applied sciences

MDPI

Article

MFCosface: A Masked-Face Recognition Algorithm Based on Large Margin Cosine Loss

Hongxia Deng [1], Zijian Feng [1,*], Guanyu Qian [1], Xindong Lv [1], Haifang Li [1] and Gang Li [2]

[1] Department of Information and Computer, Taiyuan University of Technology, Taiyuan 030024, China; denghongxia@tyut.edu.cn (H.D.); qianguanyu0395@link.tyut.edu.cn (G.Q.); lvxindong0593@link.tyut.edu.cn (X.L.); lihaifang@tyut.edu.cn (H.L.)
[2] College of Software, Taiyuan University of Technology, Taiyuan 030024, China; ligang02@tyut.edu.cn
* Correspondence: fengzijian0360@link.tyut.edu.cn

Abstract: The world today is being hit by COVID-19. As opposed to fingerprints and ID cards, facial recognition technology can effectively prevent the spread of viruses in public places because it does not require contact with specific sensors. However, people also need to wear masks when entering public places, and masks will greatly affect the accuracy of facial recognition. Accurately performing facial recognition while people wear masks is a great challenge. In order to solve the problem of low facial recognition accuracy with mask wearers during the COVID-19 epidemic, we propose a masked-face recognition algorithm based on large margin cosine loss (MFCosface). Due to insufficient masked-face data for training, we designed a masked-face image generation algorithm based on the detection of the detection of key facial features. The face is detected and aligned through a multi-task cascaded convolutional network; and then we detect the key features of the face and select the mask template for coverage according to the positional information of the key features. Finally, we generate the corresponding masked-face image. Through analysis of the masked-face images, we found that triplet loss is not applicable to our datasets, because the results of online triplet selection contain fewer mask changes, making it difficult for the model to learn the relationship between mask occlusion and feature mapping. We use a large margin cosine loss as the loss function for training, which can map all the feature samples in a feature space with a smaller intra-class distance and a larger inter-class distance. In order to make the model pay more attention to the area that is not covered by the mask, we designed an Att-inception module that combines the Inception-Resnet module and the convolutional block attention module, which increases the weight of any unoccluded area in the feature map, thereby enlarging the unoccluded area's contribution to the identification process. Experiments on several masked-face datasets have proved that our algorithm greatly improves the accuracy of masked-face recognition, and can accurately perform facial recognition with masked subjects.

Keywords: facial recognition; cosine; detection of key features; attention mechanism

Citation: Deng, H.; Feng, Z.; Qian, G.; Lv, X.; Li, H.; Li, G. MFCosface: A Masked-Face Recognition Algorithm Based on Large Margin Cosine Loss. *Appl. Sci.* **2021**, *11*, 7310. https://doi.org/10.3390/app11167310

Academic Editor: Enrico Vezzetti

Received: 15 July 2021
Accepted: 4 August 2021
Published: 9 August 2021

1. Introduction

As a convenient and fast method of identification, facial recognition technology has been widely used in the fields of public security, financial business, justice, and criminal investigation. Facial recognition technology extracts facial features for classification and recognition, and it is has been one of the hotspots of research in recent years [1].

The world today is being hit by COVID-19, which is an infectious virus that causes severe acute respiratory syndrome [2]. According to the CDC's instructions, the best ways to avoid infection or spread of the disease are to maintain social distance and wear a mask in public. The current identification methods based on ID cards and fingerprints require contact with specific sensors, and facial recognition technology can avoid this unnecessary contact to a certain extent, avoiding the spread of COVID-19. However, wearing a mask

affects the extraction of facial features [3,4], leading to low recognition accuracy, so the algorithm research on masked-face recognition has great practical significance at the moment [5].

Generally speaking, mask occlusion leads to the obstruction of the feature structure of the face, and the most important problem in facial recognition with masks is how to effectively represent the face when the feature structure is obstructed [6,7]. At present, most scholars use sparse representation and local feature extraction to solve the problem. Sparse representation is to use as few training samples as possible to re-represent the test object, that is, to seek the sparsest representation of the test sample [8]. Wen et al. [9] proposed structured occlusion coding, which separates the occlusion and classifies at the same time through an additional occlusion dictionary. Wu and Ding [10] used a hierarchical, sparse, low-rank regression model and proposed a SRC-based, gradient direction hierarchical adaptive sparse low-rank (GD-HASLR) model. Dong et al. [11] proposed a hybrid model that combines robust sparsity constraints and low-rank constraints. The model can simultaneously deal with random errors caused by random noise and structural errors caused by occlusion. However, due to the large-area occluded by a mask, the identity information is severely affected; a sparse representation is difficult to effectively reconstruct, making its recognition rate with masked-face images low.

The method based on local feature extraction mainly focuses on the relationships between local features and facial features [12]. Wang et al. [13] used high-dimensional local binary patterns to obtain local features of human faces, and used densely connected convolutional neural networks to extract human faces. Song et al. [14] proposed a robust facial recognition method for occlusion based on the pairwise differential siamese network, which explicitly establishes the relationship between some occluded facial section and the occluded feature. Shi et al. [12] decomposed feature embeddings, set different confidence values for the decomposed sub-embeddings, and aggregated the sub-embeddings for identity recognition. However, the feature extraction method mainly relies on artificially designed feature representations, so it has a low recognition rate in an unconstrained environment.

Many scholars have also made many other attempts at masked-face recognition [15–18]. Xie et al. [19] proposed the robust nuclear norm to characterize the structural error and a new robust matrix regression model composed of RMR and S-RMR. Ejaz et al. [20] used PCA for feature extraction and dimensionality reduction. They calculated the average facial features of each identity and performed identity recognition. Xu et al. [21] proposed a Siamese convolutional neural network for facial recognition based on the Siamese convolutional neural network and the Inception module. In the absence of masked-face datasets, most methods only simulate mask occlusion by adding random noise or black pixels, which makes their abilities with real mask occlusion questionable.

For the above problems, this paper proposes a masked-face recognition algorithm based on large margin cosine loss (MFCosface). It uses an algorithm based on the detection of key facial features to generate masked-face images as a training set; then it uses the large margin cosine loss to train the model; and finally, it adds an attention mechanism to the model to optimize the representations of facial features, which effectively solves the problem of low recognition rates with mask occlusion.

In summary, our contributions are as follows:

- We designed a masked-face image generation algorithm based on the detection of key facial features, and generate masked-face images from face datasets and mask templates. The images were used construct a dataset for training, which alleviated the problem of insufficient data.
- A masked-face recognition algorithm based on large margin cosine loss is proposed. We analyzed the characteristics of the masked-face dataset, and proved the rationality of using the large margin cosine loss function.

- Experiments on our artificial masked-face dataset and a real masked-face image dataset proved that our algorithm greatly improves the accuracy of masked-face recognition, and can accurately perform facial recognition in spite of mask occlusion.

