ULN: An efficient face recognition method for person wearing a mask

Zijun Zhuang1 · Hongtao Lu1

Received: 1 April 2021 / Revised: 21 August 2021 / Accepted: 13 July 2022 / Published online: 12 August 2022 © The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Although the face recognition has advanced by leaps and bounds in recent years, recognizing faces with large occlusion, e.g., masks, is still a challenging problem. In the context of the COVID-19 outbreak, wearing masks becomes mandatory, which fails numerous face attendance and surveillance systems. Therefore, a robust face recognition algorithm that can deal with facial masks is urgently needed. To build a mask-robust face recognition algorithm, we first generate numerous facial images with masks based on public face datasets, which obviously alleviates the problem of the training data shortage. Second, we propose a novel network architecture called Upper-Lower Network (ULN) to recognize the faces with masks efficiently. The upper branch of ULN with the mask-free images as input is pretrained that provides supervisory information for the training of the lower branch. Considering that the occlusion areas of masks usually appear in the lower parts of faces, we further divide the high-order semantic features into upper and lower parts. The designed loss function force the learned features of the lower branch similar to those of the upper branch with the same mask-free image inputs, but only the upper part of features similar to the mask counterparts. Extensive experiments demonstrate that the proposed method is effective for recognizing persons with masks and outperforms other state-of-the-art face recognition methods.

Keywords Face recognition · Facial mask · COVID-19 · Occlusion · Security surveillance

Zijun Zhuang
zhuangzijun@sjtu.edu.cn

Hongtao Lu
htlu@sjtu.edu.cn

1 Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China
1 Introduction

Face recognition is the most commonly used biometric authentication in daily life. When faced with different specific problems, face recognition is generally divided into two categories, face identification which classifies a given face to a known identity and face verification which determines whether a pair of faces belongs to the same identity or gives a ‘similarity’ between two faces. Since GaussianFace [21] first surpassed the human-level on the LFW [15] dataset, face recognition becomes one of the most promising topics in computer vision and artificial intelligence. Thanks to the development of deep Convolutional Neural Networks (CNNs), there has been considerable progress in the field of face recognition.

Coronavirus disease 2019 (COVID-19) is a respiratory disease caused by a new strain of coronavirus. By the end of Nov 9, 2020, the virus has already infected more than 50 million patients, and more than 1.2 million died from the virus [9]. There is clear evidence shows wearing masks can impede virus transmission [13, 22, 28]. Meanwhile, more and more national health departments worldwide issued regulations requiring residents to wear masks in public to ensure safety. On the contrary, wearing masks does bring many challenges to the face recognition algorithm used now, which brings a safety hazard. When the surveillance systems cannot handle the person wearing a mask (PWM) problem, the evildoers would mingle them in the crowd without being found. Thus, a reliable face recognition system that works well on PWM is an urgent need.

Although deep learning models have achieved great success in general face recognition scenarios, existing face recognition methods for PWM have the following two main shortcomings. The first one lacks of large-scale datasets with people wearing masks. The AR Face Database [23] provides occluded faces, but there are only about 4,000 photos from 126 people, which is insufficient for a large deep neural net training process. We know that how critical the data is for deep learning. Without the support of good quality training data, the optimization of the algorithm will be very hard. The second one is that the current methods cannot work well on images with large occlusion areas. ArcFace [7] is recently proposed and becomes one of the representative face recognition methods. Arcface uses the angular margin method to impose an additional penalty on the logits of the training sample classes, leading to better intra-class compactness. Although ArcFace provides a much more robust feature than other methods, it still can not handle the faces with large occlusion areas well enough. Song et al. proposed an occlusion robust method based on predicting the occluded area of the photo [30] which is inspirational and outperforms other methods on AR dataset [23] with natural occlusions. The method requires additional computation on the occluded part both in the training phase and testing phase which is very time-consuming if we need the algorithm to recognize a person from a relatively big database in real-time.

Figure 1 illustrates three different people wearing different masks, but the occluded part by the masks seems similar. From this observation, we can suppose the mask always covers the same part of the face. Most of the valuable information for face recognition is concentrated in the upper part of the face, such as eyes, eyebrows, and forehead. The covered part of the face gives some of the face’s ambiguous information, such as the face shape and height of the nose bridge. Inspired by this observation, we develop a method to synthesize a photo of a person wearing a mask from a portrait photo and a mask photo in the real world. Landmarks of the face are located so that the mask photo could be put to the proper area of the face. This method can be easily performed on existing face recognition datasets such as LFW [15], CASIA-WebFace [41], VggFace2 [3], and the generated photos can be directly
used as the training data to improve the model performance on the PWM task. Moreover, we proposed a novel upper-lower network (ULN) structure to overcome the difficulties of the PWM task and still keeps high performance on the normal face recognition situation in the meantime. The face feature is divided into the upper and lower parts. The upper part will play a more critical role when the photo is from a person wearing a mask, and the lower part will guarantee the accuracy of the recognition on faces without a mask.

We summarize our main contributions as follows:

(1) We propose a novel data generation method. The method generates numerous facial images with masks based on public face recognition datasets and efficiently alleviates the data shortage problem.

