석사 학위논문

Question Generation for Educational Assessment

2023 년 8월 22일

전 북 대 학 교 대 학 원

전자공학부(컴퓨터공학)

신달리아

Question Generation for Educational Assessment

지 도 교 수 이 경 순

이 논문을 공학학 석사 학위논문으로 제출함

2023 년 4월 1일

전 북 대 학 교 대 학 원

전자공학부(컴퓨터공학)

신달리아

신달리아의 석사학위논문을 인준함

위원장 전북대학교 교수 오일석 (인)

부위원장 한국전자통신연구원 책임연구원 황금하(인)

위원장 전북대학교 교수 이경순 (인)

2023 년 4월 1일

전 북 대 학 교 대 학 원

**Table of Contents**

[Table of Contents i](#_Toc100363395)

[List of Figures iii](#_Toc100363396)

[List of Tables iv](#_Toc100363397)

[List of Acronyms v](#_Toc100363398)

[Abstract - 1 -](#_Toc100363399)

[Chapter 1. Introduction - 3 -](#_Toc100363400)

[1.1 Overview - 3 -](#_Toc100363401)

[1.2 Challenges of MFR - 5 -](#_Toc100363402)

[1.3 Contributions - 5 -](#_Toc100363403)

[Chapter 2. Related Works - 7 -](#_Toc100363404)

[2.1 Face Recognition - 7 -](#_Toc100363405)

[2.2 Attention Mechanism - 11 -](#_Toc100363406)

[2.3 Masked Dataset - 12 -](#_Toc100363407)

[Chapter 3. Methodology - 16 -](#_Toc100363408)

[3.1 Convolutional Neural Network (CNN) - 16 -](#_Toc100363409)

[3.2 Feature Extraction Network - 19 -](#_Toc100363410)

[3.3 Attention Modules - 21 -](#_Toc100363411)

[3.3.1 Channel Attention Module - 21 -](#_Toc100363412)

[3.3.2 Spatial Attention Module - 22 -](#_Toc100363413)

[3.4 Network Architecture - 23 -](#_Toc100363414)

[3.5 Loss Function - 24 -](#_Toc100363415)

[Chapter 4. Experiments and Results - 27 -](#_Toc100363416)

[4.1 Datasets - 27 -](#_Toc100363417)

[4.1.1 Proposed Masked Face Dataset - 27 -](#_Toc100363418)

[4.1.2 Dataset Generation - 27 -](#_Toc100363419)

[4.1.3 Training Dataset - 29 -](#_Toc100363420)

[4.1.4 Validation Dataset - 30 -](#_Toc100363421)

[4.2 Experimentation Details - 33 -](#_Toc100363422)

[4.3 Evaluation Metrics - 33 -](#_Toc100363423)

[4.3.1 Accuracy - 33 -](#_Toc100363424)

[4.3.2 Precision - 34 -](#_Toc100363425)

[4.3.3 Recall - 34 -](#_Toc100363426)

[4.3.3 F1 score - 35 -](#_Toc100363427)

[4.4 Evaluation Results - 36 -](#_Toc100363428)

[4.4.1 Comparison with FR methods - 38 -](#_Toc100363429)

[4.4.2 Comparison with MFR methods - 39 -](#_Toc100363430)

[4.4.3 Ablation Experiment - 41 -](#_Toc100363431)

[Chapter 5. Conclusion and Future Works - 42 -](#_Toc100363432)

[Acknowledgment - 43 -](#_Toc100363433)

[References - 45 -](#_Toc100363434)

**List of Figures**

[Figure 2.1 Samples of the masked face datasets. (a) The SMFRD dataset; (b) The VGG-Face2\_m dataset. - 13 -](#_Toc100363238)

[Figure 3.1 The basic CNN architecture for image classification. - 16 -](#_Toc100363244)

[Figure 3.2 The residual block of ResNet. - 18 -](#_Toc100363245)

[Figure 3.3 Structure of the improved residual unit: BN-Conv(stride=1)-BN-PReLu-Conv(stride=2)-BN. - 20 -](#_Toc100363246)

[Figure 3.4 The overall structure of the CBAM. - 21 -](#_Toc100363247)

[Figure 3.5 The structure of the channel attention module. - 22 -](#_Toc100363248)

[Figure 3.6 The structure of the spatial attention module. - 23 -](#_Toc100363249)

[Figure 3.7 The overall structure of the proposed network architecture. The convolutional block attention module (CBAM) is integrated into each output of the block. - 24 -](#_Toc100363250)

[Figure 4.1 The standard mask templates. - 28 -](#_Toc100363251)

[Figure 4.2 The generation diagram of the masked face dataset. - 29 -](#_Toc100363252)

[Figure 4.3 Sample images from CASIA-WebFace\_m dataset. - 30 -](#_Toc100363253)

[Figure 4.4 Sample images of validation datasets LFW\_m, AgeDB-30\_m, and CFP\_FP\_m. Each dataset has two columns of corresponding images. - 32 -](#_Toc100363254)

[Figure 4.5 Sample images of real masked face validation datasets MFR2. Each identity has two columns of images. - 32 -](#_Toc100363255)

[Figure 4.6 The accuracy curve of the training model - 37 -](#_Toc100363256)

[Figure 4.7 The loss curve of the training model. - 37 -](#_Toc100363257)

**List of Tables**

[Table 2. 1 A summary of occlusion and MFR approaches. - 14 -](#_Toc100363265)

[Table 4.1 Summary of datasets used for model training and evaluation. - 31 -](#_Toc100363271)

[Table 4.2 Verification accuracy results (%) of 10-fold cross-validation on the validation dataset. - 36 -](#_Toc100363272)

[Table 4.3 Results of accuracy, precision, recall, and f1 score (%). - 38 -](#_Toc100363273)

[Table 4.4 Comparison of face verification results (%) on validation dataset with different methods. - 38 -](#_Toc100363274)

[Table 4.5 Comparison of face verification (%) with different methods. - 39 -](#_Toc100363275)

[Table 4.6 Verification accuracy results (%) of 10-fold cross-validation on the validation dataset. - 40 -](#_Toc100363276)

[Table 4.7 Results of accuracy, precision, recall, and f1 score (%) with the VGG-Face2\_m dataset. - 40 -](#_Toc100363277)

[Table 4.8 Comparison of face verification (%) with different methods. - 40 -](#_Toc100363278)

[Table 4.9 Ablation experiment results of the attention module. - 41 -](#_Toc100363279)

ABSTRACT

Question Generation for Educational Assessment

Dalya Sin

Division of Computer Science and Engineering

Graduate School

Jeonbuk National University

Masked Face Recognition (MFR), which refers to recognizing an individual identity when wearing a facial mask, is the greatest challenging research topic in computer vision. At the same time, the research on facial recognition has reached the peak of progress and introduced many robust methods. However, since the outbreak of COVID-19 has spread worldwide, people have to wear facial masks regularly, making existing face recognition methods less reliable. To overcome the difficulties, we introduce a novel deep learning method based on the convolutional block attention module (CBAM) and angular margin Arcface loss to solve the problem of MFR. CBAM is integrated with convolutional neural networks (CNNs) to extract the input image feature maps, particularly the region around the eyes. ArcFace is used as a training loss function to optimize the feature embedding and enhance the discriminative feature for MFR. Because of the insufficient masked face images for the training model, this study uses the data augmentation method to generate masked face images from the common face recognition dataset. We evaluated the proposed method using the well-known masked image version of LFW, AgeDB-30, CFP-FP, and real mask image MFR2 verification datasets. A variety of experiments were carried out, which confirmed that the proposed method offers improvements to MFR compared with the current state-of-the-art methods.