2. Related Work

2.1. Face Alignment and the Detection of Key Facial Features

Face alignment is an important preprocessing method in a facial recognition algorithm. By constraining the geometric parameters of the face to reduce the differences arising from facial posture, one can effectively improve the robustness of a facial recognition network to facial posture changes. The existing face alignment algorithms can be mainly divided into two categories: generative methods and discriminative methods [22]. Generative methods regard face alignment as an optimization problem of fitting the appearance of a face [23], and generate an aligned facial image by optimizing the shape and appearance parameters [24]. Discriminative methods train multiple key facial feature detectors and infer face information from these feature points. As the size of the dataset increases, discriminative methods have shown obvious advantages in training and alignment speed, and have become the preferred methods of face alignment—e.g., take the commonly used algorithms MTCNN [25] and LAB [26].

Key facial features are also called facial landmarks [27]. They include the eyebrows, eyes, nose, mouth, facial contours, etc. The detection of key facial features is a key step in the field of facial recognition and analysis, and it is the key to other face-related issues, such as automatic facial recognition, expression analysis, three-dimensional face reconstruction, and three-dimensional animation.

2.2. Softmax Loss Function

Using a deep convolutional neural network (DCNN) for feature extraction for face representation is the preferred method of facial recognition [28,29]. DCNN performs a mapping operation on face images in a feature space with a small intra-class distance and a large inter-class distance. Some scholars trained a classifier based on the softmax loss function to separate different identities in the training set to solve the facial recognition problem. The softmax loss function is shown in Equation (1).

$$L_s = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{yi}^T xi+b_{yi}}}{\sum_{j=1}^{n} e^{W_j^T x_i+b_j}} \quad (1)$$

Among them, $x_i \in R^d$ represents the feature of the i-th sample, which belongs to the class y_i. $W_j \in R^d$ represents the j column of weight $W \in R^{d\times n}$; $b_j \in R^n$ is the deviation term; and the batch size and category are N and n, respectively. The method based on the softmax loss function can be trained on large-scale training data and deep convolutional neural networks to obtain excellent facial recognition performance, but this method also has many shortcomings:

1. The size of the linear transformation matrix $W \in R^{d\times n}$ increases linearly with the number of identities n. Since a $W \in R^{d\times n}$ matrix is used to output n identity prediction probabilities, the number of identities n in the large-scale test set is usually very large, so the matrix W will show a linear increasing trend with n, which will reduce the training efficiency of the model.
2. In the open-set facial recognition problem, a face cannot be fully distinguished. If a face image that has not been trained with before appears in the model recognition process, this method cannot distinguish it well, because the final output of the model does not include the probability of the identity never having appeared.
3. The softmax loss function does not explicitly optimize the characteristics of images to increase the similarity of the samples within the class and the diversity of the samples between the classes, which leads to a large number of appearance changes within each class. Hence, the performance for facial recognition is poor.

In order to solve these problems, many scholars have improved the softmax loss function. Wen et al. [30] proposed center loss, which can reduce the intra-class variance by optimizing the feature distance between the class center and the sample. Liu et al. [31] proposed the large margin softmax (L-softmax) loss function, which adds an angle constraint to each sample by increasing the margin, which effectively expands the distance between classes and compresses the distance within classes. A-softmax [32] adds multiple restrictions to the angle in the L-softmax loss function, normalizes the weights, and provides a good geometric explanation by constraining the learning features to be distinguishable on the hyperspherical manifold. Wang et al. [33] proposed large margin cosine loss (CosFace) to apply feature normalization and use the global scale factor s to replace the sample-related feature norm in A-softmax. CosFace converts the angular distance to the cosine distance to achieve the smallest intra-class variance and the largest inter-class variance possible, which effectively improves the recognition accuracy. Deng et al. [34] proposed the Arcface facial recognition model, which improved AM-softmax and replaced the cosine distance with the angular distance. This method improves the loss based on softmax, and a large number of experiments have proved that it has superior recognition accuracy rates in facial recognition.

2.3. Attention Model

The attention model (AM) was originally used in machine translation, and has now become an important concept in the field of neural networks [35–37]. The attention mechanism can be explained intuitively by using the human visual mechanism: we usually pay more attention to the specific things that attract our attention [38–40]. In deep network learning, the attention mechanism is shown to give higher weights to elements understood from intuition—that is, it allocates more resources to important parts, and allocates less resources to unimportant or bad parts. This is conducive to obtaining higher revenue from fixed computing resources [41].

3. Proposed Methods

3.1. Dataset Preprocessing

Since deep learning models usually require large-scale datasets for training, the lack of masked-face datasets makes it difficult for a model to learn the feature mapping when a face is occluded by a mask, resulting in a poor recognition rate. To solve this problem, we used a masked-face image generation algorithm based on the detection of key facial features to construct a dataset. We generated realistic masked-face images for model training through face detection, the detection of key features, and mask coverage analysis. The process of masked-face image generation is shown in Figure 1.

1. Face detection: We used a multi-task cascaded convolutional neural network (MTCNN) [25] for preprocessing, and got an image containing only faces. The result is shown in Figure 1. MTCNN is mainly composed of three convolutional neural networks, cascaded. First our system involves resizing the image and generating image pyramids of different scales; then we send them to P-Net to generate many candidate frames containing faces or partial faces; then we filter out a large number of poor candidate frames through R-Net and perform regression on the candidate frames to optimize the prediction results; finally, we use O-Net to regress the features and output the positions of the key features.
2. Key feature detection: MTCNN can only detect the key points of the eyes, nose, left mouth, and right mouth, and it is difficult to generate a more realistic masked-face image using only 5 key points, so we used HOG features to detect 68 key points of the face [42]. This method is more detailed in the detection of key points. Note that we used the Dilb library to implement this process.
3. Mask coverage: To generate a mask, we calculate the distance and structure information based on the relative positions of the chin and the bridge of the nose, get the coordinates of the mask, use common mask templates (surgical mask, KN95, etc.) to

cover the face, and generate the masked-face image. The result is shown in Figure 1. Anwar et al. [43] also used a method for generating masked-face images for model training, but their method needs to collect mask templates from different angles, which has significant limitations. Our method uses only a front image of the mask, analyzes the relative positions of the key points of the face, and distorts the mask to generate a more realistic masked-face image.

