Improving Representation Consistency with Pairwise Loss for Masked Face Recognition

Hanjie Qian*[1], Panpan Zhang[2], Sijie Ji[1], Shuxin Cao[1], and Yuecong Xu[3]

[1]Nanyang Technology University, Singapore
[2]National University of Singapore, Singapore
[3]Institute for Infocomm Research, A*STAR, Singapore

Abstract

Given the coronavirus disease (COVID-19) pandemic, people need to wear masks to protect themselves and reduce the spread of COVID, which brings new challenge to the traditional face recognition task. Since features like the nose and mouth, which are well distinguishable, are hidden under the mask, traditional methods are no longer simply applicable, even though they once achieved a high degree of accuracy. In response to this problem, the Masked Face Recognition Challenge & Workshop (MFR) was held in conjunction with the International Conference on Computer Vision (ICCV) 2021. This article details a method that combining the classic ArcFace and pairwise loss to target the new masked face recognition task. So far, our method has achieved the second place in the competition.

1. Introduction

The COVID-19 pandemic has permanently changed the original life and production habits of mankind. Now, in order to protect themselves and others and prevent the wanton spread of the virus, most governments require people to wear masks in public places. This brings difficulties to the common face recognition technology [1]. On the one hand, the models of these algorithms are trained on unoccluded face datasets. On the other hand, most algorithms rely on features extracted from the entire face. For faces with masks, the accuracy of these recognition algorithms will significantly decrease. The lack of accurate masked face recognition algorithms has forced people to compromise by taking off their masks and exposing themselves to the risk of infection in certain situations, such as at boarding security checkpoints. Therefore, it is particularly urgent to develop robust face recognition technology for masked face recognition.

Figure 1. The images of faces with and without mask and the results after training with pairwise loss.

The earliest face recognition can be traced back to 1990s [2]. In recent years, with the development of deep learning technology, classical CNN and many of its variants [3, 4] are constantly being proposed. Researchers have proposed many face recognition applications based on deep learning technology and obtained high accuracy with promising prospects. Facebook proposed DeepFace [5] to perform face alignment before training for better feature learning, and this process is now widely used in face recognition pipeline [6]. Some recent margin-based algorithms, such as SphereFace [7], CosFace [8] and ArcFace [9], achieved high accuracy on different face related tasks and face related competitions [10]. However, these algorithms are vulnerable when facing occluded face targets, because a lot of effective information cannot be extracted by the network, resulting in significant accuracy decrease.

In real life, certain parts of the face cannot be obtained for various reasons. Common causes of occlusion include light occlusion due to uneven light distribution, object occlusion due to sunglasses or hats, and self-occlusion due to particular face angles. Thus, the more challenging face

*Corresponding author hanjie001@e.ntu.edu.sg

recognition in obscured conditions has attracted the interest of researchers. Typical mainstream approaches include two categories. One is the reconstruction-based approach [11], and the other is attention mechanism-based approach [12, 13, 14].

Masked face recognition is a special case of occlusion face recognition but has its own characteristics. As mentioned above, in previous studies, the occlusion objects usually covered only a small part of the face. In the case of wearing a mask, half or even more than half of the area is occupied by the mask, especially the mouth and nose with important physiological characteristics. In addition, the lack of masked face data makes large-scale model training prohibitive. As far as we know, there is little work related to face recognition with masks before. From last year, some researchers[15, 16] paid attention to this field but their studies are either based on a small dataset or only involve the detection of masked faces.

Due to the sudden outbreak of the epidemic, there are currently no publicly available masked face recognition benchmarks, which encouraged the holding of the Masked Face Recognition Challenge. Based on the WebFace260M dataset [17], we propose a framework that combines ArcFace method and pairwise loss [18] in order to improve the performance of masked face recognition task. ArcFace is a successful method in face recognition related tasks, which penalizes the angle between the deep features and their corresponding weight vectors in an additive manner, thereby simultaneously enhancing intra-class tightness and inter-class differences. However, for the masked face recognition, such penalty is not enough because the features of the human face become inconspicuous due to the presence of the mask. In this case, we apply pairwise loss into the network to get a better decision boundary. And currently our algorithm ranks second in the competition.

In next section, we will introduce the competition dataset first, then introduce the whole network architecture we used. In section 3, the results of our model are discussed. Conclusions are presented in Section 4.

2. Methodology

2.1. Database

The dataset used in the competition is called WebFace260M, which is a new million-scale face benchmark. The WebFace260M contains noisy 4M identities and 260M faces. And after an automatic cleaning process, the cleaned WebFace42M is obtained. The WebFace42M contains 2M identities and 42M faces, which is the biggest public masked face recognition dataset to date. The WebFace42M contains 7 face attribute annotations, especially the masked faces. Because WebFace42M is relatively large, for more efficient training, we conducted preliminary experiments on WebFace12M. Figure 2 shows some samples from the WebFace260M dataset that we used for training.