(2) We propose an upper-lower network (ULN) for the PWM task. The upper network of ULN is pretrained to provide supervisory information for the lower one, while the lower one is trained to learn similar features with the upper one for the uncovered cover parts of faces.

(3) Extensive experiments on LFW, MegaFace, MFR2, and AR datasets demonstrate the proposed method’s efficiency and effectiveness.

2 Related work

In order to improve the accuracy and robustness of the face recognition model, different scholars have done much research from different perspectives.

SOTA Face Recognition Methods Recently, many face recognition methods focus on improving the discriminability of the feature representation. These methods get more robust models, and the performance improves on challenging datasets e.g. Megaface [17] every year. Liu et al. proposed L-Softmax [20], which is the first work that introduces the idea of large angular margin into the softmax loss. After that, Liu et al. proposed SphereFace [19], which is an improved version of L-Softmax. SphereFace normalizes the weights of the fully connected layer to make the training more focused on optimizing the angle of the deep feature map. Wang et al. proposed CosFace [36], which uses an additive angular margin other than the multiplicative margin used in the SphereFace, and its performance surpassed
SphereFace. Deng et al. proposed ArcFace [7]. The idea is similar to SphereFace and CosFace but outperforms them. Regularization-based learning methods [5, 6] are always used to obtain more discriminative and more robust features. Zhao et al. proposed RegularFace [44], which imposes an exclusive regularization on the weight and explicitly enlarges the angular distance between different identities. UniformFace [10] has very similar ideas. These methods can be combined with classification-based methods such as CosFace and ArcFace and improve their performance.

Occlusion Robust Face Recognition Methods Some methods are focused on eliminating the bad effect of the occlusion in the photo. Yin et al. proposed SRC [42] and achieved significant improvement on the occlusion face recognition. The sparse linear representation of the image in the training set is learned to reconstruct the testing image. The sparse error term is added to compensate for the linear representation error of the image and then do the classification. Song et al. proposed an occlusion robust method based on predicting the occluded area of the photo [30]. They first learn a pairwise differential siamese network (PDSN) from occlusion-free photo and photo add artificial occlusion pairs to determine the correspondence between occluded blocks and useless feature units. Next, a mask dictionary is learned from the previous mask generators, and then the item in the dictionary (FDM) eliminates the feature elements that are affected by the occluded facial blocks. Ding et al. proposed latent part detection (LPD) model [8], which can locate the latent facial part that robust to the occlusion of the facial masks.

Data Preprocessing Methods Masi et al. [24] use 3D models to generate face photos from different angles and expressions. They prove that well-designed augmented data can greatly improve the performance of the model and even be as useful as real-world data. Trigueros et al. [33] proposed using synthetic occluded faces in a strategic way to augment the training data and observed performance improvements in the test. With the development of the Generative Adversarial Networks (GAN), it becomes a popular method to augment the data [1]. People began to use the face generated by the network to train the face recognition model. Fu et al. [11], the new face image data generation method can produce 200,000 face virtual images that do not exist in the real world from scratch. This method can effectively alleviate the high cost of data collection in heterogeneous face recognition and make full use of a small number of authentic images for deep learning. Researchers use these generated realistic virtual images and achieve significant improvement in recognition performance in a series of challenging face recognition applications such as near-infrared-visible light, thermal infrared-visible light, sketch-photo, profile-front face, ID card-camera photo.

3 PWM data generating

As shown in Fig. 1, the masks always cover the same part of the face despite the gender of the wearer or the type of the mask. This observation can be easily associated with the facial landmarks detection problem, whose job is to locate the key points on the face precisely. We can overlay the facial mask image on top of a mask-free photo with the help of the facial landmarks. The face with mask data generation in three steps: 1) Grab the facial mask template from the real-world photos. 2) Determine the proper area where the ‘mask’ should be taken on. 3) Overlay the mask template on the face to keep them away from the virus threats.
Step 1. We use the photos of the person wearing a mask to generate the mask template. The PWM photos were collected from the search engine results of the keyword ‘facial mask’ by our spider. We manually choose 100 of them to be used as the resources for generating the mask template. Figure 2(a) is an illustration of the photos of a person wearing a mask we grab from the internet. Dlib [18] library is used to determine the landmarks of the face in our algorithm. It generates 68 key points of the face, including the edge of the face, eyes, eyebrows, nose, mouse. The landmarks are illustrated in Fig. 2(b). The facial mask area is encircled by the landmarks that represent the nose bridge and jawline below the ears. Figure 2(c) shows the grabbed mask template. The facial mask image and the the position of the keyponits are stored for the further generating process. We totally have 100 mask images from 100 photos in the template.

Step 2. When we get a mask-free image, like Fig. 3(a), we first do the 68 facial landmarks detection just as mentioned above, which is shown in Fig. 3(b). The same area encircled by the landmarks that represent the nose bridge and jawline below the ears is considered to be the proper place to put the facial mask. The black area in Fig. 3(c) indicates the facial mask area.

