the proposed MFUM on the only available real-world masked face dataset i.e.

MFR2 and we have to use mask augmented dataset. As the large-size mask-augmented dataset is producing better results on the augmented mask datasets as well as small real-world mask face datasets.

5.8 The advantage of MFUM over retraining-based approach

Many different approaches discussed in the chapter 2 have retrained the existing state-of-the-art face recognition models using a combination of masked and unmasked faces. But retraining is a resource and time expensive operation. For example, if we retrain ArcFace Resnet-101 [49], then 66 million parameters would be trained, which is quite expensive. Alternately, if we use our proposed approach, the number of trainable parameters is 1.83 million for MFU-DRU and 2.2 million for MFU-ADRU architectures. A drastic decrease in the number of learnable parameters makes our approach lightweight and efficient. Further, retraining-based approaches have reported a decrease in network performance on unmasked faces. However, this is not the case for our proposed method, as we only process the masked face embedding from MFUM, and unmasked face embedding remains unaffected.

5.9 Mask face detector architecture performance with transfer learning

The mask face detector trained using a transfer learning-based approach performed better than the non-transfer learning-based approach because small-scale features are learned better by a model trained on a bigger dataset, i.e., ImageNet [12] dataset. These small-scale features are common for almost every dataset. Therefore, with the transfer learning-based approach mask face detector achieves better performance than the non-transfer learning-based approach.

Conclusion

In this thesis, we have addressed the challenges faced by face recognition algorithms in masked face recognition scenarios. The challenges include degradation in the performance of existing facial recognition models due to the presence of masks on faces which results in hiding essential facial features required by the face recognition algorithms, resulting in corrupted embeddings which lead to poor performance. In addition, the existing methods proposed for solving this problem are not only computationally expensive but also have downgraded performance in unmasked face recognition scenarios. Therefore, to solve this problem, we have proposed the Mask Face Unveiling Model (MFUM), having Attention Augmented Dense Residual Unit (MFU-ADRU) architecture trained with quadruplet loss, which works on top of the existing face recognition model. The MFUM works upon the facial embeddings generated by the existing face recognition backbone and does not require retraining of the existing facial recognition models. The MFUM tries to unveil the mask faces by making masked facial embedding of a person similar to the facial embedding of the unmasked face of the same person and different from the unmasked facial embedding of a different person. For experimentation, we have taken two pretrained face recognition model backbones, i.e., Resnet-101 and MobileFacenet. In addition, We have also conducted different ablation studies upon the architecture of MFUM, i.e., Masked Face Unveiling Dense Residual Unit (MFU-DRU), Masked Face Unveiling Attention Augmented Dense Residual Unit (MFU-ADRU), Masked Face Unveiling Conv Unit (MFU-CU) and different variations of Masked Face Unveiling Conv Unit with attention mechanism (MFU-CUAM).

Additionally, we have done an ablation study using different loss functions, i.e., Triplet, Self Restrained Triplet, and Quadruplet loss. Further, we have trained MFUM upon the MS1MV2 dataset with augmented mask and tested upon mask augmented LFW

dataset and a real-world mask face recognition dataset, i.e., MFR2, for the evaluation of the proposed MFUM model. We have evaluated our proposed MFUM in nine different experimental settings, for example, Unmasked Reference and Masked Probe (UMR-MP) and Masked Reference and Masked Probe (MR-MP) scenarios. Upon evaluation of test datasets, we have found that the MFUM with MFU-ADRU architecture and trained with quadruplet loss is not only improving the performance of the existing face recognition model by unveiling the masked facial embeddings but also is performing best among all evaluated MFUM architectures. All the experiments are evaluated using biometrics verification metrics used globally for biometric system evaluation, e.g., EER, FMR, FNMR, etc. All the proposed MFUM architectures are more lightweight and efficient than conducting retraining of facial recognition models. In addition, the proposed approach does not impact existing models' performance in unmasked face recognition scenarios. Further, we have also developed a verification and classification portal for testing the complete pipeline in verification and classification scenarios.