**Keywords:** face recognition, masked face recognition, deep learning, attention mechanism, margin loss, COVID-19.

The author of this thesis is a Global Korea Scholarship(GKS) scholar sponsored by the Korean Government.

# Introduction

## 1.1 Overview

Face Recognition (FR) has been one of the most important research topics for many years. The trend of developing methods for FR has almost reached its peak. With the influence of the deep convolutional neural networks (DCNNs), the current existing algorithms using deep learning methods [1-6] have achieved superior accuracy for FR. The systems based on FR are widely used in many areas worldwide, including airports, community gates, and trains. It is also employed in some authentication applications, such as face-to-face attendance monitoring and mobile payments based on face profiles.

Since the emergence of COVID-19 pneumonia, a viral infection caused by server acute respiratory syndrome [7], has spread globally, many major challenges have arisen in daily human lives. COVID-19 disease is transmitted from person to person through close contact, as presented in [8]. People need to follow the rules by keeping a social distance, washing their hands often, and using a protective mask face in public regularly to protect themselves from virus infection. However, following the rule by wearing a face mask will impact current FR application systems based on human facial recognition. In the real-world FR application system, face occlusion, particularly masked face occlusion, will significantly affect existing FR performance and decrease re-identification accuracy [9].

Furthermore, many businesses have already implemented the required datasets for facial recognition as a person verification or identification method. Facial verification, often known as one-to-one (1:1) matching, is a method of verifying if someone is who they claim to be. In order to execute secure verification, a personal facial image is captured, from which a biometric template is constructed and matched to an existing facial signature. In contrast, facial identification, often known as one-to-man (1:N) matching, is a biometric recognition that compares and analyzes an individual pattern against a vast dataset of known faces.

Modern deep learning-based models using DCNN have enough abilities to extract the face image features and learn important key features such as face edges, mouth, nose, and eyes [10]. However, with the presence of face masks, most of the key features are occluded, making the features extraction process more complicated. Since traditional methods for FR have been designed specifically to work with the entire face information, wearing a face mask makes the models lose about 50% of the useful information [11]. The face mask blocks important features such as the mouth and nose, making the face feature structure an obstruction, as stated in [12]. This particular issue has lately emerged as a serious barrier to the field of FR. Therefore, novel methods must be invented, or modification of the existing algorithms must be determined in this context.

There are two closely different research domains. One is face mask recognition (FMR), and another is MFR. FMR is used to recognize faces wearing a mask. Many robust solutions have been introduced [13-15] to solve the FMR, while MFR still needs further exploration. Recently, many scholars have presented their methods to address the MFR problems using deep learning techniques [10, 12, 16-21]. In addition, because of insufficient masked face images, several masked face datasets [12, 22, 23] and data augmentation tools [10, 12] for generating simulated masked images have been proposed. A review paper summarized the currently proposed method for MFR is studied in [24].

Hence, this thesis proposes a method to solve the problem of MFR by verifying individuals with the masked face using an attention module and angular margin loss ArcFace. This thesis uses ResNet-50 [25] network with some modifications based on work implemented by Deng et al. [5] integrated with the CBAM attention module presented by Woo et al. [26].

## 1.2 Challenges of MFR

Existing works on MFR have certain limitations, such as the lack of robust methods and datasets to tackle this problem. Deep learning methods require a large number of training datasets, and it is an interesting challenge to find the best solution to handle MFR tasks for several reasons.

* Key features such as mouth, nose, and chin are occluded. It means effective and discriminative features are significantly reduced
* There is no large-scale masked face training and testing dataset available publicly
* The lack of a large-scale masked face dataset causes the model not to be able to learn all of the significant feature maps
* Collecting and annotating millions of masked face datasets is laborious and time-consuming.

## 1.3 Contributions

This thesis introduces an approach based on the attention mechanism integrated with the refined ResNet-50 network architecture to contribute to this task. The following are the primary contributions of this thesis:

1. Generate simulated masked face images from existing regular face recognition datasets for the training dataset and verification dataset using data augmentation
2. Propose a new novel masked face recognition using deep learning network architecture based on attention module and ArcFace angular loss
3. Integrate attention module with refined ResNet-50 as a backbone for feature extraction network without adding any additional computational cost
4. Furthermore, we evaluate the performance of the proposed method on our generated verification, and the real masked face image dataset proved that our method achieves high accuracy.

# Related Works

## 2.1 Face Recognition

FR is one of the most significant tasks in computer vision, and it has received much attention from scholars. Many researchers have introduced robust methods [1-6] to solve the FR problem. Therefore, many recognition systems have used facial recognition techniques for security check purposes. However, the existence of face disturbances such as occlusions and variations in lighting and facial emotions has a detrimental impact on the performance of FR algorithms. Traditional FR approaches are confused with intricate and occluded faces for MFR, increasing the necessity for altering them to learn masked face representations.

The National Institute of Standards and Technology (NIST) [27] demonstrated the performance of a collection of facial recognition algorithms built and turned after the COVID-19 epidemic. They observed that when the facial mask covers faces, recognition performance deteriorates, and most recognition algorithms assessed after the epidemic still perform poorly. After the success of FR research, the researchers have continued to focus on occluded face recognition challenges [18, 28, 29]. Occluded face recognition is a general facial occlusion challenge since the human face can be obscured by something anywhere, in any size or shape [30]. After the COVID-19 pandemic, MFR becomes one of the greatest challenges in the FR domain. MFR is a specific facial occlusion challenge since the essential part of the face (mouth, nose, chin) is occluded. The objective of the study on MFR is to identify or verify a specific identity when wearing a facial mask. It has attracted many scholars who have intended to challenge this problem by introducing different methods. Below are some existing methods proposed by many scholars to solve occluded face recognition and MFR.

Song et al. [18] presented a new technique to address partial occlusion by discovering and disposing corrupted feature elements from recognition. This study decomposed the face recognition challenge under random partial occlusions into three stages. First, pair-wise differential siamese network (PDSN) as a learning mask generator to capture the correspondence between the occluded facial block and corrupted feature elements. Second, build a masked dictionary from the learned mask generator in the previous stage to composite the Feature Discarding Mask (FDM). Third, a combination of the FDM of random partial occlusions from the dictionary is multiplied by the original feature to eliminate the effect of partial occlusions from recognition.

Various research adopted restoration-based methods to restore the missing part of the face image and reconstruct the new face image from the training dataset. Sparse representation-based classification (SRC) [31] on robust occlusion face recognition is a pioneering work in image construction. Later, various extended versions of SRC were introduced for specific challenges in face recognition, such as the extended SRC (ESRC) [32] and sparse group coding (GSC) [33] for under-sample face recognition tasks, and increasing the power of discriminator for image reconstruction, respectively. Many other traditional methods focused on reconstructing the missing parts of the image. For occlusion face recognition, Yuan et al. [34] applied a support vector discrimination dictionary and Gabor occlusion dictionary based-based SRC (SVGSRC). A robust occlusion face recognition classification scheme based on depth dictionary representation was also discussed in [35]. These image reconstruction methods have done great work. However, these methods necessitate an extensive dictionary when the gallery images of each subject are frequently insufficient in practical scenarios. Increasing large gallery images leads to a complex issue and limitation in the generalization.

Since the introduction of Generative Adversarial Nets (GANs) [36], many researchers have abandoned traditional methods and replaced them with GAN methods. Yeh et al. [37] used a semantic image inpainting-based to generate the corrupted pixel and region. Li et al. [38] presented a model to learn the global and local structure of the image and complete the missing region by reconstructing the corrupted part. Din et al. [19] also proposed a model that can remove the mask part and complete the mask occlusion part of the facial images. First, their model detects the mask region and produces it as binary segmentation. Then, they use two discriminators based on the GAN network to learn the global structure and missing part of the facial image. However, these approaches do not evaluate the recognition performance with their model. Unlike the previous GAN-based method, Li et al. [39] also presented an algorithm framework that consists of de-occlusion and distillation modules. The de-occlusion module uses GAN to perform masked face completion, which recovers the occluded features under the mask and eliminates the appearance uncertainty. The distillation module uses pre-trained model to perform face classification. On the simulated LFW dataset, their highest accuracy is 95.44% for recognition performance.