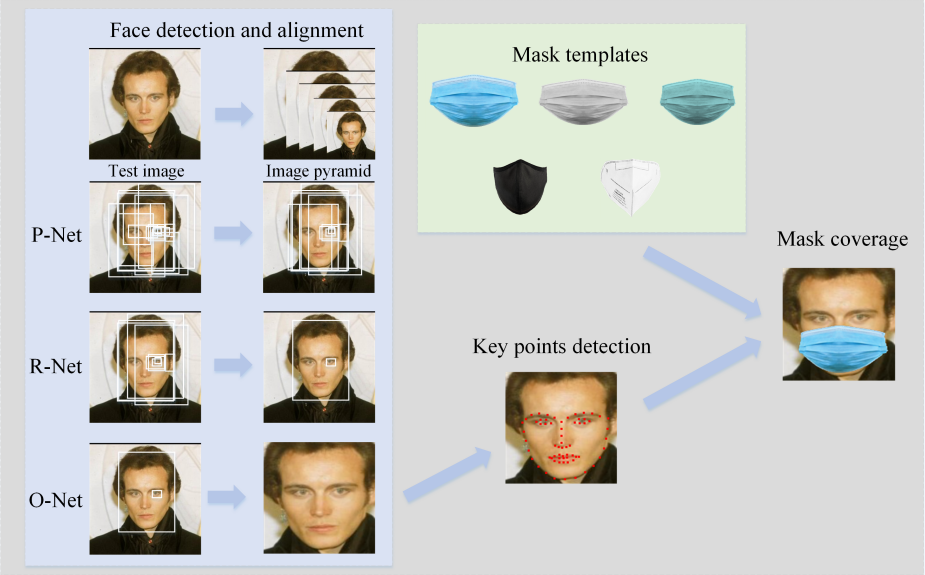

Figure 1. The process of masked-face image generation. First, the face in the test image is detected and aligned; then, the information about the chin and the bridge of the nose is output; finally, the mask templates are used to cover the face to generate the masked-face image.

3.2. Loss Function

FaceNet is a facial recognition model proposed by the Google team [44] which mainly uses triplet loss for model training. A triplet is composed of three samples (x^a, x^p, x^n), where x^a and x^p are two face images with the same identity (positive pair), where x^a and x^n are two face images with different identities (negative pair). Assuming that the mappings of x^a, x^p, and x^n in the feature space are, respectively, $f(x^a)$, $f(x^p)$, and $f(x^n)$, in order to make the feature distance of the same identity image smaller than the feature distances of different identity images, the following inequality can be used:

$$\|f(x^a) - f(x^p)\|_2^2 + \alpha < \|f(x^a) - f(x^n)\|_2^2 \tag{2}$$

where α is the margin between the positive pair and negative pair. Through the above formula, the feature distance between the positive pair can be forced to be much smaller than the feature distance between the negative pair—that is, the mapping of the same identity in the feature space is closer, and the mapping of different identities is farther. Hence, the triplet loss is as follows:

$$L_t = \left[\|f(x^a) - f(x^p)\|_2^2 - \|f(x^a) - f(x^n)\|_2^2 + \alpha\right]_+ \tag{3}$$

Since the network selects those valuable triplets for training as much as possible, the selection of two images with high similarity for the positive pair will make it difficult for the model to learn effective feature representation, and selecting the two most dissimilar images may lead to training collapse, so a semi-hard strategy is generally adopted. The semi-hard strategy selects two images with poor similarity to form a positive pair, and two images with higher similarity form a negative pair. Compared with extreme selection of images with the highest or lowest degree of difference to form triplet, this method is more balanced, and the model's iteration speed is faster.

According to the characteristics of the masked-face dataset, ideally, triplets such as (Anchor, Positive1, Negative) and (Anchor, Positive3, Negative) in Figure 2 will be selected for training, because they contain more mask changes; this helps the model learn the relationships between mask changes and facial feature changes. Since the sample feature distances were $L1 < L2 < L3$ in the feature space, the model chose triplets such as (Anchor, Positive2, Negative) for training after adopting the semi-hard strategy, which contains fewer mask changes. Therefore, it is difficult for the model to extract the facial features occluded by a mask.

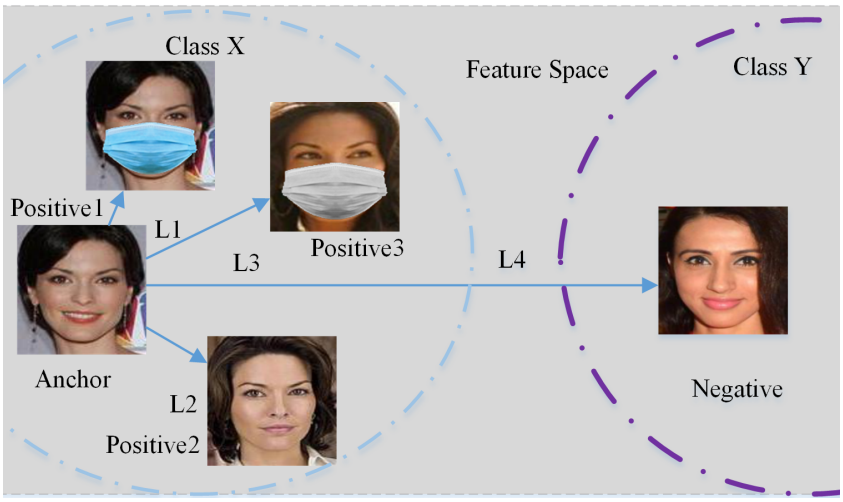

Figure 2. A representation of triplets in feature space. Anchor, Positive1, Positive2, and Positive3 belong to class X; Negative belongs to class Y; L represents the feature distance between different samples.

In order to solve this problem, we used the large margin cosine loss function to train the model. The large margin cosine loss function replaces the selection of triplet training with a non-grouping learning method, and its expression is as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cdot(\cos(\theta_{yi},i)-m)}}{e^{s\cdot(\cos\theta_{yi},i)-m)}+\sum_{j\neq yi}^{n}e^{s\cdot\cos(\theta_j,i)}} \tag{4}$$

subject to

$$\begin{aligned} W &= \frac{W*}{\|W*\|}, \\ x &= \frac{x*}{\|x*\|}, \\ \cos(\theta_j, i) &= {W_j}^T x_i, \end{aligned} \tag{5}$$

where N is the number of training samples, $x_i \in R^d$ represents the feature vector of the i-th sample, and its identity label is y. $W_j \in R^d$ is the weight of the class j. θ is the angle between W_j and x_i, S is the scaling factor, and m is the margin of the angle, used to limit the distance between classes. The large margin cosine loss function effectively solves the problem of insufficient mask changes in the model training process; all masked-face images are used for model training. Figure 3 shows its representation in the feature space; α and β, respectively, represent the angles between sample x and W1 and W2. For each sample x belonging to class X, $\cos\beta - \cos\alpha \geq m$ should be satisfied, and the margin m also increases the inter-class difference while further compressing the intra-class distance. Experiments have also proved that this method greatly improves the recognition accuracy.