(a) photo with mask  (b) landmarks of face  
(c) generated mask template

Fig. 2  An illustration of the steps of we generating the mask template from real world images
Step 3. For a mask-free photo, we randomly choose one facial mask image from the template to generate the PWM data. Since the shape of the mask and the shape of the face needed to be put on mask is not same, we need to do a transformation on facial mask so that it can fit the face well. To calculate the affine transformation matrix much easier, we first cut the mask image into triangles by linking the most lower keypoint (the most lower part of chin) to all the other keypoints. For each
triangle in the facial mask image, it should be transformed to the respective triangle in the mask-free image. Suppose the three angles' coordinate position of the facial mask triangle is \( a(a_x, a_y), b(b_x, b_y), c(c_x, c_y) \) and the corresponding position needs to be transformed is \( a'(a'_x, a'_y), b'(b'_x, b'_y), c'(c'_x, c'_y) \). We can have three equations from the transformation:

\[
\begin{align*}
A(a_x, a_y, 1)^T &= (a'_x, a'_y, 0)^T \\
A(b_x, b_y, 1)^T &= (b'_x, b'_y, 0)^T \\
A(c_x, c_y, 1)^T &= (c'_x, c'_y, 0)^T
\end{align*}
\]

where \( A \) is the affine transformation matrix to be calculated,

\[
A = \begin{pmatrix} A_0 & A_1 & A_2 \\ A_3 & A_4 & A_5 \\ 0 & 0 & 1 \end{pmatrix}
\]  

We can use the least square method to calculate the approximate solution of the equation set. After we get the affine transformation matrix \( A \), we can transform the small triangle from the facial mask to the mask-free face image and overlay it on the top of the face image. Figure 3(d) is the generated PWM image from the mask-free image and facial mask template, which has a satisfactory visual quality.

We do the generation on CASIA-WebFace dataset \([41]\) and VggFace2 \([3]\) for further training and LFW dataset \([15]\) for testing. Figure 4 is the samples from the generated PWM dataset. The upper row is the photos augmented from the CASIA-WebFace dataset, and the bottom row is augmented from the LFW dataset.

### 4 Upper-lower network

The upper-lower network (ULN) is introduced to overcome the difficulties of the high occlusion PWM recognition. Both the network architecture and the feature layer use a novel upper-lower structure, making the model work well on both the mask-free photos and PWM photos. Figure 5 provides an overview of the ULN structure and the training framework.
4.1 Upper-lower branches

To solve the problem of the large occlusion area of the PWM task, we proposed the upper-lower network (ULN) structure. ULN is a two-branch network consists of the upper branch and lower branch. It receives two images at the same time. As illustrated in Fig. 5, we send the mask-free image to the upper branch, and we send the same photo as above or the corresponded generated face wearing a mask image to the lower branch randomly. Both of the two images then send to a CNN to extract the feature. Although the upper and lower branches have the same CNN architecture, the upper branch is initialized by the pretrained model, which trains on the mask-free dataset and is fixed during the ULN training process. Only the lower branch is trained in the ULN, and the upper branch works as a teacher of the lower one.

4.2 Upper-lower feature partition

Just as the name of the network ‘upper-lower’, we also divide the features of the network into upper and lower parts. As a result of the receptive field of the convolutional layer, the upper part of the feature grasps most of the information from the upper part of the input image and some high-level semantic information from the pixels far from it, the same as the lower part of the feature. If we receive a mask-free photo, both the upper part of the image, which contains the eyes, brow, etc. and the lower part of the image, which contains nose, mouse, etc. plays an important role in face recognition. However, when we have a PWM photo, the upper part of the image will not change a lot. While, the lower part no longer contains the clear information of mouse and nose, and it only provides some of the ambiguous information of the face such as the face shape, the height of the nose bridge. Clearly, we should not treat the feature come from different parts equally. So, we partitioned the features and divided the feature into the upper half and lower half in the space domain. Then the four features from two branched are connected to four feature vectors by the fully
connected layer. We name the upper feature vector from the upper branch as \( f_{uu} \). Rest of the feature vectors names \( f_{ul}, f_{lu}, f_{ll} \) by analogy. We will use different strategies to treat these four feature vectors.

### 4.3 Selective cosine distance teacher loss

As mentioned above, the upper branch will act as a supervisory signal to the lower branch. We use the selective cosine distance teacher loss to constraint the lower branch. The adjective ‘selective’ in the name of the loss function means the teacher network will teach the students in accordance with their aptitude. When the lower branch receives a mask-free image, both of feature vectors, i.e. \( f_{lu} + f_{ll} \), the element-wise summation of the features from the lower branch will calculate the distance with the upper branch feature vector \( f_{uu} + f_{ul} \). The loss enforces the lower branch learning the similar feature as the pretrained model which performance well on the mask-free dataset. This ensures the model trained on ULN will not bias to the PWM images and lost a lot of accuracy on the mask-free dataset.