MFR became the necessary topic for research during the COVID-19 epidemic. Mandal et al. [16] proposed a new framework to handle the FMR problem using deep network-based ResNet-50 [25]. The authors trained the network using the small Real-world Masked Face Recognition Dataset (RMFRD) offered by [22]. However, this method did not yield a decent result because the used network only works with the non-occlusion face. Anwar and Raychowdhury [10] presented a similar strategy by using a deep network-based face recognition system, FaceNet [1], to train with their dataset VGGFace2-mini-SM1. They used their own proposed simulated masked face dataset to train the network. This method produced better results than the first method since they trained with a large dataset from scratch.

Huang et al. [40] used deep network-based face recognition systems, ArcFace [5], to train with their simulated dataset WebFace-OCC. Their simulated dataset is generated with random occlusion (mask, glasses); the network can learn more features than the masked dataset. However, their performance results greatly decreased when tested with only the masked face dataset. Walid Hariri [17] proposed a new method to discard the occlusion region. Their method is deep learning-based using the pre-train network to handle the MFR problem. They applied the cropping filter technique to remove the occlusion part covered with a face mask and extract only the non-masked face region. Firstly, they normalized all input images into pixels. After then, they divided the images partition into 100 fixed-size square blocks. It means each block is pixels size. Finally, they successfully extract only the non-masked region by considering the upper half part of the block from 1 to 50. This technique can discard non-masked face areas from each image well. However, it cannot guarantee a clean elimination of non-masked face parts since all face masks do not stay in the same position

Recent work attempts to deal with MFR using attention mechanisms. Li et al. [20] proposed a new strategy by integrating a cropping-based and attention-based approach with the CBAM [26]. The cropping-based process removes the masked face region from face images. They conducted several cropping proportion cases of the input image to find the best one to achieve the best recognition accuracy. In the attention-based process, the masked face features and features around the eyes are given a low and higher weight, respectively. The author reported that their approach achieves 92.61% MFR accuracy. Hongxia Deng et al. [12] proposed an algorithm using cosine loss (MFCosfacce) to address the MFR. As a result, their method improved the accuracy of masked face recognition compared to the first method based on attention. They also designed an Att-Inception module that combined the convolutional block attention module with Inception-ResNet to help the model pay greater attention to the region not covered by the mask. This technique provided slightly improve in verification task on generation dataset.

Current works inspire our work. By observing the strength of the attention module that plays an important role in MFR work, we extend them further by proposing a novel network architecture by integrating the attention module with the refined ResNet-50 network provided in the ArcFace repository. We also use a data augmentation tool to generate the masked face images version for model training and evaluation.

## 2.2 Attention Mechanism

In general, attention is the cognitive activity of selectively focusing on something and taking more important notice while disregarding others. For example, there are one apple and an orange on the table. If we look at the apple, our attention allows us to focus on the apple with high resolution while perceiving the orange and the surrounding in low resolution. It means human visual attention allows us to focus on an object of interest effectively. In deep learning, the attention mechanism was first introduced in Natural Language Processing (NLP) in 2017 by Google Brain [41]. Later, it works as an essential role in many visual tasks, e.g., image classification [26, 42], semantic segmentation [43, 44], object detection [45, 46], person re-identification [47, 48], action recognition [49, 50], face recognition [51, 52]. In [26], CBAM is a lightweight attention module that consists of two sub-modules. These modules can pay more attention to the meaningful and informative part of the input image. Motivated by this observation, this paper adopts CBAM in the proposed network to figure out MFR.

## 2.3 Masked Dataset

Many face recognition datasets have been proposed and provided publicly in the past years. However, it is not sufficient for the masked face dataset to train the deep learning model. Several recent works have contributed to tackling the insufficient masked face dataset. Wang et al. [22] introduced three types of masked faces: Real-world Masked Face Recognition Dataset (RMFRD) contains 5,000 masked images and 90,000 no-masked images of 525 identities. The Masked Face Detection Dataset (MFDD) includes 24,771 masked face images. The Simulated Masked Face Recognition Dataset (SMFRD) consists of 500,000 face images of 10,000 identities. We cannot use these datasets to train and evaluate our work since real masked images are small and imbalanced. Furthermore, there have no corresponding images for the simulated dataset are provided. Boutros et al. [21] generated a masked face image version of MS1MV2 [5] that consists of 58 million images of 85,000 identities noted as MS1MV2-Masked. However, they do not provide it for public usage. Anwar and Raychowdhury [10] proposed a tool for masking faces in images so-called MaskTheFace. MaskTheFace can generate the masked face based on a face landmarks detector to construct masked face datasets. Deng et al. [12] introduced VGG-Face2\_m simulated dataset, which is generated from the VGG-Face2 face image dataset. VGG-Face2\_m contains around 3.3 million images and 9131 identities. They selected only forty images for each identity to generate masked images and combined them with corresponding images to get the VGG-Face2\_m dataset. We follow [10] to generate the new dataset for the training and validation of our model. Moreover, we also use the VGG-Face2\_m dataset to train our model for comparing performance. Figure

2.1 shows some samples of the available masked face dataset.

|  |
| --- |
|  |
| (a) |
|  |
| (b) |

**Figure 2.1** Samples of the masked face datasets. (a) The SMFRD dataset; (b) The VGG-Face2\_m dataset.

**Table 2. 1** A summary of occlusion and MFR approaches.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref | Model | Method | Require  -ment | Dataset |
| [18] | CNN-based PDSN | Pair-wise differential siamese network | ResNet-50 | CASIA-WebFace, LFW, AR, MegaFace |
| [31] | SRC | Image restoration-based | - | Extended Yale B, AR |
| [32] | ESRC | Image restoration-based | - | AR, FERET |
| [33] | SVGSRC | Image restoration-based | - | AR |
| [35] | NMR | Image restoration-based | - | Extended Yale B, AR, FRGC, EURECOM |
| [37] | GANs | Inpainting-based | DCGANs | CelebA, SVHN, Stanford Cars |
| [19] | GANs | Image reconstruction-based | VGG-19 | CelebA |
| [38] | GANs | Image reconstruction-based | VGG-19 | CelebA |
| [10] | MaskTheFace | MaskTheFace with FaceNet | FaceNet | VGGFace2-mini,  VGGFace2-mini-SM1, LFW-SM, MFR2 |
| [16] | CNNs | Re-train network-based | ResNet-50 | RMFRD |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ref | Model | Method | Require  -ment | Dataset |
| [40] | CenterFace, SphereFace, FaceNet, CosFace, ArcFace, MaskNet | Re-train network-based | ResNet-50 | Webface-OCC, LFW, LFW-mask, CFP-FP, CFP-FP-mask, AgeDB-30, AgeDB-30-mask, RMFRD |
| [53] | Pre-trained CNNs | Occlusion discarding-based | VGG-16, AlexNet,  ResNet-50 | RMFRD, SMFRD |
| [26] | CBAM | Face cropping-based | ResNet-50 | SMFRD, Webface, AR, Extended Yela B |
| [12] | MFCosface | Large margin cosine loss | Inception-ResNet-v1 | VGGFace2\_m, LFW\_m, CASIA-FaceV5\_m, MFR2, RMFD |

# Methodology

## 3.1 Convolutional Neural Network (CNN)

A variety of advancements in Computer Vision (CV) have occurred throughout the years. Especially with the introduction of Convolutional Neural Networks (CNNs), many research approaches are getting state-of-the-art results on image classification and image recognition problems. To handle the complexity of the data, the researchers tend to make a deeper neural network (increasing the number of layers) to solve such complex tasks and improve the classification/recognition accuracy.