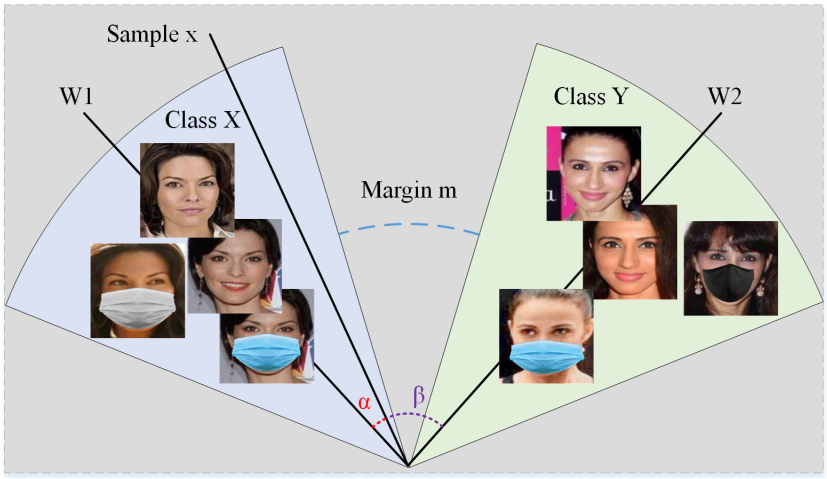

Figure 3. A representation of Cosface in feature space. α represents the angle between x and W1; β represents the angle between x and W2; m is the margin of the angle. The cosine angle and margin effectively limit the inter-class difference while further compressing the intra-class distance.

3.3. Attention Model

According to the characteristics of masked-face images, it can be known that most of the features are unavailable after the mask is put on [45]. If the model still focuses on the features of the global image, those effective features will be ignored. By adding a convolutional block attention module (CBAM) attention mechanism to the network, the model can focus on those truly effective image features, that is, the features of the areas that are not covered by the mask.

The convolutional block attention module is an attention mechanism based on convolutional neural networks [38] which mainly includes a channel attention module and a spatial attention module. By calculating the feature information of these two modules, an attention mapping is generated; then the attention mapping and the feature mapping are multiplied element-wise to obtain the output features. For the output $F \in R^{C\times H\times W}$ of any convolutional layer, CBAM could generate a 1-dimensional channel attention mapping $M_C \in R^{C\times 1\times 1}$ and a 2-dimensional spatial attention mapping $M_S \in R^{1\times H\times W}$, as in Equations (6) and (7), where $\otimes$ is the element-wise multiplication and F'' is the final output feature mapping.

$$F' = M_C(F) \otimes F \tag{6}$$

$$F'' = M_S(F') \otimes F' \tag{7}$$

The channel attention module performs average-pooling and max-pooling on the input feature map to obtain the average-pooling feature F^C_{AVG} and the max-pooling feature F^C_{MAX}. These two features are input into a shared multi-layer perceptron (Shared MLP) to perform the channel information aggregation to generate channel attention mapping $M_C \in R^{C\times 1\times 1}$, as shown in Equation (8), where σ is the sigmoid activation function. Figure 4 shows the process of channel attention mapping.

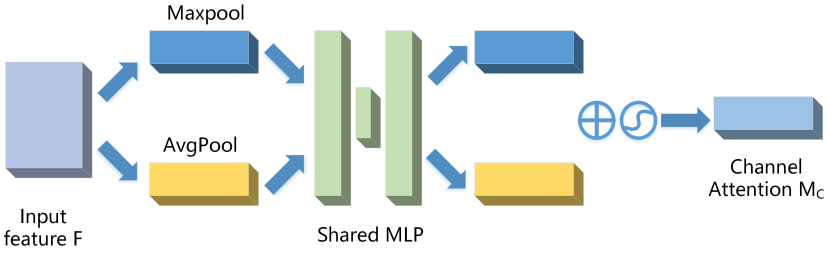

Figure 4. A channel attention module.

$$M_C(F) = \sigma(MLP(AvgPool(F) + MLP(\text{MaxPool}(F))) \quad (8)$$

The spatial attention module multiplies the channel attention mapping $M_C(F)$ as input, and performs channel-based average-pooling and max-pooling on it to obtain the average-pooling feature F^S_{AVG} and the max-pooling pooling feature F^S_{MAX}. The obtained features are concatenated and convolved to obtain a feature mapping of dimension 1, and the final feature mapping is obtained after sigmoid function activation, as shown in Equation (9), which $f^{7\times7}$ represents a convolution operation with the filter size of 7×7 , represented by Conv layers in the Figure 5. Figure 5 shows the process of spatial attention module mapping.

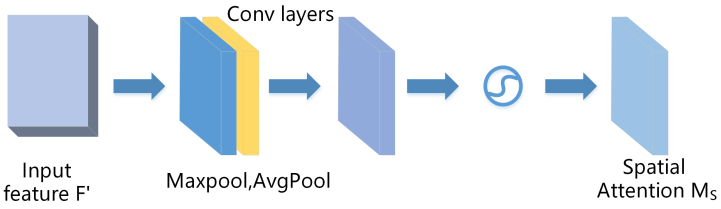

Figure 5. A spatial attention module.

$$M_S(F') = \sigma\left(f^{7\times7}\left(AvgPool(F'); MaxPool(F')\right)\right) \quad (9)$$

3.4. Network Structure

We used Inception-ResNet-v1 as the basic network. The network structure is shown in Figure 6, and for the specific structure one can refer to [46]. Inception-ResNet-v1 is mainly composed of a reduction module and an Inception-ResNet module. The reduction module uses a parallel structure to extract the features while reducing the size of the feature map. The Inception-ResNet module replaces the pooling operation via the residual connection, and cancels the size transformation of the feature map. The model refers to the idea of multi-scale methods, uses convolution kernels of different sizes to increase the receptive field of the model, and fuses features from multiple scales.

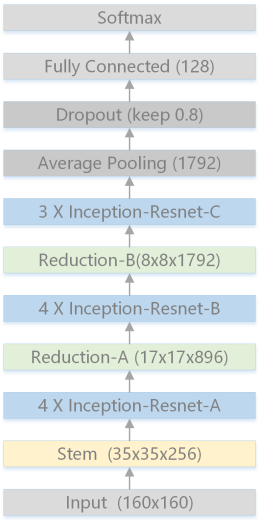

Figure 6. Inception-ResNet-v1 network structure diagram.

We used the Att-Inception module to replace the Inception-ResNet module in the network. The Att-Inception module integrates the CBAM attention mechanism on the basis of the Inception-ResNet module, which can make the model focus more on the effective features of the image. Figure 7 is the model structure diagram of our model, in which the modules of the same size output the same dimension feature map.