6.1 Future Work

The future work can be extended but is broader than creating a large real-world masked face dataset. As we have discussed earlier, there is no large real-world masked face recognition dataset, and data is the most important thing in training a deep learning model. Therefore creating a large real masked face recognition dataset is important for further research in this field. Further, face mask augmentation techniques need improvements to make them produce results that are more similar to the real masked face image. Current masked face augmentation techniques augment masks in 2D and do not keep facial shape/structure under consideration. As in the real world, an image of a person wearing a mask shows facial curves, and the 3D shape of the person is preserved, which can help facial recognition algorithms. In addition to this, the impact of ethnicity, skin color, pose, and lighting on masked face recognition also requires investigation. Further, regenerative approaches like Generative Adversarial Networks and Diffusion Probabilistic Model also require investigation in this masked face recognition domain. This is because the presence of a mask destroys the facial features, and GANs based approaches can help in the regeneration of facial features essential for facial recognition to work, especially in scenarios where the entity of the probe image is already present in the gallery.

Bibliography

- [1] Matthew Turk and Alex Pentland. “Eigenfaces for recognition”. In: *Journal of cognitive neuroscience* 3.1 (1991), pp. 71–86.
- [2] P.J. Phillips et al. “The FERET evaluation methodology for face-recognition algorithms”. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997, pp. 137–143. DOI: [10.1109/CVPR.1997.609311](https://doi.org/10.1109/CVPR.1997.609311).
- [3] James W Tanaka and Joseph A Sengco. “Features and their configuration in face recognition”. In: *Memory & cognition* 25.5 (1997), pp. 583–592.
- [4] Aleix Martinez and Robert Benavente. “The ar face database: Cvc technical report, 24”. In: (1998).
- [5] Guodong Guo, Stan Z Li, and Kapluk Chan. “Face recognition by support vector machines”. In: *Proceedings fourth IEEE international conference on automatic face and gesture recognition (cat. no. PR00580)*. IEEE. 2000, pp. 196–201.
- [6] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. “From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose”. In: *IEEE Trans. Pattern Anal. Mach. Intelligence* 23.6 (2001), pp. 643–660.
- [7] Javid Sadr, Izzat Jarudi, and Pawan Sinha. “The role of eyebrows in face recognition”. In: *Perception* 32.3 (2003), pp. 285–293.
- [8] Xiaofei He et al. “Face recognition using laplacianfaces”. In: *IEEE transactions on pattern analysis and machine intelligence* 27.3 (2005), pp. 328–340.
- [9] K.C. Lee, J. Ho, and D. Kriegman. “Acquiring Linear Subspaces for Face Recognition under Variable Lighting”. In: *IEEE Trans. Pattern Anal. Mach. Intelligence* 27.5 (2005), pp. 684–698.

- [10] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. “Face description with local binary patterns: Application to face recognition”. In: *IEEE transactions on pattern analysis and machine intelligence* 28.12 (2006), pp. 2037–2041.
- [11] Gary B Huang et al. “Labeled faces in the wild: A database for studying face recognition in unconstrained environments”. In: *Workshop on faces in ‘Real-Life’ Images: detection, alignment, and recognition*. 2008.
- [12] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [13] Lior Wolf, Tal Hassner, and Itay Maoz. “Face recognition in unconstrained videos with matched background similarity”. In: *CVPR 2011*. IEEE. 2011, pp. 529–534.
- [14] Hongwei Ng and Stefan Winkler. “A data-driven approach to cleaning large face datasets”. In: *2014 IEEE International Conference on Image Processing (ICIP)* (2014), pp. 343–347.
- [15] Dong Yi et al. “Learning face representation from scratch”. In: *arXiv preprint arXiv:1411.7923* (2014).
- [16] Brendan F Klare et al. “Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1931–1939.
- [17] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. “Deep face recognition”. In: (2015).
- [18] Florian Schroff, Dmitry Kalenichenko, and James Philbin. “Facenet: A unified embedding for face recognition and clustering”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.
- [19] Yandong Guo et al. “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition”. In: *European conference on computer vision*. Springer. 2016, pp. 87–102.
- [20] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.
- [21] Ira Kemelmacher-Shlizerman et al. “The megaface benchmark: 1 million faces for recognition at scale”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 4873–4882.