Figure 2. Data samples in WebFace260M. The left and right columns of each row are pictures of the same person without and with a mask respectively. The WebFace260M contains noisy 4M identities and 260M faces.

2.2. Algorithm and architecture

In this part, we provide a detailed introduction of the proposed algorithm to improve the performance of the mask face recognition task.

2.2.1 ArcFace loss

As mentioned above, due to the superiority in terms of accuracy, complexity and efficiency, we used ArcFace as the basic of our method.

The ArcFace loss is the extension of softmax function. The softmax function can be presented as follows:

$$L_{softmax} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}, \quad (1)$$

However, softmax function mainly focuses on whether the samples can be correctly classified, and lacks constrains on intra-class and inter-class so that it cannot explicitly increase the gap between different classes and reduce the discrepancy of samples in the same classes. So some researchers proposed an improved loss function, so called ArcFace loss function:

$$L_{Arcface} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{e_i}{e_i + \sum_{j=1, j\neq y_i}^{n} e^{scos\theta_j}}, \quad (2)$$

Here e_i is $e^{s(cos(\theta_{y_i}+m))}$. In the angle space, ArcFace adds the angle θ between the feature and the weight W_{y_i}. In this case, the optimization of loss function will simultaneously enhance the intra-class compactness and inter-class difference.

2.2.2 Pairwise loss

However, for masked face recognition, it is found that the performance of most existing algorithms on this task will

deteriorate. We assume the reasons are that, on the one hand, half area of the face is occluded so many information can not be extracted, which results in the extracted features not sufficiently distinguishable. On the other hand, these algorithms are designed based on the common dataset like LFW (Labled Faces in the Wild), which contains images with different attributes. While for the masked face recognition task, we are more concerned about the performance on face with masks, and this is only a small part of common tasks. In this particular task, *the algorithm should be designed to deliberately reduce the distance between the features of the same face with and without a mask. At the same time, the distance between images of faces belonging to different identities should be expanded*, thus improving the performance. Motivated by this idea, our approach introduces pairwise loss.

The pairwise loss is designed to reduce the intra-class discrepancy and enlarge the difference of inter-class, which has been widely applied in other fields [19]. Suppose that x_i and x_j are the input training samples of dimension $\mathbb{R}^d$. And Y_{ij} represent the similarity between x_i and x_j, the value of Y_{ij} can be obtained:

$$Y_{ij} = \begin{cases} 1, & x_i,\ x_j \ are\ belong\ to\ same\ identities; \\ -1, & otherwise. \end{cases} \tag{3}$$

Then $f^k(x_i|\theta)$ and $f^k(x_j|\theta)$ are the corresponding features extracted from the input data, the L_2 distance is utilized to measure the difference between features:

$$D_k^2 = \left\| f^k(x_i|\theta) - f^k(x_j|\theta) \right\|_2^2; \tag{4}$$

Pairwise loss is a flexible loss function compared to others, such as center loss, which try to minimize the distance between samples and the corresponding class center. Mask and mask free features can be maintained consistency in the feature space by introducing pairwise loss. The explanation of pairwise loss is presented in Figure 1. The pairwise loss contains two hyper-parameters b and m, which generate the margin between different classes, the loss function can be formulated as:

$$L_{pairwise} = max(0,\ b - Y_{ij}(m - D_k^2(x_i, x_j, \theta))); \tag{5}$$

Here, $0 < b < m$. And after the training with pairwise loss, the distance of samples from the same identity, with and without masks, will be smaller than a margin $m - b$. On the contrary, the samples from different identities will be bigger than the margin $m + b$. So if the b is equal to m, the L_2 distance of two related samples will be limited to 0, just like the center loss [20] or contrastive loss, which is too strong for the real task.

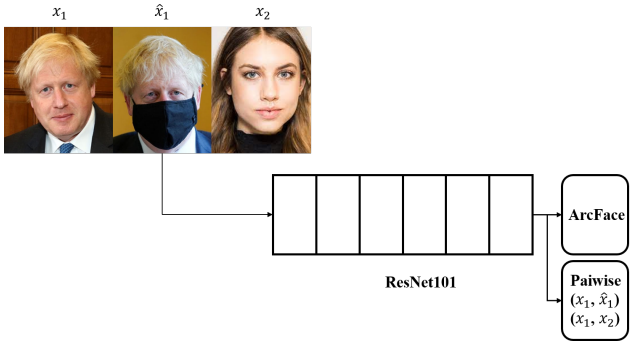

Figure 3. Illustration of the network architecture. The ResNet101 is used as the backbone, then the ArcFace loss and pairwise loss are obtained based on the extracted features. In each step, images from the same and different identities are feeded into the network.