* **Basic architectures of CNN**

CNN, also called ConvNet, is a deep learning algorithm that can take an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and differentiate one from the other. The basic architectures of CNN are primarily comprised of three layers: convolutional layer, pooling layers, and fully connected layer. A rough draft of CNN architecture for image classification is illustrated in Figure 3.1.

|  |
| --- |
|  |

**Figure 3.1** The basic CNN architecture for image classification.

The convolutional layer plays an essential role in CNN operation. The parameters of these layers focus on the use of learnable kernels, which are usually small in spatial dimensionality. However, spread along the entirety of the depth of the input. The convolutional layer can significantly reduce the complexity of the model by optimizing its output from three hyperparameters depth, stride, and zero-padding. Parameter sharing is another feature of the convolutional layer, which works on the assumption that if one region feature is helpful to compute at a set spatial region, then it is likely to be useful in other regions. Thus, the convolutional layer produces a massive reduction in the number of parameters.

The pooling layer aims to gradually reduce the dimensionality of the feature maps and thus further reduce the number of parameters and the computational complexity of the model. The pooling layer operates over each activation map in the input and scales its dimensionality using the max function. In most CNNs, these are implemented in the form of max-pooling layers with kernels of dimensionality applied stride of 2 along the spatial dimensions of the input. Therefore, this scales the feature maps down to 1/4 of the previous size while maintaining the depth volume to its standard size.

At the top of the stack, several regular, fully connected layers (also known as dense layers) are added. The fully connected layer contains neurons directly connected to the other neurons in the two adjacent layers without connecting them. This is analogous to how neurons are arranged in traditional artificial neural networks (ANN) forms.

* **Popular CNN model**

Various CNN models are devised for image classification and further applied in other CV tasks. For instance, AlexNet [54], VGGNet [55], GogLeNet [56], and ResNet [25]. The ResNet is emphatically introduced here because it is employed in this thesis.

ResNet, short for Residual Network, is again a specific type of neural network introduced in 2015 by Kaiming He et al. [25] in their paper “Deep Residual Learning for Image Recognition.” It has multiple depth variations, i.e. 18, 34, 50, 101, 152 layers. Two or more digits follow the name ResNet after the ResNet architecture with a certain number of neural network layers. The ResNet architecture comprises a stack of residual blocks; thus, it can build a very deep layer compared to other networks. The main idea of ResNet is to make skip connections between different layers in the residual block. The residual block consists of several convolutional layers and a global average pooling layers. Only one fully connected layer is connected with 1000 neurons for image classification at the end of ResNet. Different from VGGNet, there is no dropout layer in ResNet. A residual block for ResNet is illustrated in Figure 3.2.

|  |
| --- |
| Diagram  Description automatically generated |

**Figure 3.2** The residual block of ResNet.

As the main idea of ResNet is to skip connections between different layers in the residual block, we can go a little more about the residual block. Residual blocks are stacks of layers set so that the output of a layer is taken and added to another layer deeper in the block. In theory, having a deeper network should only help; however, in reality, the deeper network has higher training error and thus test error. When deeper networks start converging, a degradation problem has been exposed: accuracy gets saturated. It degrades rapidly with the network depth increasing. Using deeper networks degrades the performance of the model. The key idea is that, instead of letting layers learn the underlying mapping, let the network fit the residual mapping. So, instead of, say  , initial mapping,  which gives  The approach of this technique is to add a shortcut or a skip connection that allows information to flow more easily from one layer to the next’s next layer. Therefore, researchers have researched and created Residual Block

**Table 3.1** Architecture of different ResNet blocks. Building blocks are shown in brackets with the number of blocks stacked.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer name | output size | 18-layer | 34-layer | 50-layer | 101-layer |
| conv1 |  |  | | | |
| con2\_x |  |  | | | |
|  |  |  |  |
| con3\_x |  |  |  |  |  |
| con4\_x |  |  |  |  |  |
| con5\_x |  |  |  |  |  |
|  |  |  | | | |
| FLOPs | |  |  |  |  |

## 3.2 Feature Extraction Network

Feature extraction is a crucial process in the FR task. It aims to extract the critical face components such as eyes, nose, mouth, and texture from the face image. However, this process becomes more complicated with the mask covering the face. So, choosing the feature extracting network is a critical decision. We select the refined CNN architecture ResNet-50 implemented by ArcFace work as a backbone to extract the feature. It is one of the best trade-offs between accuracy and the number of parameters. The network configuration is shown in Table 3.2. We follow [5] to modify the layer block in the third stage from the original ResNet-50 [25] architecture [3, 4, 6, 3] to [3, 4, 14, 3] layer blocks. Besides that, we use the improve the residual unit, which has a BN-Conv-BN-PReLu-Conv-BN structure and set stride = 2 for the second convolutional layer instead of the first one (as in Figure 3.3). After the last layer, we employ the batch normalization, dropout, fully connected layer, and batch normalization (BN-Dropout-FC-BN) structure to achieve the final 512-D embedding feature as output.

**Table 3.2** Architecture of different ResNet blocks. Building blocks are shown in brackets with the number of blocks stacked.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| layer name | output size | 18-layer | 34-layer | 50-layer | 100-layer |
|  |  |  |  |  |  |
| stem |  |  |  |  |  |
| con1\_x |  |  |  |  |  |
| con2\_x |  |  |  |  |  |
| con3\_x |  |  |  |  |  |
| con4\_x |  |  |  |  |  |
| FC |  |  |  |  |  |
| #Params(M) | | 24.03 | 34.14 | 43.59 | 65.16 |

|  |
| --- |
| Graphical user interface, text, application, chat or text message  Description automatically generated |

**Figure 3.3** Structure of the improved residual unit: BN-Conv(stride=1)-BN-PReLu-Conv(stride=2)-BN.

## 3.3 Attention Modules

We use the Convolutional Block Attention Module (CBAM) presented by Woo et al. [26] in our work. CBAM consists of Channel Attention Module and Spatial Attention Module, arranged in a particular order. It is a lightweight module that can smoothly integrate with any DCNN architecture. Given an input feature map  of the convolutional layer,  denotes a 1D channel attention map and  denotes a 2D spatial attention map. The overall CBAM process shows as follows:

|  |
| --- |
| A picture containing text  Description automatically generated |

**Figure 3.4** The overall structure of the CBAM.

|  |  |
| --- | --- |
|  | (2) |

where  denotes element-wise multiplication and  is the final output of the feature maps or refined feature maps.

### 3.3.1 Channel Attention Module

The channel attention module focuses on ‘what’ is the meaning of the input image. It utilizes the relationship of the features between channel to channel. This module used average pooling  and max pooling  operations to generate two spatial information vectors. Both vectors are consecutively forwarded to a shared network multi-layer perceptron (MLP) with one hidden layer and filter kernel size  to produce a channel attention map . Then, the output feature vectors from the shared network are merged using element-wise submission. The final output after element-wise submission is passed to the sigmoid function  to generate the channel weights. The channel attention module process shows as follows:

|  |
| --- |
| Logo  Description automatically generated |

**Figure 3.5** The structure of the channel attention module.

|  |  |
| --- | --- |
|  | (2) |

where  is the sigmoid function, and MLP uses the ReLu activation function.