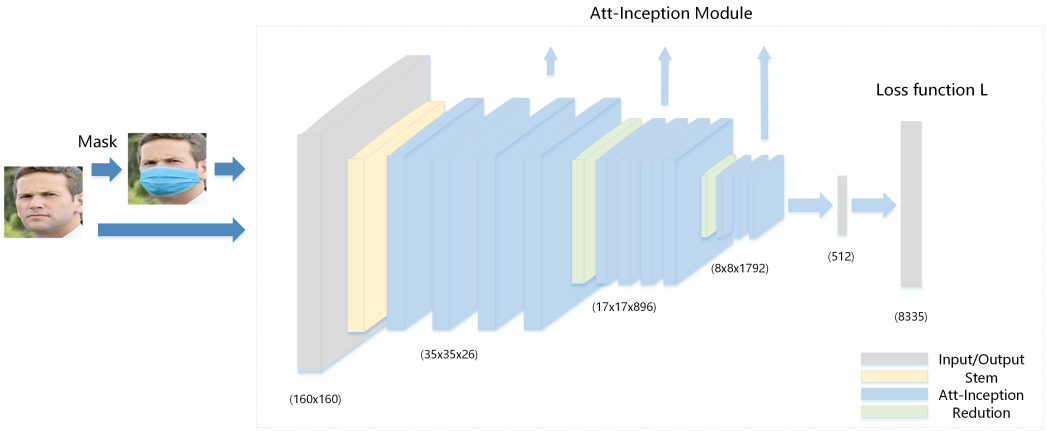

Figure 7. MFCosface network structure diagram. The color of the module corresponds to the structure in Figure 6. After the face is covered with a mask, through the detection of the key facial features we proposed, the image is sent to a model composed of Att-inception modules that incorporate the attention mechanism. Training the model through large margin cosine loss L can increase the intra-class differences while reducing the inter-class differences.

4. Experimental Results

4.1. Datasets and Evaluation Criteria

We conducted experiments on five face datasets, VGGFace2_m, LFW_m, CASIA-FaceV5_m, MFR2, and RMFD. VGG-Face2_m was used for model training, and the remaining datasets were used for testing. We divided the datasets into two types: generated masked-face datasets and real masked-face datasets. VGGFace2_m, LFW_m, and CASIA-FaceV5_m are generated masked-face datasets, including the original images and the masked-face images generated by our method; MFR2 and RMFD are the real masked-face datasets. Examples of the datasets are shown in Figure 8.

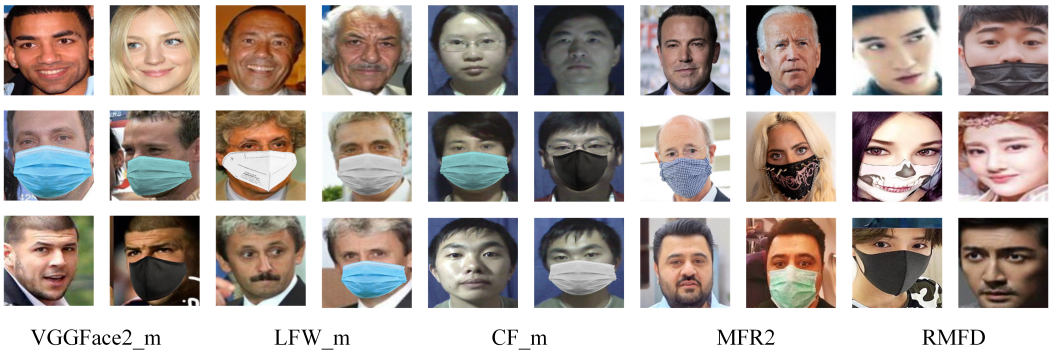

Figure 8. Dataset examples. From left to right are examples of datasets VGGFace2_m, LFW_m, CF_m, MFR2, and RMFD. Each dataset has two columns of images.

VGG-Face2_m was generated from the VGG-Face2 face dataset. VGG-Face2 contains a large number of scenes, lighting settings, and ethnicities. It contains about 3.3 million pictures and 9131 identities. We used 8335 identities in the VGG-Face2 training set and randomly selected 40 pictures from each identity to form VGGFace2_mini. Note that we only used one-tenth of the original dataset. We used our method to generate the corresponding masked-face dataset, and mixed it with VGGFace2_mini to form VGGFace2_m.

LFW_m was generated with the LFW dataset. LFW is currently the most commonly used dataset for facial recognition, having a total of 13,233 face images and 5749 identities. The face images are all photos from real life scenarios, which have high test difficulty. On the LFW dataset we used the same method to generate masked-face images and mixed them with the original images to generate the LFW_m dataset.

CASIA-FaceV5_m was generated from the CASIA-FaceV5 dataset, referred to as CF_m. We use the same method to generate CF_m as the test set. CASIA- FaceV5 contains images of 500 people, and 5 for each person—2500 images in total. All pictures are of Asian faces.

MFR2 is a small dataset containing 53 celebrity and politician identities. The dataset contains unmasked and masked images. There are a total of 269 images, and each identity has an average of about five pictures. This dataset contains more than just the common surgical mask and KN95 mask. It also contains masks with strange patterns.

The RMFD dataset was collected and created by scholars of Wuhan University during the COVID-19 pandemic. It contains 525 objects, 90,000 unmasked-face images, and 2000 masked-face images. As it relies on the network to collect images, the dataset contains a large number of identity errors, duplicate images, and images in which it is too blurry to identify anyone's identity. We manually cleaned the dataset and used the 85,000 pictures obtained after cleaning as the test data.

The experiment in this paper adopted the LFW dataset test method. Except for MFR2, which only used 400 pairs of images due to the small number of images, the other test sets randomly selected 6000 pairs of images as the test data, of which 3000 pairs had the same identities and the other 3000 pairs were from different people. Judging whether these images were of the same person or different people is the recognition result. We used the 10-fold cross-validation method to test the model, divided the test data into 10 randomly, selected 9 of the portions in turn as the training data, and used the rest as the test data. We repeated the training 10 times, and used the average of the 10 test results as the recognition accuracy.

4.2. Experimental Configuration

The experimental platform operating system was Ubuntu 18.04.2, the GPU was a single Tesla V100 with 32 GB memory, and we set the batch size to 90. Through experiments, it was found that the model basically converged at 150 iterations, so we set the number of iterations to 200, all experiments did not use the pre-training model. The input image size was 160×160 and the input data was standardized. The output feature vector dimensions was 512; the dropout parameter was set to 0.4. Both training data and test data used random flip to prevent overfitting. We used common surgical masks (blue, white, green), KN95 masksm and black masks, a total of five types of masks, as mask templates for the experiments, as shown in Figure 8.

4.3. Experimental Results

We tested the model on the generated datasets and the real datasets. For its training set, only our method used the VGGFace2_m dataset; and other methods used VGG-Face2_mini as their training set. The test results of the generated dataset are shown in Table 1. It can be seen that our method greatly improved on the generated datasets compared with the original method FaceNet, and the accuracy rate on the LFW_m dataset reached 99.33%. Since most of the data in the VGGFace2 dataset are European and American faces, the test accuracy on the Asian dataset CF_m was not as good as that in the LFW_m dataset, at only 97.03%.

Table 1. Recognition accuracies on generated datasets.

Method	Training Set	LFW_m	CF_m
Facenet	VGGFace2_mini	83.40	66.07
Softmax	VGGFace2_mini	90.92	71.72
Cosface	VGGFace2_mini	96.82	87.75
Arcface	VGGFace2_mini	97.30	89.62
MFCosface	VGGFace2_m	**99.33**	**97.03**

The test results of the real dataset are shown in Table 2. Our method increased the accuracy of the original network Facenet from 84.25% to 98.50%. Since our method has a

great advantage in masked-face image recognition, and RMFD only contains 5% masked-face images, it did not achieve the best performance in this dataset. However, our method was only 0.13% worse than the most advanced method Arcface. It can be seen that our method was also greatly improved on the real masked-face dataset. Experimental results shows that our method has better recognition performance than other methods, and it can complete the facial recognition task in the mask occlusion state.