- [22] Soumyadip Sengupta et al. “Frontal to profile face verification in the wild”. In: *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE. 2016, pp. 1–9.
- [23] Kaipeng Zhang et al. “Joint face detection and alignment using multitask cascaded convolutional networks”. In: *IEEE signal processing letters* 23.10 (2016), pp. 1499–1503.
- [24] Stylianos Moschoglou et al. “Agedb: the first manually collected, in-the-wild age database”. In: *proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 51–59.
- [25] Kaspars Sudars. “Face recognition Face2vec based on deep learning: Small database case”. In: *Automatic Control and Computer Sciences* 51.1 (2017), pp. 50–54. ISSN: 1558-108X.
- [26] Cameron Whitelam et al. “Iarpa janus benchmark-b face dataset”. In: *proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017, pp. 90–98.
- [27] Tianyue Zheng, Weihong Deng, and Jiani Hu. “Cross-age lfw: A database for studying cross-age face recognition in unconstrained environments”. In: *arXiv preprint arXiv:1708.08197* (2017).
- [28] Qiong Cao et al. “Vggface2: A dataset for recognising faces across pose and age”. In: *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE. 2018, pp. 67–74.
- [29] Sheng Chen et al. “Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices”. In: *Chinese Conference on Biometric Recognition*. Springer. 2018, pp. 428–438.
- [30] Ziwei Liu et al. “Large-scale celebfaces attributes (celeba) dataset”. In: *Retrieved August 15, 2018* (2018), p. 11.
- [31] Brianna Maze et al. “Iarpa janus benchmark-c: Face dataset and protocol”. In: *2018 international conference on biometrics (ICB)*. IEEE. 2018, pp. 158–165.
- [32] Sanghyun Woo et al. “Cbam: Convolutional block attention module”. In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 3–19.
- [33] Tianyue Zheng and Weihong Deng. “Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments”. In: *Beijing University of Posts and Telecommunications, Tech. Rep* 5 (2018), p. 7.

- [34] Aqeel Anwar and Arijit Raychowdhury. “Masked face recognition for secure authentication”. In: *arXiv preprint arXiv:2008.11104* (2020).
- [35] Nizam Ud Din et al. “A novel GAN-based network for unmasking of masked face”. In: *IEEE Access* 8 (2020), pp. 44276–44287.
- [36] Mei L Ngan, Patrick J Grother, and Kayee K Hanaoka. “Ongoing face recognition vendor test (FRVT) Part 6A: Face recognition accuracy with masks using pre-COVID-19 algorithms”. In: (2020).
- [37] Zhongyuan Wang et al. “Masked face recognition dataset and application”. In: *arXiv preprint arXiv:2003.09093* (2020).
- [38] Fadi Boutros et al. “MFR 2021: Masked face recognition competition”. In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*. IEEE. 2021, pp. 1–10.
- [39] Mustafa Ekrem Erakın, Uğur Demir, and Hazım Kemal Ekenel. “On Recognizing Occluded Faces in the Wild”. In: *2021 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE. 2021, pp. 1–5.
- [40] Marco Huber et al. “Mask-invariant face recognition through template-level knowledge distillation”. In: *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*. IEEE. 2021, pp. 1–8.
- [41] Yande Li et al. “Cropping and attention based approach for masked face recognition”. In: *Applied Intelligence* 51.5 (2021), pp. 3012–3025. ISSN: 1573-7497.
- [42] David Montero et al. “Boosting Masked Face Recognition with Multi-Task Arc-Face”. In: *arXiv preprint arXiv:2104.09874* (2021).
- [43] Yaoyao Zhong and Weihong Deng. “Face transformer for recognition”. In: *arXiv preprint arXiv:2103.14803* (2021).
- [44] Fadi Boutros et al. “Self-restrained triplet loss for accurate masked face recognition”. In: *Pattern Recognition* 124 (2022), p. 108473. ISSN: 0031-3203.
- [45] Walid Hariri. “Efficient masked face recognition method during the covid-19 pandemic”. In: *Signal, image and video processing* 16.3 (2022), pp. 605–612.
- [46] Abdul Jabbar et al. “AFD-StackGAN: Automatic Mask Generation Network for Face De-Occlusion Using StackGAN”. In: *Sensors* 22.5 (2022). ISSN: 1424-8220. DOI: [10.3390/s22051747](https://doi.org/10.3390/s22051747). URL: <https://www.mdpi.com/1424-8220/22/5/1747>.