### 3.3.2 Spatial Attention Module

The spatial attention module focuses on ‘where’ is an informative region of the input images features. It utilized the relationship between spatial and spatial. Similar to the channel attention module, the spatial attention module adopts average pooling  and max pooling  operations to obtain two  maps. Those are then concatenated with a convolution layer with a filter kernel size  to obtain a  spatial attention map  The spatial attention module process shows as follows:

|  |
| --- |
| A picture containing text, clock  Description automatically generated |

**Figure 3.6** The structure of the spatial attention module.

|  |  |
| --- | --- |
|  | (2) |

where  is the sigmoid function and  denotes a convolution operation with the filter kernel size  .

## 3.4 Network Architecture

The overall proposed network architecture diagram is shown in Figure 3.8. As mentioned in section 3.2, this work employs refined ResNet-50 architectures as a backbone to extract features. We use the no-masked and masked image as the input with the size . The network backbone architecture consists of four main convolutional layer block stages with the number of blocks stacked. Therefore, the number of blocks stacked in the first, second, third, and fourth stages is [3, 4, 14, 3]. The size of the feature maps in the first, second, third, and fourth stage is ,, , and  with kernel size , respectively. For CBAM, it is applied in each output of the convolutional block of the backbone network to focus more on an object of interest effectively.  represents the feature maps after the pre-operation of the convolution. Then the channel and spatial attention modules compute sequentially to produce refined feature maps . Finally, the refined output features  are summed with the input feature maps of the previous block. The network repeats the same operation until the last convolutional layer block. We apply batch normalization (BN), dropout, and fully connected layers to get 512-D embedding features. ArcFace adds an angular margin  to the target (ground truth), and multiplies by the feature scale  . Then, go through the softmax function and contribute to the cross-entropy loss. This technique helps optimize the embedding feature to obtain highly discriminative features for MFR.

|  |
| --- |
|  |

**Figure 3.7** The overall structure of the proposed network architecture. The convolutional block attention module (CBAM) is integrated into each output of the block.

## 3.5 Loss Function

In DCNN, the loss function plays a critical role in learning discriminative features. It helps to optimize the model and stabilize the training process. Softmax loss is the traditional loss function most widely used in the classification task. The softmax function is presented as follows:

|  |  |
| --- | --- |
|  | (1) |

where denotes the deep feature vector of the -th sample belonging to the -th class and  is the feature dimension.  denotes the -th column of the weight and  is the bias.  and  are the batch size and class number, respectively. Although traditional softmax loss is commonly used in deep face recognition [51, 52], it has drawbacks. The linear transformation matrix  grows linearly with the class number ; the learned features are separable for the closed-set classification task. However, not discriminative enough for the open-set face recognition task. Moreover, face representation using DCNN features embedding is the preferred method for face recognition [1, 2, 5, 51, 52]; the softmax loss function, on the other hand, does not explicitly optimize the feature embedding to impose more similarity intra-class samples and diversity for inter-class samples [5]. Researchers are encouraged to develop other losses or modify the current softmax loss function [2,4,5].

Several modifications [2, 4, 5] have been developed to improve the softmax loss’s discriminative power. We use Additive Angular Margin Loss (ArcFace) [5], one of the most popular loss functions in our work. ArcFace loss is an angular margin loss function constructed by modifying the softmax loss function. It helps to optimize the feature embedding and improve the discriminative power of the model. Furthermore, ArcFace optimizes the feature embedding to have a smaller distance between the same classes as much as possible and a higher distance between the different classes as further as possible. Because ArcFace loss is modified from softmax loss, the authors set the bias  and weight  , where  is the angle between the weight  and the deep feature  The softmax loss can be rewritten as follow:

|  |  |
| --- | --- |
|  | (1) |

Now is normalized by  the normalization making . Similarly, the author also used  normalization to normalize and re-scale it to , making . So now the loss can be rewritten as:

|  |  |
| --- | --- |
|  | (1) |

The feature  and weight lie on the surface feature dimension  hypersphere with a radius of  1, respectively. To minimize the loss function, the authors maximize and minimize  for all , and. Where is the angle between the weight of the true logit and the embedding feature . The softmax function is slightly modified as follows:

|  |  |
| --- | --- |
|  | (1) |

where  is the true logit. Add an angular margin penalty  between and  to simultaneously enhance the embedding features. That means the embedding features belonging to the same class gather together, and those belonging to different classes get away from each other. Finally, the ArcFace loss function is defined as follows:

|  |  |
| --- | --- |
|  | (1) |

# Experiments and Results

## 4.1 Datasets

In the past years, many face recognition datasets have been available publicly. It provides possibilities for human facial recognition methods based on deep learning to achieve superior accuracy. The datasets include CASIA-WebFace [57], CelebA [58], VGGFace2 [52], MS-Celeb-1M [59]. In this research thesis, we selected CASIA-WebFace [57], one of the most popular datasets for the training model.

### 4.1.1 Proposed Masked Face Dataset

Many research areas use deep learning methods. However, using deep learning methods requires large-scale data samples for training. There are no widely used standard benchmarks of masked face datasets available to the public. Although the deep learning method has been performing well in recent years, the lack of masked image datasets has made it difficult for a model to learn all significant feature maps. We develop a new simulated masked face dataset generated from the existing face recognition dataset to handle these difficulties. As mentioned above, we selected CASIA-WebFace [57] dataset as our model training dataset for MFR tasks. Next section, we will describe datasets generation in more detail.

### 4.1.2 Dataset Generation

In this section, we talk about the process of datasets generation. As shown in Figure 4.2, there are many steps to generate the simulated masked face datasets. Anwar and Raychowdhury [10] proposed a toolkit for masking faces in images so-called MaskTheFace. MaskTheFace can generate the masked face based on a face landmarks detector to construct masked face datasets. This tool supports different types of masks as well. However, the masked face images generated by this tool are not well aligned, resized, and rotated, which makes it difficult for model training. We need to handle more work and integrate with that tool to produce better and more efficient datasets. Kaipeng Zhang et al. [60] proposed a deep cascaded multitask CNNs-based framework, so-called Multi-task Cascaded Convolutional Networks (MTCNN), for joining face detection and alignment in unconstrained environments. First, we used MTCNN to detect faces from the raw images and obtain five facial landmark key points: nose, right-eye, left-eye, right-mouth, left-mouth, and then we aligned, rotated, and resized the face image to pixels. MTCNN detects faces and obtains five key points. It is still challenging to overlay a mask on the face and generate more realistic masked face datasets. Therefore, to obtain more detail on face detection, we use the Dlib library to detect 68 key points of the face [61]. Lastly, we calculate the mask positions of the face and select the suitable mask template. We use standard mask templates (surgical, cloth, N95, KN95) and overlay different patterns and colors to cover the masks on the faces. Figure 4.1 shows the sample of standard mask templates.

|  |
| --- |
|  |

**Figure 4.1** The standard mask templates.

|  |
| --- |
|  |

**Figure 4.2** The generation diagram of the masked face dataset.

### 4.1.3 Training Dataset

In this work, we conduct the model training on the CASIA-WebFace\_m dataset. CASIA-WebFace\_m is generated from CASIA-WebFace [57] dataset. This dataset is a large-scale public face recognition dataset. It contains 494,414 images of 10,575 unique identities. The generated masked face image version is combined with the original regular face images to produce CASIA-WebFace\_m for the training model. It means the total training samples are double the size of the original dataset.

We select this dataset over other datasets depending on three main points:

* It is a well-known large-scale face dataset widely used in face recognition tasks and available in public;
* It is a dataset that creates with the specific purpose to motivate more researchers to build more new face datasets or enlarge this existing dataset;
* Easy to train with high-performance baseline deep CNNs for face recognition.

|  |
| --- |
|  |

**Figure 4.3** Sample images from CASIA-WebFace\_m dataset.

### 4.1.4 Validation Dataset

We conduct the model evaluation on four datasets in this work such as LFW\_m, AgeDB-30\_m, CFP-FP\_m, and MFR2. Three datasets are generated from the most widely used dataset benchmark for facial recognition LFW [62], AgeDB [63], and CFP [64], respectively; MFR2 [10] is a real masked face dataset.