Table 2. Recognition accuracies on real datasets.

Method	Training Set	MFR2	RMFD
Facenet	VGGFace2_mini	84.25	73.77
Softmax	VGGFace2_mini	92.50	90.30
Cosface	VGGFace2_mini	92.75	92.03
Arcface	VGGFace2_mini	93.11	**92.28**
MFCosface	VGGFace2_m	**98.50**	92.15

4.4. Ablation Experiment

In order to prove the effectiveness of MFCosface, ablation experiments were performed on the generated dataset LFW_m and the real dataset MFR2. According to the experimental results of LFW_m in Table 3, the corresponding ROC (receiver operating characteristic curve) was drawn, as shown in Figure 9. The ROC graph uses the false positive rate and the true positive rate as the coordinate axes, reflecting the relationship between them, and can better represent the model's recognition ability. AUC (area under the curve) is the area under the ROC curve. The higher the AUC value, the stronger the classification ability of the model. It can be seen from the experimental results that the accuracy was improved regardless of whether a single method or a combination of multiple methods was used. The highest accuracy was obtained when the three methods were used at the same time. In terms of ROC, the MFCosface method was also significantly better than the other methods, and the addition of each method improved the performance of the model.

It is worth noting that in our Cosface + CBAM experiment, the combination of these two methods was not as good as Cosface. This is because the training dataset did not contain masked-face images. At this time, the attention mechanism made the model focus on the entire face, not areas that would not be occluded by the mask. This had an adverse effect; that is, when the face is occluded by the mask, the extracted facial features are severely obscured, resulting in a decrease in the recognition accuracy. The masked method solves the problem of insufficient data, the Cosface method optimizes the distribution of the feature space based on the data generated by the masked method, and CBAM makes the model more capable of learning useful features and is committed to solving hard samples. These three methods complement each other, making the model have stronger masked facial recognition capabilities.

Table 3. Results of ablation experiments.

Method	LFW_m	MFR2
Facenet	83.40	84.25
Masked	93.58	89.75
Cosface	96.82	96.75
Cosface + CBAM	96.70	96.25
Masked + Cosface	99.23	97.00
MFCosface	**99.33**	**98.50**

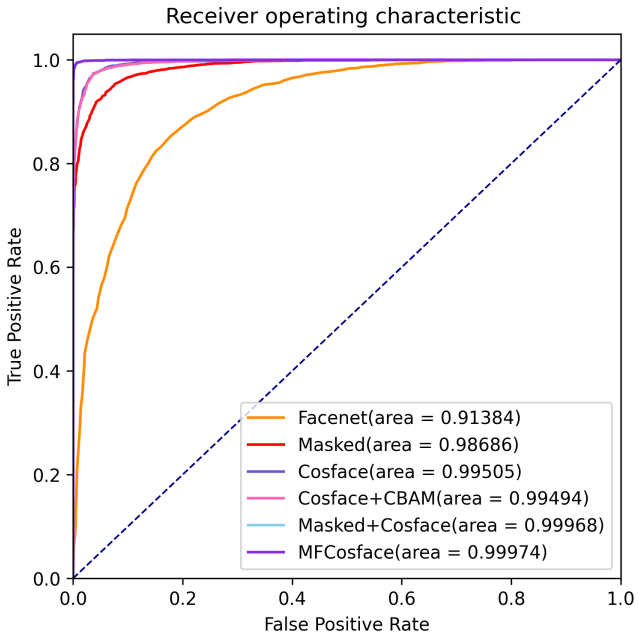

Figure 9. The receiver operating characteristic curve. Facenet represents the original network, masked represents the use of images generated by our method for training, Cosface represents the use of large margin cosine loss as the loss function, CBAM means training with an attention mechanism, and MFCosface represents the method we proposed.

4.5. Noise Experiment

The key to solving the problem of masked-face recognition is to ignore the invalid features of the mask occlusion area and pay more attention to the effective features. In order to prove that our method pays more attention to the unoccluded facial area (the upper part of the face) during feature extraction, a noise experiment was designed. First, each entire image was divided into an upper part and a lower part according to the key points at the bridge of the nose; and then salt and pepper noise, Gaussian noise, and random noise were added for recognition. The results are shown in Table 4. The noise part "Up" represents the upper area of the face. "Down" represents the lower area of the face. "All" represents adding noise to the entire image.

The addition of noise destroyed the facial feature information and caused a decrease in accuracy to a certain extent. In the experiment, MFCosface was compared with the original method FaceNet. It can be seen that on different datasets with different noises, our methods were much better than FaceNet and showed strong robustness. The recognition accuracy after adding noise to the lower half of the face was high, which shows that our method can still extract reliable facial features for recognition after the features in the lower half are destroyed. This proves that our method pays more attention to the facial features in the upper half of the area, and is less dependent on the facial features in the lower half. In summary, our method pays more attention to the upper half of the face that is not covered by the mask, and has strong robustness in the face of noise.

Table 4. Recognition accuracy rate in the noise experiment.

Location	Dataset	Noise Type	Example	Method	Accuracy
UP	LFW	salt and pepper noise (probability = 5%)		FaceNet	84.55
				MFCosface	**97.00**
Down	LFW	salt and pepper noise (probability = 5%)		FaceNet	82.05
				MFCosface	**97.92**
All	LFW	salt and pepper noise (probability = 5%)		FaceNet	77.45
				MFCosface	**95.12**
UP	CASIA-FaceV5	Gaussian noise (Mean = 0, variance = 1)		FaceNet	75.71
				MFCosface	**93.62**
Down	CASIA-FaceV5	Gaussian noise (Mean = 0, variance = 1)		FaceNet	75.58
				MFCosface	**96.47**
All	CASIA-FaceV5	Gaussian noise (Mean = 0, variance = 1)		FaceNet	74.21
				MFCosface	**92.10**
UP	MFR2	random noise (num = 1000)		FaceNet	76.75
				MFCosface	**94.50**
Down	MFR2	random noise (num = 1000)		FaceNet	84.50
				MFCosface	**97.50**
All	MFR2	random noise (num = 1000)		FaceNet	79.25
				MFCosface	**94.50**

4.6. Attention Mechanism Experiment

The basic network Inception-ResNet-v1 is mainly composed of the Reduction module and the Inception-ResNet module. We set up the attention mechanism experiment based on different modules—that is, we added the CBAM attention mechanism to different modules. Table 5 shows the experimental results, where CBAM_Reduction represents CBAM being added to the Reduction module. The network structure is shown in Figure 10a (take the Reduction-A module as an example). V represents the padding mode as valid; Att-Inception means that CBAM was added to the Inception-ResNet module. The network structure is shown in the Figure 10b (take the Inception-ResNet-A module as an example). CBAM_All is where CBAM was added to all modules.