When the lower branch receives a PWM image, only the upper part is selected, i.e. \( f_{lu} \) will calculate the distance with the upper branch upper feature \( f_{uu} \). The loss enforces the feature to be consistency of the upper part from both two branches. This will ensures the robustness of the model when handling the PWM images. The loss function can be written as bellow:

\[
L_{SCDT}(x_i^l, x_i^u) = \begin{cases} 
  d((f_{uu}^l + f_{ul}^l), (f_{lu}^l + f_{ll}^l)), & \text{if } x_i^l \text{ mask-free} \\
  d(f_{uu}^l, f_{lu}^l), & \text{if } x_i^l \text{ is PWM image} 
\end{cases}, \quad (3)
\]

where \( d(, ) \) is the cosine distance of two vectors and can be defined as follow:

\[
d(x, y) = 1 - \frac{x \cdot y}{\|x\| \cdot \|y\|}, \quad (4)
\]

and

\[
L_{SCDT} = \sum_i L_{SCDT}(x_i^l, x_i^u). \quad (5)
\]

### 4.4 Recognition loss

The lower branch features will be connected to a classifier formed by a fully connected layer. The output shape is the number of identities to calculate the recognition loss. We suppose the number of identities of the training set is \( N \), the weight of the last fully connected layer can be written as \( W = \{w_1, w_2, \ldots, w_N\} \), where \( w_i \) can be explained as the class center of the \( i \)-th identity. The traditional softmax loss can be presented as follows:

\[
L_{Rec} = \frac{1}{M} \sum_{i=1}^{M} - \log \frac{e^{w_{y_i}^T f_i + b_{y_i}}}{\sum_{j=1}^{N} e^{w_j^T f_i + b_j}}, \quad (6)
\]

where \( f_i = f_{lu}^i + f_{ll}^i \).

The recognition loss can be easily generalized to other more loss functions with better performance, such as ArcFace [23] loss which is actually used in our paper.

Whether the received images are mask-free or not, both the upper and lower part of the feature will be used to calculate the recognition loss. The reason is that the identity information from the recognition loss can directly supervise the model to be effective on mask-free and PWM datasets simultaneously.
4.5 Loss function

The total loss function of ULN is combining the selective cosine distance teacher loss and classification loss and can be written as:

$$L(\theta) = L_{\text{Rec}} + \lambda L_{\text{SCDT}}.$$  (7)

The hyperparameter $\lambda$ is set to be 0.5 in our model.

5 Experiments

5.1 Implementation details

Preprocessing We follow the recent papers [7, 36] to do the preprocessing. All the images in the training set and test set are aligned by MTCNN [43] through the five-point face landmarks. Then the images are cropped into $112 \times 112$. The cropped image will be used to generate the PWM images. Each pixel in both mask-free and PWM images ($112 \times 112 \times 3$) is normalized by subtracting 127.5 then dividing by 128 so that the pixel can be represented in the interval $(-1, 1)$.

Network Setup Pytorch [27] is used as the framework to implement ULN. We have models of three different sizes denoted ULN, ULN-M, and ULN-L. ULN and ULN-M use ResNet [12] with 18 layers as the network architecture. We also use the BN [16]-Dropout [31]-FC-BN after the last convolutional layer, L inputs. IR blocks and SE module [14] is not used if no specific mention. ULN uses 128-d features and ULN-M uses 512-d features. The larger model ULN-L uses ResNet with 50 layers as the network architecture. We also use the BN-Dropout-FC-BN after the last convolutional layer, L inputs. IR blocks and SE module is also used. The feature dimension is 512. Both ULN, ULN-M, and ULN-L models use the ArcFace with the corresponding same architecture trained on CASIA-WebFace [41](denoted as ArcFace*, ArcFace*-M, ArcFace*-L) as the initialization of the upper branch.

 Initialization ULN and ULN-M use pretrained ResNet-18 ArcFace [23] to initialize the upper branch. The batch size = 192, momentum = 0.9, weight decay = 0.0005, learning rate = 0.1, training epoch = 24 and the learning rate will reduce to 1/10 at [12, 18, 22] epoch. ULN-L use pretrained ResNet-50 ArcFace [23] to initialize the upper branch. The batch size = 128, momentum = 0.9, weight decay = 0.0005, learning rate = 0.1, training epoch = 20 and the learning rate will reduce to 1/10 at [10, 15, 18] epoch.

Training Datasets We used the CASIA-WebFace [41] and generated CASIA-WebFace-Mask to train the ULN. The CASIA-WebFace dataset has about 10 thousand individuals and 0.5 million face images in total. VggFace2 [3] and generated VggFace2-Mask is used to train the ULN-L. In the experiment of the MegaFace, we only use VggFace2 to train the ULN-L. VggFace2 is a large-scale face recognition dataset with over 3.3 million faces from more than 9 thousand individuals.

Baseline Model The baseline model using ResNet-18 and ResNet-50 as the backbone for small and large models respectively and ArcFace [23] as its loss function. ArcFace is one of
the best face recognition methods recently and is widely used. Use the ArcFace model as the baseline can prove the effectiveness of our proposed ULN and data augmentation method.

**Testing** When we do a test on the ULN, if an image is known as mask-free image, both features of the lower branch i.e., \( f_{lu} \) and \( f_{ll} \) will be calculated and stored; otherwise, only \( f_{lu} \) will be calculated and stored. When we calculate the cosine distance between two samples, if both of the samples is known as mask-free, the cosine distance will be calculated by using \( f_{lu} + f_{ll} \); otherwise, the distance will be calculated by using only \( f_{lu} \).

### 5.2 Performance on LFW and LFW-mask

LFW [15] dataset contains 13,233 images from 5749 different people. We used the method mentioned in Chapter 3 to generate the PWM images of the LFW dataset, and we named the generated dataset LFW-Mask. LFW-Mask is used to test the robustness of the face recognition model on the PWM task.