* LFW\_m is generated from the LFW dataset. LFW is a public benchmark dataset, and It is the most commonly used for face verification. This dataset contains 5,749 unique identities and a total of 13,233 face images. In the experiments, this paper follows the LFW standard protocol using 6,000 pre-defined comparison pairs, of which 3,000 pairs have the same identities, and another 3,000 pairs have different identities.
* AgeDB-30\_m is generated from the public benchmark dataset AgeDB. AgeDB is an unconstrained face recognition dataset and is most commonly used for cross-age face verification. This dataset contains 568 unique identities and a total of 16,588 face images. In the experiment, this paper follows the protocol of AgeDB-30 using 6,000 pre-defined comparison pairs, of which 3,000 pairs have the same identities, and another 3,000 pairs have different identities.
* CFP-FP\_m is generated from the public benchmark dataset CFP. The 500 celebrities in frontal and profile view are in the CFP dataset. This dataset has two verification protocols, CFP-FF and CFP-FP. In the experiment, this paper follows the CFP-FP protocol using 7,000 pre-defined comparison pairs, of which 3,500 pairs have the same identities, and another 3,500 pairs have different identities.
* MFR2 is a small real masked face dataset. It contains 53 identities of celebrities and politicians with 269 images, and each identity has an average of five images. This dataset consists of strange mask patterns. This paper selects 800 pairs of images as real masked face verification in the experiment. It means 400 pairs have the same identities, and 400 pairs have different identities.

**Table 4.1** Summary of datasets used for model training and evaluation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **Type** | | **Identities** | | **Images** | **Pairs** |
| CASIA-WebFace\_m | Simulated | | 10,575 | | 789,296 | - |
| LFW\_m | Simulated | | 5,749 | | 12,000 | 6,000 |
| AgeDB-30\_m | Simulated | | 568 | | 16,588 | 6,000 |
| CFP-FP\_m | Simulated | | 500 | | 7,000 | 7,000 |
| MFR2 | Real masked image | | 53 | | 269 | 800 |
|  | |  | |  | | |
| LFW\_m | | AgeDB-30\_m | | CFP-FP\_m | | |

**Figure 4.4** Sample images of validation datasets LFW\_m, AgeDB-30\_m, and CFP\_FP\_m. Each dataset has two columns of corresponding images.

|  |
| --- |
|  |

**Figure 4.5** Sample images of real masked face validation datasets MFR2. Each identity has two columns of images.

## 4.2 Experimentation Details

Initially, this work follows [5] to generate the normalized face crops (112 × 112) in the data processing. This paper applies Barch-Normalization (BN) [65]-Dropout [66] structure after the last convolutional layer to get the output embedding feature 512D. Dropout can effectively help avoid over-fitting and obtain a better generalization for deep face recognition. In the experiment, the dropout parameter is set to 0.4. Feature scale ** and angular margin penalty  are 64 and 0.5, respectively. All experiments in this work are implemented by Pytorch [67], an open-source deep learning framework. The batch size is 128, and the model trains on NVIDIA Quadro RTX 6000 (48GB) GPUs. The overall model architecture is trained up to 100 epochs, and the only CASIA-WebFace\_m dataset is used to train the model. The learning rate is 0.01 and divided by ten at 13 and 21 epochs. In addition, this paper sets the momentum to 0.9 and weight decay 5e-4.

## 4.3 Evaluation Metrics

### 4.3.1 Accuracy

Accuracy is one of the metrics to describe the accuracy of an algorithm on a classification task. It is the number of samples that are paired divided by the number of samples. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively.

### 4.3.2 Precision

Precision is a metric that determines the number of accurate positive predictions. Precision, therefore, computes the accuracy for the minority class. It is computed as the ratio of correctly predicted positive samples divided by the predicted number of positive samples. There are two types of classification tasks that use precision metrics. One is binary classification, and the other one is multi-class classification. In an imbalanced classification issue with binary classification, precision is computed as all True Positives divided by the total number of True Positives sum with False Positives. The result is a value between (no precision) and (full or perfect precision). It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

Precision is not restricted to binary classification issues. In an imbalanced classification issue with more than two classes (multi-class classification), precision is measured by dividing the total number of True Positives across all classes by the total number of True Positives and False Positives. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

where TP denotes true positive, and FP denotes false positive.

### 4.3.3 Recall

The recall is a metric that measures the number of correct positive predictions made from all positive predictions that could have been made. As opposed to the precision that only comments on the correct positive predictions out of all positive predictions, it indicates missed positive predictions. Not different from precision, recall is also used in binary and multi-class classification. In an imbalanced classification issue with two classes, recall is computed as all True Positives divided by the total number of True Positives sum with False Negatives. The result is a value between (no recall) and (full or perfect recall). It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

The recall function is not restricted to binary classification issues. In an imbalanced classification issue with more than two classes, recall is computed as the total number of True Positives across all classes divided by the sum of True Positives and False Negatives across all classes. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

where TP denotes true positive, and FN denotes false negative.

### 4.3.3 F1 score

The F1 score allows combining precisions and recalls into a single measure that captures both properties. Alone, neither precision nor recall tells the whole story. We can have high precision with poor recall, or alternately, terrible precision with perfect recall. The F1 score comes up with a way to express precision and recall with a single score. Once precision and recall have been calculated for a binary or multi-class classification problem, the two scores can be combined into calculating the F-1 score. It can be computed as follows:

|  |  |
| --- | --- |
|  | (2) |

## 4.4 Evaluation Results

In this section, the reports of the model evaluation results are described. We performed experiments in the face verification task and used the 10-fold cross-validation technique to evaluate the predictive model by randomly dividing the evaluation dataset into ten partitions. Nine partitions turn into a training set, remaining as a validation set. The model was evaluated on simulated masked face images LFW\_m, AgeDB-30\_m, CFP-FP\_m, and real masked face images MFR2. The model extracts features of all face pairings and then computes the cosine similarities between the face pairs. The accuracy is expressed as a percentage of right predictions, with the highest accuracy being chosen as the threshold. The verification performance results of the 10-fold cross-validation are shown in Table 4.2. Table 4.3 reported measuring the performance of the model with precision, recall, and f1 score metrics.

**Table 4.2** Verification accuracy results (%) of 10-fold cross-validation on the validation dataset.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **10-fold** | **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** | |
| 1 | 99.83 | 97.14 | 96.67 | 92.50 | |
| 2 | 99.33 | 96.71 | 98.00 | 100.00 | |
| 3 | 99.33 | 97.00 | 98.17 | 96.25 | |
| 4 | 99.50 | 96.57 | 99.86 | 96.25 | |
| 5 | 99.33 | 99.00 | 96.50 | 98.75 | |
| 6 | 99.17 | 96.14 | 96.83 | 95.00 | |
| 7 | 99.33 | 96.86 | 97.50 | 95.00 | |
| 8 | 99.33 | 97.00 | 95.33 | 96.25 | |
| 9 | 99.67 | 98.29 | 97.00 | 96.25 | |
| 10 | 99.33 | 96.43 | 97.33 | 96.25 | |
| Average | **99.41** | **97.11** | **96.88** | **96.25** | |
|  | | | | |
| **Figure 4.6** The accuracy curve of the training model | | | | |

|  |
| --- |
|  |

**Figure 4.7** The loss curve of the training model.

As reported in Table 4.2, the proposed method achieved high performance in the face verification task. The average accuracy rate of 10-fold cross-validation on the LWW\_m, AgeDB-30\_m, and CFP-FP datasets reached, 99.41%, 97.11%, 96.88%, respectively. However, MFR2 achieved 96.25% since this dataset contained different facial postures, expressions, and cloth masks in different textures and colors.