Table 5. Results of the attention mechanism experiment.

Method	LFW_m	CF_m	MFR2
CBAM_Reduction	99.08	96.78	98.00
CBAM_All	99.18	**97.17**	98.00
Att-Inception	**99.33**	97.03	**98.50**

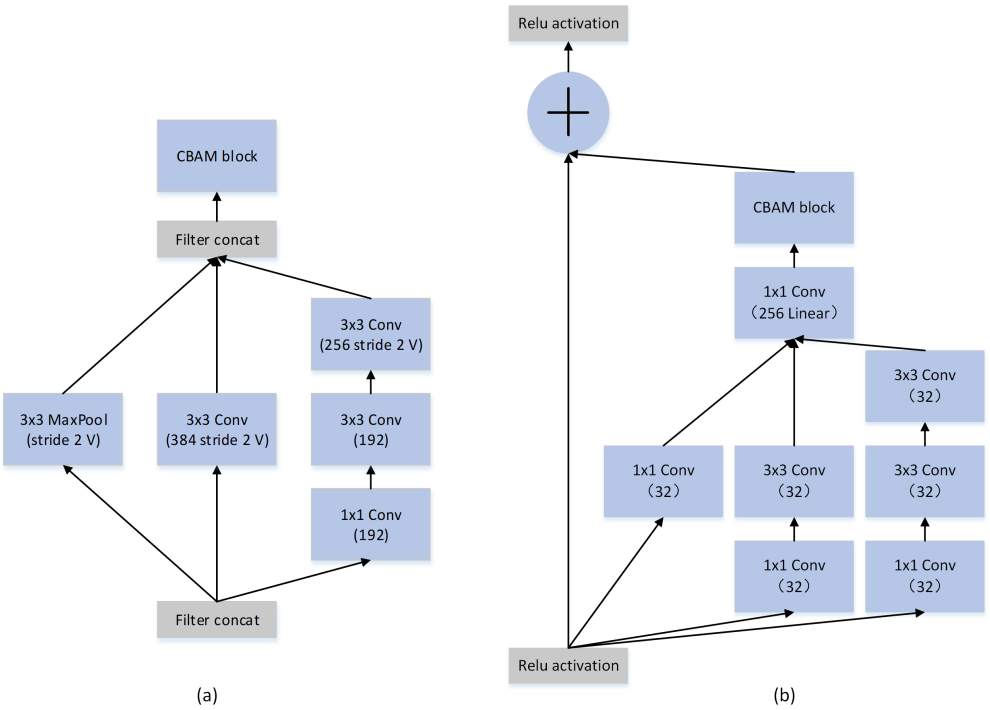

Figure 10. (**a**) Reduction-A module with the attention mechanism; (**b**) Inception-ResNet-A module with the attention mechanism.

According to the experimental results, the Att-Inception module constructed in this paper is significantly better than the other two modules. On the CF_m dataset, our recognition accuracy was not the highest, but it was only 0.14% worse than the highest—a relatively high performance. Experiments on several datasets proved that our method performs better in most cases and achieves higher recognition accuracy.

4.7. Real-World Experiment

In order to verify the accuracy of our method in masked-face recognition in real situations, we collected masked-face images from 137 students in the laboratory to form a dataset for testing. We took five unmasked images of each subject and five images of him/her while wearing a mask. The dataset contains changes in expression, light, angle, etc. We used five unmasked-face images and five corresponding images generated by our method as the training set, and the original dataset as the test set. The recognition accuracy was 98.54%. At the same time, we used the camera to collect data for recognition. The recognition result is shown in Figure 11. It can be seen from the recognition results that our method can recognize the identity of the person wearing a mask in a real situation, showing facial recognition ability.

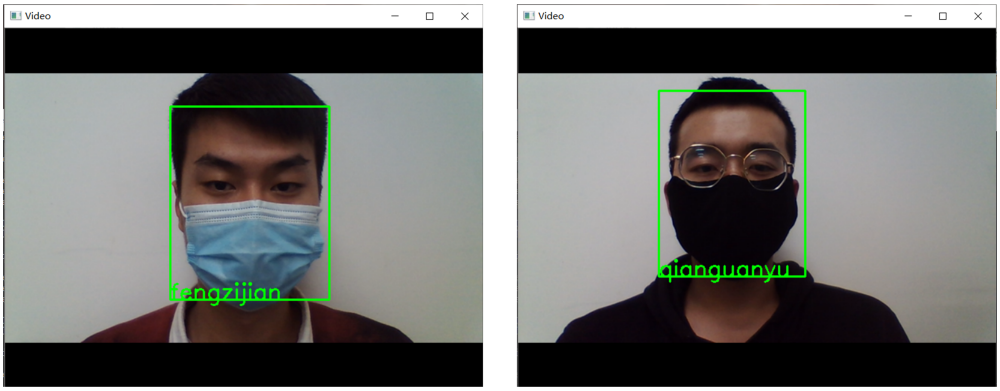

Figure 11. Real-world results.

5. Conclusions

In this paper, we proposed a masked-face recognition algorithm based on large margin cosine loss, which has high recognition accuracy. To address the problem of insufficient masked-face images, we used the detection of key facial features to cover face images with common mask templates to generate corresponding datasets. Through the analysis of the masked-face dataset, we found that triplet loss is not applicable to our dataset, and we used large margin cosine loss to train the model. Since the mask destroys some of the facial feature information, we added an attention mechanism to make the model focus on effective regions to extract more important feature information. Through experiments on generated masked-face datasets and real masked-face image datasets, it was proven that our method is superior to the other existing methods. Finally, a real-world experiment was undertaken that simulated a real situation, and the results show that our method performs masked-face recognition with high accuracy.

In the future, we will explore how to combine semantic information to generate more realistic masked-face images to solve the problem of insufficient data. Like most algorithms, our method suffers from performance degradation when encountering extreme posture and expression changes, and we will focus on solving this problem in future research. Our method can also be extended to relatively regular occlusion objects, such as sunglasses and scarves.