We list the model performance result on LFW and LFW-Mask in Table 1. When we use the 128-dimension feature, which is a pretty small model, the result shows that if we do not use the generated PWM data in the training phase, the test accuracy of the baseline model drops about 5.5% when the test images have facial masks. If we retrain the baseline model by the augmented PWM data together with the original CASIA-WebFace dataset, the test result on LFW-Mask improves about 4.4% compared with the baseline model. It is clear evidence that suggests our proposed PWM augmentation method can improve the robustness of the face recognition model facing the PWM tasks. Another thing worth noting is that the test result on the LFW dataset drops more than 1% if we use the augmented data in the baseline model. The drop in the LFW dataset accuracy is foreseeable since the baseline model is focused on the mask-free data, and when we add the augmented PWM data, it has to find a balance point between them. Our proposed ULN trained on augmented PWM data together with the CASIA-WebFace dataset achieves the highest accuracy on the LFW-Mask dataset. The result shows the ability of the ULN structure to handle the challenging PWM task. What’s more, the test result on the LFW dataset is slightly lower than the baseline model, which focuses on mask-free data. When we look at the average accuracy over LFW and LFW-Mask dataset, our proposed ULN performs best and lead the SOTA baseline using

| Method      | feature-dim | LFW   | LFW-Mask | Avg.  |
|-------------|-------------|-------|----------|-------|
| ArcFace*    | 128         | 97.43 | 91.95    | 94.69 |
| ArcFace*+Aug| 128         | 96.35 | 96.32    | 96.33 |
| ULN+Aug     | 128         | 97.40 | **96.35**| **96.88** |
| ArcFace*-M  | 512         | **99.24** | 94.80    | 97.02 |
| ArcFace*-M+Aug| 512         | 97.60 | 97.42    | 97.51 |
| ULN-M+Aug   | 512         | 99.02 | **98.19**| **98.60** |

The third column is the average accuracy of the LFW and LFW-Mask datasets. ArcFace is the SOTA face recognition algorithm, and the ‘+Aug’ means the method uses the generated PWM data for training. We can find that our proposed PWM data augmentation method can help improve the test accuracy on the LFW-Mask dataset a lot. Our proposed ULN outperforms the SOTA face recognition method. The bold entries represent the algorithms that achieve best results in 128 and 512 feature dimensions.
the same training data by 0.55%. An even more significant improvement in our ULN-M also shows when we use a 512-dimension feature.

5.3 Performance in hybrid environment

In a real-world surveillance scene, we want to match the person wearing a mask in the camera to the gallery composed of non-mask photos. That requires our model should be capable of handling the cross-domain problem. We do two experiments to validate our model: 1) do a cross dataset recognition on LFW and LFW-Mask, which means the test pairs are composed of one image with a facial mask and one without a mask. 2) do a test on a newly released real-world mask dataset MFR2 [2] which contains 269 photos from 53 people, and each person has both masked and non-mask photos.

The result is demonstrated in Table 2. LFW-SM is a generated LFW testing set from LFW by adding facial masks, which is similar to our LFW-Mask. We compare our result with MTF [2], which is also focused on the PWM problem and uses augmented facial mask data to help training. Our proposed ULN still maintains high accuracy and leads the baseline ArcFace model by 0.75% on LFW & LFW-Mask dataset and 1.8% on the MFR2 dataset. Our ULN model also has much better performance in low FAR requirements. Our larger ULN-L model outperforms SOTA face recognition methods in all the indicators.

5.4 Performance on MegaFace

MegaFace [17] is a challenging benchmark dataset for large-scale face identification and verification. MegaFace includes more than 1 million images of 690 thousand different individuals collected from Flickr [32] photos as the gallery set and Facescrub [26] which has 100 thousand photos of 530 unique individuals as the probe set. We evaluated the performance of our proposed method on Megafce Challenge under both small protocol whose training dataset contains less than 0.5 million images and large protocol, which does not have a limit on the training data.

We compare our method with many state-of-the-art methods, and the result can be found in Table 3. In the face identification task, we give out the Rank-1 accuracy of these methods, and in the face verification task, we test the TAR(True Accept Rate) under the limitation

| Method       | Augmentation | LFW & LFW-Mask | LFW-SM | MFR2 | MFR2 TAR @FAR=0.2% |
|--------------|--------------|----------------|--------|------|-------------------|
| MTF [2]      | ×            | -              | 91.04  | 90.34| 48.85             |
| ArcFace*     | ×            | -              | -      | 90.45| 50.94             |
| MTF [2]      | ✓            | -              | 97.25  | 95.99| 91.18             |
| LPD [8]      | ✓            | -              | 95.70  | -    | -                 |
| ArcFace*     | ✓            | -              | 95.87  | 93.63| 87.03             |
| ArcFace*-L   | ✓            | -              | 96.30  | 95.05| 90.80             |
| ULN          | ✓            | -              | 96.62  | 95.51| 91.04             |
| ULN-L        | ✓            | -              | 98.02  | 96.82| 92.45             |