**Table 4.3** Results of accuracy, precision, recall, and f1 score (%).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 score** |
| LFW\_m | 99.41 | 99.26 | 99.56 | 99.40 |
| AgeDB-30\_m | 97.11 | 95.02 | 99.17 | 97.05 |
| CFP-FP\_m | 96.88 | 95.73 | 97.99 | 96.85 |
| MFR2 | 96.25 | 95.50 | 97.05 | 95.55 |

### 4.4.1 Comparison with FR methods

We conducted experiments with other state-of-the-art FR methods. Only our method used the CASIA-WebFace\_m dataset, and other methods used the original CASIA-WebFace dataset from scratch. We compared the verification accuracies by validating on the same validation dataset. The result reports are shown in Table 4.4.

**Table 4.4** Comparison of face verification results (%) on validation dataset with different methods.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Training**  **Dataset** | **Validation Dataset** | | | |
| **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| CosFace [4] | CASIA-Webface | 95.23 | 93.40 | 92.21 | 63.00 |
| Softmax [5] | CASIA-Webface | 96.68 | 93.50 | 94.78 | 69.75 |
| ArcFace [5] | CASIA-Webface | 96.85 | 94.10 | 95.10 | 71.87 |
| **Ours** | CASIA-Webface\_m | **99.41** | **97.11** | **96.88** | **96.25** |

As reported in Table 4.4, we observed that our method yielded much better results in both generated masked face images and real masked face images. The accuracy rates with generated images results of the existing FR methods are high and comparable. However, the accurate rates with real mask images drop down remarkably. These results prove that our proposed method is superior to the other FR methods.

### 4.4.2 Comparison with MFR methods

Several MFR methods are performed with their proposed training and validation dataset. In this section, we separated the comparison into two parts. We compared our results with other results in the first part, as shown in Table 4.5.

**Table 4.5** Comparison of face verification (%) with different methods.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Training**  **Dataset** | **Validation Dataset** | | | |
| **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| Huang et al.  [40] | WebFace-OCC | 97.08 | 87.18 | 86.07 | - |
| Anwar et al.  [10] | VGGFace2-mini-SM | 97.25 | **-** | **-** | 95.99 |
| Ours | CASIA-WebFace\_m | **99.41** | **97.11** | **96.88** | **96.25** |

In the second part, we conducted another experiment to compare the current method MFCosface [12] with their masked dataset VGG-Face2\_m. We followed them using 400 pairs of the MFR2 dataset for face verification. The verification performance results of the 10-fold cross-validation are shown in Table 4.6; accuracy, precision, recall, and f1 score results are shown in Table 4.7. The accuracy comparison results with the MFCosface method are shown in Table 4.8.

**Table 4.6** Verification accuracy results (%) of 10-fold cross-validation on the validation dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **10-fold** | **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| 1 | 99.67 | 95.50 | 98.29 | 100.00 |
| 2 | 99.67 | 96.00 | 96.14 | 95.00 |
| 3 | 99.33 | 96.50 | 97.86 | 100.00 |
| 4 | 99.33 | 95.33 | 96.43 | 100.00 |
| 5 | 99.50 | 95.17 | 98.43 | 97.50 |
| 6 | 99.33 | 95.17 | 97.14 | 100.00 |
| 7 | 98.67 | 96.50 | 94.86 | 100.00 |
| 8 | 99.00 | 94.33 | 96.57 | 97.50 |
| 9 | 100.00 | 94.50 | 97.71 | 100.00 |
| 10 | 99.67 | 94.83 | 96.43 | 100.00 |
| Average | **99.41** | **95.38** | **96.98** | **99.00** |

**Table 4.7** Results of accuracy, precision, recall, and f1 score (%) with the VGG-Face2\_m dataset.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 score** |
| LFW\_m | 99.41 | 99.26 | 99.56 | 99.40 |
| AgeDB-30\_m | 95.38 | 93.10 | 98.11 | 95.53 |
| CFP-FP\_m | 96.98 | 96.17 | 98.40 | 97.27 |
| MFR2 | 99.00 | 95.50 | 98.54 | 99.02 |

**Table 4.8** Comparison of face verification (%) with different methods.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Training**  **Dataset** | **Validation Dataset** | | | |
| **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| MFCosface | VGG-Face2\_m | 99.33 | **-** | **-** | 98.50 |
| Ours | VGG-Face2\_m | **99.41** | **95.38** | **97.30** | **99.00** |

### 4.4.3 Ablation Experiment

In order to prove the effectiveness of our method, ablation experiments were performed. We applied all experimental settings, such as image size, batch size, learning rate, and more, same as our previous experiments. First, we experimented with the CBAM module on our masked face dataset and then explored each attention module with the backbone. We search for an effective approach to channel attention and then spatial attention with our backbone network. After then, we evaluated the model on all validation datasets. Table 4.9 shows the performant reports of the ablation experiments.

**Table 4.9** Ablation experiment results of the attention module.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **LFW\_m** | **AgeDB-30\_m** | **CFP-FP\_m** | **MFR2** |
| CBAM | 98.38 | 93.85 | 95.97 | 94.62 |
| Backbone + channel | 98.98 | 95.26 | 97.37 | 94.75 |
| Backbone + spatial | 98.96 | 95.38 | 96.67 | 93.37 |
| **Ours** | **99.41** | **97.11** | **96.88** | **96.25** |

# Conclusion and Future Works

This thesis contributed to solving the masked face recognition problems using deep learning technology. Traditional methods based on deep learning can address common face recognition problems and achieve high performance. However, the performance drops dramatically when a face image is covered with a mask. This work introduced new network architecture based on an attention mechanism and an angular margin loss function. New simulated masked face images were generated by a data augmentation tool for model training and evaluation to handle the insufficient masked face datasets. The refined CNNs architecture ResNet-50 was employed as a backbone to extract face features. The attention module was adopted into each ResBlock to focus on the most informative part around the eyes of the masked face images and obtain more discriminative feature information. Angular margin loss ArcFace optimized the feature embedding to increase similarity for the intra-class samples and diversity for the inter-class samples.

Experiments in training and validation on the generated masked face and real masked image datasets found that the proposed method provided outstanding performance and a better recognition rate. Despite the success of this approach, there is still room for improvement. Future research is to improve the method to solve masked faces recognition with different postures, expressions, illumination, and wearing a hat.

**Acknowledgment**

Time flies and my master’s degree is coming to an end. It has been my great honor to study and grow in Software Engineering Laboratory at Jeonbuk National University over the past few years. Many people have participated in my growing, helping, and motivating in my studies. Taking this great opportunity, I would like to express my deep gratitude.

My heartfelt gratitude goes first to my supervisor, Professor Hyo Jong Lee, for his relentless mentoring, encouragement, and scholarship support. His professional dedication and vast knowledge affected me greatly and became my study example for my life. In addition, he guides the skills and guides the meaning of life, being a good person, strong, and successful in life.

Many thanks to the lecturers, Sithy Cheam and Borey Ka, who are my lecturers in the undergraduate program, and for helping to issue a letter of recommendation to receive a scholarship for a master’s degree at Jeonbuk National University.

Thanks to all my lab mates for creating a cheerful working atmosphere and tremendous support. Thanks to Dr. ThaiLeang Sung, who has helped me since joining the lab. Thanks to Weiguo Wan, Yue Que, Seungtaek Kim, Khwaja Monib Sediqi, Shuli Xing, who have graduated currently, but we had a pleasant time together; Thanks to Areszoo Valavi, Suli Li, Chenmou Wu, Yulan Zhao, Jiahua Wu, Yingmei Zhang, Aung Si Min Htet, Nindam Somsawut, Suhyeon Lee, Jianli Zhao, Kalupahanage Dilusha Malintha De Silva. Thanks to them for their assistance during my study period.