Author Contributions: Conceptualization, H.D. and Z.F.; methodology, Z.F.; software, G.Q.; validation, H.D., Z.F., and X.L.; formal analysis, G.Q.; investigation, X.L.; resources, H.D., H.L. and G.L.; data curation, Z.F.; writing—original draft preparation, Z.F.; writing—review and editing, H.D. and G.L.; visualization, Z.F.; supervision, H.D.; project administration, H.L.; funding acquisition, H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China grant number 61976150; and by the Natural Science Foundation of Shanxi Province, China grant numbers 201801D121135 and 201901D111091.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets is available at the following link: https://github.com/emuliey/mask_face_datasets (accessed on 6 August 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, J.; Su, G.; Xiong, Y.; Chen, J.; Shang, Y.; Liu, J.; Ren, X. Sparse representation for based on constraint sampling and face alignment. *Tsinghua Sci. Technol.* **2013**, *18*, 62.
2. Shenvi, D.R.; Shet, K. Cnn based covid-19 prevention system. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; pp. 873–878.
3. Yin, X.; Yu, X.; Sohn, K.; Liu, X.; Chandraker, M. Feature transfer learning for face recognition with under-represented data. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 5697–5706.
4. Hu, G.; Yang, Y.; Yi, D.; Kittler, J.; Li, S.; Hospedales, T. When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshop, Santiago, Chile, 7–13 December 2015; pp. 384–392.
5. Oloyede, M.O.; Hancke, G.P.; Myburgh, H.C. A review on face recognition systems: Recent approaches and challenges. *Multimed. Tools Appl.* **2020**, *79*, 37–38.
6. Sun, Y.; Liang, D.; Wang, X.; Tang, X. DeepID3: Face Recognition with Very Deep Neural Networks. *arXiv* **2015**, arXiv:1502.00873.
7. Deng, H.; Feng, Z.; Liu, Y.; Luo, D.; Yang, X.; Li, H. Face recognition algorithm based on weighted intensity pcnn. In Proceedings of the 2020 Eighth International Conference on Advanced Cloud and Big Data (CBD), Taiyuan, China, 5–6 December 2020; pp. 207–212.
8. Yang, M.; Zhang, L.; Yang, J.; Zhang, D. Robust sparse coding for face recognition. In Proceedings of the Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011; pp. 625–632.

9. Wen, Y.; Liu, W.; Yang, M.; Fu, Y.; Xiang, Y.; Hu, R. Structured occlusion coding for robust face recognition. *Neurocomputing* **2016**, *178*, 11–24.
10. Wu, C.; Ding, J. Occluded face recognition using low-rank regression with generalized gradient direction. *arXiv* **2019**, arXiv:1906.02429.
11. Dong, J.; Zheng, H.; Lian, L. Low-rank laplacian-uniform mixed model for robust face recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 11889–11898.
12. Shi, Y.; Yu, X.; Sohn, K.; Chandraker, M.; Jain, A. Towards universal representation learning for deep face recognition. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 6816–6825.
13. Wang, X.; Han, C.; Hu, X. Densely connected convolutional networks face recognition algorithm based on weighted feature fusion. *J. Front. Comput. Sci. Technol.* **2019**, *13*, 1195–1205.
14. Song, L.; Gong, D.; Li, Z.; Liu, C.; Liu, W. Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 773–782.
15. Cen, F.; Wang, G. Dictionary representation of deep features for occlusion-robust face recognition. *IEEE Access* **2019**, *7*, 26595–26605.
16. Du, H.; Shi, H.; Liu, Y.; Zeng, D.; Mei, T. Towards nir-vis masked face recognition. *IEEE Signal Process. Lett.* **2021**, *28*, 768–772.
17. Wang, Z.; Kim, T.S. Learning to recognize masked faces by data synthesis. In Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Jeju Island, Korea, 13–16 April 2021; pp. 36–41.
18. Amin, M.I.; Hafeez, M.A.; Touseef, R.; Awais, Q. Person identification with masked face and thumb images under pandemic of covid-19. In Proceedings of the 2021 7th International Conference on Control, Instrumentation and Automation (ICCIA), Tabriz, Iran, 23–24 February 2021; pp. 1–4.
19. Xie, J.; Yang, J.; Qian, J.; Tai, Y.; Zhang, H. Robust nuclear norm-based matrix regression with applications to robust face recognition. *IEEE Trans. Image Process.* **2017**, *26*, 2286–2295.
20. Ejaz, M.; Islam, M.; Sifatullah, M.; Sarker, A. Implementation of principal component analysis on masked and non-masked face recognition. In Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), Dhaka, Bangladesh, 3–5 May 2019; pp. 1–5.
21. Xu, X.-F.; Zhang, L.; Duan, C.-d.; Lu, Y. Research on inception module incorporated siamese convolutional neural networks to realize face recognition. *IEEE Access* **2019**, *8*, 12168–12178.
22. Xin, J.; Tan, X. Face alignment in-the-wild: A Survey. *Comput. Vis. Image Underst.* **2017**, *162*, 1–22.
23. Georgios, T.; Maja, P. Optimization Problems for Fast AAM Fitting in-the-Wild. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 593–600.
24. Epameinondas, A.; Joan, A.; Stefanos, Z. Active Pictorial Structures. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5435–5444.
25. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 4.
26. Wu, W.; Qian, C.; Yang, S.; Wang, Q.; Cai, Y.; Zhou, Q. Look at Boundary: A Boundary-Aware Face Alignment Algorithm. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2129–2138.
27. Kowalski, M.; Naruniec, J.; Trzcinski, T. Deep alignment network: A convolutional neural network for robust face alignment. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 2034–2043.
28. Huang, Z.; Zhang, J.; Shan, H. When age-invariant face recognition meets face age synthesis: A multi-task learning framework. *arXiv* **2021**, arXiv:2103.01520.
29. Zhong, Y.; Deng, W.; Wang, M.; Hu, J.; Peng, J.; Tao, X.; Huang, Y. Unequal-training for deep face recognition with long-tailed noisy data. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7804–7813.
30. Wen, Y.; Zhang, K.; Li, Z.; Qiao, Y. A discriminative feature learning approach for deep face recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–12 October 2016; pp. 499–515.
31. Liu, W.; Wen, Y.; Yu, Z.; Yang, M. Large-margin softmax loss for convolutional neural networks. *arXiv* **2016**, arXiv:1612.02295.
32. Liu, W.; Wen, Y.; Yu, Z.; Li, M.; Raj, B.; Song, L. Sphereface: Deep hypersphere embedding for face recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
33. Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; Liu, W. Cosface: Large margin cosine loss for deep face recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5265–5274.
34. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4685–4694.

35. Tay, C.-P.; Roy, S.; Yap, K.-H. Aanet: Attribute attention network for person re-identifications. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7127–7136.
36. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3141–3149.
37. Gao, Z.; Xie, J.; Wang, Q.; Li, P. Global second-order pooling convolutional networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 3019–3028.
38. Woo, S.; Park, J.; Lee, J.; Kweon, I.S. Cbam: Convolutional block attention module. In *Computer Vision–ECCV 2018*; Springer International Publishing: Munich, Germany, 8–14 September 2018; pp. 3–19.
39. Wang, X.; Girshick, R.; Mulam, H.; He, K. Non-local neural networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
40. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
41. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. Eca-net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
42. Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014.
43. Anwar, A.; Raychowdhury, A. Masked face recognition for secure authentication. *arXiv* **2020**, arXiv:2008.11104.
44. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
45. Damer, N.; Grebe, J.H.; Chen, C.; Boutros, F.; Kirchbuchner, F.; Kuijper, A. The effect of wearing a mask on face recognition performance: An exploratory study. *arXiv* **2020**, arXiv:2007.13521.
46. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 4278–4284.