Augmentation means the method use the generated PWM data for training. We can find our proposed ULN-L outperforms the other SOTA face recognition method.
Table 3  Face identification and verification evaluation of different methods on MegaFace Challenge 1

| Method             | Protocol | Id (%) | Ver (%) |
|--------------------|----------|--------|---------|
| YouTu Lab          | Large    | 83.29  | 91.34   |
| NTechLAB-facenx    | Large    | 73.30  | 85.08   |
| Vocord-DeepVo3     | Large    | 91.76  | 94.96   |
| DeepSense V2       | Large    | 81.30  | 95.99   |
| Shanghai Tech      | Large    | 74.05  | 86.37   |
| FaceNet [29]       | Large    | 70.50  | 86.47   |
| Beijing FaceAll-N  | Large    | 64.80  | 67.12   |
| Beijing FaceAll    | Large    | 63.98  | 63.96   |
| ArcFace [7]        | Large    | 81.03  | 96.98   |
| CosFace [36]       | Large    | 82.72  | 96.65   |
| UniformFace [10]   | Large    | 79.98  | 95.36   |
| RegularFace [44]   | Large    | 75.61  | 91.13   |
| GRCCCV             | Small    | 77.68  | 74.89   |
| FUDAN-CS SDS [35]  | Small    | 77.98  | 79.19   |
| DeepSense          | Small    | 70.98  | 82.85   |
| Center Loss [37]   | Small    | 65.23  | 76.52   |
| L-Softmax [20]     | Small    | 67.13  | 80.42   |
| SphereFace [19]    | Small    | 72.73  | 85.56   |
| ArcFace [7]        | Small    | 77.50  | 92.34   |
| CosFace [36]       | Small    | 77.11  | 89.88   |
| RegularFace [44]   | Small    | 70.23  | 84.07   |
| ArcFace*           | Small    | 72.78  | 87.53   |
| ULN-L              | Large    | 80.05  | 95.48   |
| ArcFace*           | Small    | 72.78  | 87.53   |
| ULN                | Small    | 71.86  | 85.96   |

“Id” refers to the Rank-1 face identification accuracy, and “Ver” refers to the face verification TAR at $10^{-6}$ FAR. The ArcFace* and our proposed ULN are trained under the same setting for fair comparison. The bold entries represent the algorithms that achieve best results under Large Protocol and Small Protocol of $10^{-6}$ FAR (False Accept Rate). We used the test set without the refinement of MegaFace for fair comparisons. We trained our model together with the ArcFace [7]. The ArcFace* is the version of ArcFace implemented under the Pytorch framework by ourselves and trained with the same settings such as initialization, learning rates, etc. as our ULN for the fairness of the comparisons.

In the large protocol training process, only VggFace2 is used. In this situation, the SCDT loss can be regarded as typical teacher information. We are surprised to see that the result of our ULN-L outperforms the teacher network after training. Under the small protocol, the ULN is trained by CASIA-WebFace and the augmented PWM data. Since the MegaFace is mask-free, the slightly lower accuracy result is acceptable.

We also augmented the Facescrub of the MegaFace to test the occlusion robustness of the model in a challenging situation. The result is shown in Table 4. The Id-Mask column means...
Table 4  Face identification evaluation on MegaFace Challenge 1

| Method          | Protocol | Id (%) | Id-Occ (%) |
|-----------------|----------|--------|------------|
| Center Loss [37]| Small    | 65.49  | -          |
| DeepSense       | Small    | 70.98  | -          |
| SphereFace [19] | Small    | 72.73  | -          |
| PDSN [30]       | Small    | 74.40  | 56.34      |
| ArcFace*        | Small    | 72.78  | 45.11      |
| ULN             | Small    | 71.86  | 57.26      |

“Id” refers to the Rank-1 face identification accuracy, and “Id-Mask” refers the Rank-1 face identification accuracy with the masked Facescrub probe.

the Rank-1 face identification accuracy under the masked Facescrub probe situation. The accuracy is relatively low in this protocol, but ULN surpasses the ArcFace more than 12% and surpasses state-of-the-art occlusion-robust face recognition methods PDSN by 0.92%. In this extreme situation, our proposed ULN architecture shows outstanding robustness on the facial mask occlusion.

5.5 Performance on AR database

AR face database [23] also provides us a real-world occlusion scene to evaluate our model. AR database contains over 4,000 color images from 126 people with different facial expressions, illumination conditions and occlusions. We use the images with scarf occlusion to test both Protocol 1, which uses 14 images from each person to form the gallery set and Protocol 2, which uses only one image from each person to form the gallery set. The result is shown in Table 5. Our ULN achieves the best result on both Protocol 1 and Protocol 2 compare with other occlusion-robust methods.

5.6 Efficiency analysis

In surveillance camera systems, real-time performance is an important requirement. We compare our ULN with the ArcFace in both accuracy and inference speed in Table 6. Our ULN can run almost 3 times faster than ArcFace*-L as well as a much higher accuracy.