I sincerely thank my family, relatives, friends, and particularly my love for their constant support. Their encouragement is powerful for me to overcome all of my hard times.

Finally, I would like to commend the financial support from the Brain Korea 21 Plus Program (BK21) and the Korean National Research Foundation.

**References**

[1] F. Schroff, D. Kalenichenko, and J. 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.

[2] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212-220.

[3] B. Liu *et al.*, "Fair loss: Margin-aware reinforcement learning for deep face recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10052-10061.

[4] H. Wang *et al.*, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265-5274.

[5] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690-4699.

[6] Y. Huang *et al.*, "Curricularface: adaptive curriculum learning loss for deep face recognition," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5901-5910.

[7] S. Khan *et al.*, "Emergence of a novel coronavirus, severe acute respiratory syndrome coronavirus 2: biology and therapeutic options," *Journal of clinical microbiology,* vol. 58, no. 5, pp. e00187-20, 2020.

[8] D. K. Chu *et al.*, "Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: a systematic review and meta-analysis," *The lancet,* vol. 395, no. 10242, pp. 1973-1987, 2020.

[9] N. Damer, J. H. Grebe, C. Chen, F. Boutros, F. Kirchbuchner, and A. Kuijper, "The effect of wearing a mask on face recognition performance: an exploratory study," in *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2020, pp. 1-6: IEEE.

[10] A. Anwar and A. Raychowdhury, "Masked face recognition for secure authentication," *arXiv preprint arXiv:2008.11104,* 2020.

[11] D. Montero, M. Nieto, P. Leskovsky, and N. Aginako, "Boosting masked face recognition with multi-task arcface," *arXiv preprint arXiv:2104.09874,* 2021.

[12] H. Deng, Z. Feng, G. Qian, X. Lv, H. Li, and G. Li, "MFCosface: a masked-face recognition algorithm based on large margin cosine loss," *Applied sciences,* vol. 11, no. 16, p. 7310, 2021.

[13] M. Jiang, X. Fan, and H. Yan, "Retinamask: A face mask detector," *arXiv preprint arXiv:2005.03950,* 2020.

[14] G. Yang *et al.*, "Face Mask Recognition System with YOLOV5 Based on Image Recognition," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020, pp. 1398-1404: IEEE.

[15] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," *Measurement,* vol. 167, p. 108288, 2021.

[16] B. Mandal, A. Okeukwu, and Y. Theis, "Masked Face Recognition using ResNet-50," *arXiv preprint arXiv:2104.08997,* 2021.

[17] W. Hariri, "Efficient masked face recognition method during the covid-19 pandemic," *Signal, image and video processing,* pp. 1-8, 2021.

[18] L. Song, D. Gong, Z. Li, C. Liu, and W. Liu, "Occlusion robust face recognition based on mask learning with pairwise differential siamese network," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 773-782.

[19] N. U. Din, K. Javed, S. Bae, and J. Yi, "A novel GAN-based network for unmasking of masked face," *IEEE Access,* vol. 8, pp. 44276-44287, 2020.

[20] Y. Li, K. Guo, Y. Lu, and L. Liu, "Cropping and attention based approach for masked face recognition," *Applied Intelligence,* vol. 51, no. 5, pp. 3012-3025, 2021.

[21] F. Boutros, N. Damer, F. Kirchbuchner, and A. Kuijper, "Unmasking Face Embeddings by Self-restrained Triplet Loss for Accurate Masked Face Recognition," *arXiv preprint arXiv:2103.01716,* 2021.

[22] Z. Wang *et al.*, "Masked face recognition dataset and application," *arXiv preprint arXiv:2003.09093,* 2020.

[23] A. Cabani, K. Hammoudi, H. Benhabiles, and M. Melkemi, "MaskedFace-Net–A dataset of correctly/incorrectly masked face images in the context of COVID-19," *Smart Health,* vol. 19, p. 100144, 2021.

[24] A. Alzu’bi, F. Albalas, T. Al-Hadhrami, L. B. Younis, and A. Bashayreh, "Masked Face Recognition Using Deep Learning: A Review," *Electronics,* vol. 10, no. 21, p. 2666, 2021.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[26] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3-19.

[27] M. L. Ngan, P. J. Grother, and K. K. Hanaoka, "Ongoing Face Recognition Vendor Test (FRVT) Part 6A: Face recognition accuracy with masks using pre-COVID-19 algorithms," 2020.

[28] C. Y. Wu and J. J. Ding, "Occluded face recognition using low-rank regression with generalized gradient direction," *Pattern Recognition,* vol. 80, pp. 256-268, 2018.

[29] H. Qiu, D. Gong, Z. Li, W. Liu, and D. Tao, "End2End occluded face recognition by masking corrupted features," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2021.

[30] D. Zeng, R. Veldhuis, and L. Spreeuwers, "A survey of face recognition techniques under occlusion," *arXiv preprint arXiv:2006.11366,* 2020.

[31] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence,* vol. 31, no. 2, pp. 210-227, 2008.

[32] W. Deng, J. Hu, and J. Guo, "Extended SRC: Undersampled face recognition via intraclass variant dictionary," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, no. 9, pp. 1864-1870, 2012.

[33] J. Huang, F. Nie, H. Huang, and C. Ding, "Supervised and projected sparse coding for image classification," in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[34] L. Yuan and F. Li, "Face recognition with occlusion via support vector discrimination dictionary and occlusion dictionary based sparse representation classification," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 110-115: IEEE.

[35] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu, "Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes," *IEEE transactions on pattern analysis and machine intelligence,* vol. 39, no. 1, pp. 156-171, 2016.

[36] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems,* vol. 27, 2014.

[37] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5485-5493.

[38] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative face completion," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3911-3919.

[39] C. Li, S. Ge, D. Zhang, and J. Li, "Look through masks: Towards masked face recognition with de-occlusion distillation," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 3016-3024.

[40] B. Huang *et al.*, "When Face Recognition Meets Occlusion: A New Benchmark," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4240-4244: IEEE.

[41] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[42] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132-7141.

[43] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, "Ocnet: Object context network for scene parsing," *arXiv preprint arXiv:1809.00916,* 2018.

[44] J. Fu *et al.*, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146-3154.

[45] J. Dai *et al.*, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764-773.

[46] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, 2020, pp. 213-229: Springer.

[47] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2285-2294.

[48] B. Chen, W. Deng, and J. Hu, "Mixed high-order attention network for person re-identification," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 371-381.

[49] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794-7803.

[50] W. Du, Y. Wang, and Y. Qiao, "Recurrent spatial-temporal attention network for action recognition in videos," *IEEE Transactions on Image Processing,* vol. 27, no. 3, pp. 1347-1360, 2017.

[51] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015.

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

[53] W. Hariri, "Efficient masked face recognition method during the covid-19 pandemic," *arXiv preprint arXiv:2105.03026,* 2021.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems,* vol. 25, pp. 1097-1105, 2012.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[56] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[57] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923,* 2014.

[58] Z. Liu, P. Luo, X. Wang, and X. Tang, "Large-scale celebfaces attributes (celeba) dataset," *Retrieved August,* vol. 15, no. 2018, p. 11, 2018.

[59] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *European conference on computer vision*, 2016, pp. 87-102: Springer.

[60] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters,* vol. 23, no. 10, pp. 1499-1503, 2016.

[61] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1867-1874.

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

[63] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, "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.

[64] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, "Frontal to profile face verification in the wild," in *2016 IEEE winter conference on applications of computer vision (WACV)*, 2016, pp. 1-9: IEEE.

[65] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448-456: PMLR.

[66] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[67] A. Paszke *et al.*, "Automatic differentiation in pytorch," 2017.