BIBLIOGRAPHY

- [47] Omar Adel Muhi, Mariem Farhat, and Mondher Frikha. “Transfer Learning for Robust Masked Face Recognition”. In: *2022 6th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. 2022, pp. 1–5. DOI: [10.1109/ATSIP55956.2022.9805960](https://doi.org/10.1109/ATSIP55956.2022.9805960).
- [48] Naser Damer et al. “The effect of wearing a mask on face recognition performance: an exploratory study”. In: *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*. IEEE, pp. 1–6. ISBN: 3885797003.
- [49] Jiankang Deng et al. “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.
- [50] Devira Anggi Maharani et al. “Improving the Capability of Real-Time Face Masked Recognition using Cosine Distance”. In: *2020 6th International Conference on Interactive Digital Media (ICIDM)*. IEEE, pp. 1–6. ISBN: 1728149282.
- [51] Susanta Malakar et al. “Masked Face Recognition Using Principal component analysis and Deep learning”. In: *2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE, pp. 785–788. ISBN: 1665403829.
- [52] Imran Qayyum Mundial et al. “Towards facial recognition problem in COVID-19 pandemic”. In: *2020 4rd International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM)*. IEEE, pp. 210–214. ISBN: 1728188709.
- [53] Sourabh Sarkar and Geeta Sikka. “A comparative study of classifiers used in facial embedding classification”. In: *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, pp. 46–50. ISBN: 1538663732.
- [54] Vibhaakar Sharma, Swathi Gangaraju, and Vishal K Sharma. “Masked Face Recognition”. In: ().
- [55] Lingxue Song et al. “Occlusion robust face recognition based on mask learning with pairwise differential siamese network”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 773–782.
- [56] Yaniv Taigman et al. “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.

BIBLIOGRAPHY

- [57] Hao Wang et al. “Cosface: Large margin cosine loss for deep face recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5265–5274.
- [58] Zheng Zhu et al. “WebFace260M: A Benchmark Unveiling the Power of Million-Scale Deep Face Recognition”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10492–10502.

Publications arose from the thesis

1. M.A. Nawshad, Z. Zafar and M.M. Fraz, "Recognition of faces wearing masks using skip connection based dense units augmented with Self Restrained Triplet Loss", in the proceedings of 24th IEEE International Multi Topic Conference 2022 (INMIC 2022) 21-22 Oct 2022, Islamabad, Pakistan.
2. M.A. Nawshad and M.M. Fraz, "Improving Masked Face Recognition using Dense Residual Unit Aided with Quadruplet loss", in the proceedings of 37th International Conference on Image and Vision Computing New Zealand (IVCNZ 2022) 24-25 Nov 2022, Auckland, New Zealand.
3. M.A. Nawshad, A. Sadaat and M.M. Fraz, "Boosting Facial Recognition Capability for Faces Wearing Masks using Attention Augmented Residual Model with Quadruplet loss" in journal (Under review in Neurocomputing)