Table 5  The test result of our ULN compare with other occlusion-robust method on AR Database. The bold entries represent the algorithms that achieve best results under Protocol 1 and Protocol 2

| Method          | Protocol | Scarf |
|-----------------|----------|-------|
| MLERPM [38]     | 1        | 97.00 |
| SCF-PKR [40]    | 1        | 98.00 |
| RPSM [39]       | 1        | 97.66 |
| MaskNet [34]    | 1        | 96.70 |
| PDSN [30]       | 1        | **100.0** |
| ULN             | 1        | **100.0** |
| Stringface[4]   | 2        | 92.00 |
| LMA[25]         | 2        | 93.70 |
| PDSN[30]        | 2        | 98.33 |
| ULN             | 2        | **98.83** |
Table 6  The number of model parameters and inference time comparison between our ULN and ArcFace

| Method    | MFR2 Acc | Params | Speed |
|-----------|----------|--------|-------|
| ArcFace*-L | 95.05    | 23M    | 257ms |
| ArcFace*   | 93.63    | 11M    | 88ms  |
| ULN-L      | 96.82    | 23M    | 241ms |
| ULN        | 95.51    | 11M    | 82ms  |

The speed is testing on Intel i7-6700K with batch size 1

6 Ablation study

6.1 Significance of the loss

In Table 7 we explore the significance of different loss function. The ULN without using the Selective mechanism in SCDT loss drops 0.58% in the LFW dataset and 0.47% in the LFW-Mask dataset. It suggests the Selective mechanism plays an important role in both improving the accuracy in mask and mask-free datasets. If we remove the total SCDT loss, the ULN will degenerate to the ArcFace structure. The accuracy drops more than 1% on the LFW dataset. The ULN without Recognition Loss means the model trains only under the supervise of the teacher network. This causes about 0.64% drop on average. The result shows that both of the loss functions contribute to the training of the ULN.

6.2 Choice of λ

Figure 6 shows the ULN average test accuracy on LFW dataset and LFW-Mask dataset according to the hyperparameter λ varies from 0.1 to 0.7. λ shows the importance of the SCDT loss compare to the Recognition loss during the training process. From the chart, we easily find the best choice should be 0.5.

6.3 How many mask images do we need in the template?

Figure 7 shows the relation between the test accuracy and the number of mask images in the template used to generate the PWM dataset. Zero mask image means we just black out the mask area and do not stick the mask images on it. We find that the accuracy almost does not

Table 7  Ablation analysis on the loss function

| Method                           | LFW  | LFW-Mask | Avg.  |
|----------------------------------|------|----------|-------|
| ULN                              | 97.40| 96.35    | 96.88 |
| ULN w/o Selective                | 96.82| 95.88    | 96.35 |
| ULN w/o SCDT Loss                | 96.35| 96.32    | 96.33 |
| ULN w/o Rec Loss                 | 96.92| 95.55    | 96.24 |

ULN w/o Selective means the ULN will not use the Selective mechanism in the SCDT Loss, and both of the upper and lower features in the lower branch will compare with the upper branch no matter what the input image is. ULN w/o SCDT Loss degenerate model to the ArcFace structure. ULN w/o Rec Loss means the training does not use a classification signal.
Fig. 6 The test accuracy of the ULN vs $\lambda$ which indicates the importance of SCDT loss compare to Recognition loss during the training. The best choice of $\lambda$ should be 0.5

Fig. 7 The test accuracy of the ULN vs the number of mask images in the template
grow much when the number of facial mask images gets over 20. This indicates that we do not need more than 100 images in the template and tens of them are enough.

7 Conclusion

Affected by the COVID-19 epidemic, people began to wear masks in public to ensure their safety, which brings challenges to the face recognition system at this stage. The current face recognition method failed because of two main shortcomings: lack of large-scale datasets with people wearing masks and not robust enough to large occlusion.

In view of the insufficient dataset of faces wearing masks currently, our paper proposes a method to generate the images of the person wearing a mask from the widely used face recognition datasets to help the training process of the deep learning model. Since the face with the mask has a large amount of distracting information from the covered part, we propose a novel upper-lower structure network (ULN) to overcome the difficulties in the real scene in real-time. We do extensive experiments that show our proposed methods’ outstanding performance and prove their effectiveness in both the PWM tasks and the traditional face recognition tasks.

In future research work, we will continue to explore generating more realistic PWM images, predicting more accuracy covered face parts, and reducing the model size for mobile devices.

Acknowledgements This paper is supported by NSFC (No. 62176155, 62066002), Shanghai Municipal Science and Technology Major Project, China, under grant no. 2021SHZDZX0102.

References

1. Antoniou A, Storkey A, Edwards H (2017) Data augmentation generative adversarial networks. arXiv:1711.04340
2. Anwar A, Raychowdhury A (2020) Masked face recognition for secure authentication
3. Cao Q, Shen L, Xie W, Parkhi OM, Zisserman A (2018) Vggface2: a dataset for recognising faces across pose and age. In: International conference on automatic face and gesture recognition
4. Chen W, Gao Y (2010) Recognizing partially occluded faces from a single sample per class using string-based matching. In: European conference on computer vision, pp 496–509. Springer
5. Cheng G, Han J, Zhou P, Xu D (2018) Learning rotation-invariant and fisher discriminative convolutional neural networks for object detection. IEEE Trans Image Process 28(1):265–278
6. Cheng G, Yang C, Yao X, Guo L, Han J (2018) When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. IEEE Trans Geoscience and remote sensing 56(5):2811–2821
7. Deng J, Guo J, Xue N, Zafeiriou S (2019) Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 4690–4699
8. Ding F, Peng P, Huang Y, Geng M, Tian Y (2020) Masked face recognition with latent part detection. In: Proceedings of the 28th ACM International conference on multimedia, pp 2281–2289
9. Dong E, Du H, Gardner L (2020) An interactive web-based dashboard to track covid-19 in real time. The Lancet Infectious Diseases 20(5):533–534. https://doi.org/10.1016/S1473-3099(20)30120-1. http://www.sciencedirect.com/science/article/pii/S1473309920301201
10. Duan Y, Lu J, Zhou J (2019) Uniformface: Learning deep equidistributed representation for face recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 3415–3424
11. Fu C, Wu X, Hu Y, Huang H, He R (2019) Dual variational generation for low shot heterogeneous face recognition. In: Advances in neural information processing systems, pp 2674–2683
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
13. Howard J, Huang A, Li Z, Tufekci Z, Zdimal V, van der Westhuizen HM, von Delft A, Price A, Fridman L, Tang LH et al (2020) Face masks against covid-19: an evidence review
14. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 7132–7141
15. Huang GB, Ramesh M, Berg T, Learned-Miller E (2007) Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech Rep. 07-49, University of Massachusetts, Amherst
16. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167
17. Kemelmacher-Shlizerman I, Seitz SM, Miller D, Brossard E (2016) The megaface benchmark: 1 million faces for recognition at scale. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 4873–4882
18. King DE (2009) Dlib-ml: A machine learning toolkit. J Mach Learn Res 10:1755–1758
19. Liu W, Wen Y, Yu Z, Li M, Raj B, Song L (2017) Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE conference on Computer vision and pattern recognition, pp 212–220
20. Liu W, Wen Y, Yu Z, Yang M (2016) Large-margin softmax loss for convolutional neural networks. In: ICML, vol 2, p 7
21. Lu C, Tang X (2015) Surpassing human-level face verification performance on lfw with gaussianface. In: Twenty-ninth AAAI conference on artificial intelligence
22. MacIntyre CR, Chughtai AA (2020) A rapid systematic review of the efficacy of face masks and respirators against coronaviruses and other respiratory transmissible viruses for the community, healthcare workers and sick patients. International Journal of Nursing Studies p 103629
23. Martinez A, Benavente R (1998) The ar face database. Tech. Rep 24 CVC Technical Report
24. Masi I, an Trân AT, Hassner T, Leksut JT, Medioni G (2016) Do we really need to collect millions of faces for effective face recognition? In: European conference on computer vision, pp. 579–596. Springer
25. McLaughlin N, Ming J, Crookes D (2016) Largest matching areas for illumination and occlusion robust face recognition. IEEE Trans Cybern 47(3):796–808
26. Ng HW, Winkler S (2014) A data-driven approach to cleaning large face datasets. In: 2014 IEEE International conference on image processing (ICIP), pp 343–347. IEEE
27. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch
28. Peeples L (2020) What the data say about wearing face masks
29. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 815–823
30. Song L, Gong D, Li Z, Liu C, Liu W (2019) Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In: Proceedings of the IEEE International conference on computer vision, pp 773–782
31. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
32. Thomée B, Shamma DA, Friedland G, Elizalde B, Ni K, Poland D, Borth D, Li LJ (2015) Yfcc100m:, The new data in multimedia research. arXiv:1503.01817
33. Trigueros DS, Meng L, Hartnett M (2018) Enhancing convolutional neural networks for face recognition with occlusion maps and batch triplet loss. Image Vis Comput 79:99–108
34. Wan W, Chen J (2017) Occlusion robust face recognition based on mask learning. In: 2018 IEEE International conference on image processing (ICIP), pp 3795–3799. IEEE
35. Wang Z, He K, Fu Y, Feng R, Jiang YG, Xue X (2017) Multi-task deep neural network for joint face recognition and facial attribute prediction. In: Proceedings of the 2017 ACM International conference on multimedia retrieval, pp 365–374. ACM
36. Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, Li Z, Liu W (2018) Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5265–5274
37. Wen Y, Zhang K, Li Z, Qiao Y (2016) A discriminative feature learning approach for deep face recognition. In: European conference on computer vision, pp 499–515. Springer
38. Weng R, Lu J, Hu J, Yang G, TanYP (2013) Robust feature set matching for partial face recognition. In: Proceedings of the IEEE International conference on computer vision, pp 601–608
39. Weng R, Lu J, Tan YP (2016) Robust point set matching for partial face recognition. IEEE Trans Image Process 25(3):1163–1176
40. Yang M, Zhang L, Shiu SCK, Zhang D (2013) Robust kernel representation with statistical local features for face recognition. IEEE Trans Neural Netw Learn Syst 24(6):900–912
41. Yi D, Lei Z, Liao S, Li SZ (2014) Learning face representation from scratch. arXiv:1411.7923
42. Yin J, Liu Z, Jin Z, Yang W (2012) Kernel sparse representation based classification. Neurocomputing 77(1):120–128
43. Zhang K, Zhang Z, Li Z, Qiao Y (2016) Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Process Lett 23(10):1499–1503
44. Zhao K, Xu J, Cheng MM (2019) Regularface: Deep face recognition via exclusive regularization. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp 1136–1144

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Zijun Zhuang is now a PhD candidate in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His current research interest is face recognition.

Hongtao Lu is now a Professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. His current research interests include computer vision, deep learning and machine learning. He had authored or co-authored more than 100 papers in journals and premier conferences.