2.2.3 Architecture and training

The network structure we proposed in this competition is shown in Figure 3. The ResNet101 [21] is used as backbone to extract features. Then the features will be feed into the ArcHead to obtain the ArcFace prediction. Comparing the ArcFace prediction and ground truth, the cross-entropy loss can be calculated. Meanwhile, considered that the pairwise loss is applied in our framework, in each training batch, three samples will be selected as a group, (x_i, y_i),$(\hat{x_j}, y_i)$ and (x_k, y_k). A face image without mask is selected randomly in first step, donated as (x_i, y_i). Correspondingly, the same identity with mask is chosen, donated as $(\hat{x_j}, y_i)$. In the last, an image from another class is added into the group as (x_k, y_k). In each training step, the pairwise loss contains the intra-class loss and inter-class loss. Our final optimization function can be formulated as:

$$L_{total} = L_{ArcFace} + L_{Pairwise}; \tag{6}$$

All of the code is completed by Pytorch and trained on NVIDIA Tesla V100 32GB. And SGD [22] method is used to update the parameters of the network.

3. Results and discussion

To evaluate the performance of our proposed method, all of the experiments is conducted on the benchmark dataset WebFace: WebFace12M and WebFace42M. We firstly train our algorithm on WebFace12M for better efficiency.

3.1. Evaluation metric

Three main metrics in the competition are all-masked score, wild-masked score and controlled masked score.

These metrics can be calculated as:

$$\begin{aligned} All_{Masked(MFR)} &= 0.25 * Old\ All_{Masked(MFR)} \\ &+ 0.75 * All_{(SFR)} \\ Wild_{Masked(MFR)} &= 0.25 * Old\ Wild_{Masked(MFR)} \\ &+ 0.75 * Wild_{(SFR)} \\ Cont_{Masked(MFR)} &= 0.25 * Old\ Cont_{Masked(MFR)} \\ &+ 0.75 * Cont_{(SFR)}; \end{aligned} \tag{7}$$

These MFR metrics actually reflect a weighted average of masked face recognition(MFR) and standard face recognition(SFR). And so far, our method ranks second in the all-masked score. Ranked second and fifth in wild-masked score and controlled-masked score, respectively.

3.2. Results

Because WebFace42M is relatively bigger than WebFace12M, for a higher training efficiency, the experiments are performed on WebFace12M firstly. The result is shown in Table 1.

Method	All-masked	Wild-masked	Cont-masked
ArcFace	0.2037	0.2128	0.1356
Arc+Pairwise	**0.1438**	**0.1715**	**0.0997**

Table 1. Score trained on WebFace12M with ArcFace and ArcFace+Pairwise loss.

Based on WebFace12M, we evaluate our proposed method's performance. Overall, training the network with an effective combination of ArcFace loss and pairwise loss showed significant improvements over training the network with ArcFace loss alone in all three metrics, corresponding to 29.4%, 19.4%, and 26.5% for New All-Masked (MFR), New Wild-Masked (MFR), and New Controlled-Masked (MFR), respectively. Concretely, if we only use ArcFace loss as the optimization function, the score of All-masked MFR is 0.2037. While if the ArcFace loss and pairwise loss are united to optimize network parameters, the score is improved to 0.1438. Besides, the model also perform better on the other two metric. And it is believed that pairwise loss can effectively shorten the distance between mask free face features and masked face features in the latent feature space. Meanwhile, two hyper-parameters b and m push away the distance of samples from different classes larger than the margin $m + b$.

Naturally, the values of b and m will affect the performance of the algorithm. If b is close to 0, the gap of intra-class and inter-class will be too small for a robust classifier. For another thing, if b is set to be close to m, the penalty of intra-class will be too strong for training. Here, m is fixed as 1 and we only adjust the value of b between 0 and 1 with 0.2 as the interval. Namely, b is set to 0.2, 0.4, 0.6 and 0.8 for evaluation. The scores of different values of hyper-parameters are summarized in Table 2.

Value(m=1)	b=0.2	b=0.4	b=0.6	b=0.8
All-masked	0.1455	0.1335	**0.1332**	0.1452

Table 2. Performance with different parameters trained on WebFace12M.

The best score, 0.1332, is achieved when b is equal to 0.6. As analyzed above, when the value of b is close to 0 or 1, it is observed that the performance will decrease.

We also conduct the experiments on WebFace42M dataset and save the model as the final version. WebFace42M contain much more identities and images than WebFace12M, which will undoubtedly improve the performance of algorithm and its generality. In the WebFace42M, the hyper-parameters b and m are set to be 0.6 and 1 respectively.

Dataset	All-masked	Wild-masked	Cont-masked
WebFace12M	0.1332	0.1586	0.0928
WebFace42M	**0.1036**	**0.1246**	**0.0699**

Table 3. Scores of model trained on WebFace12M and WebFace42M.

From the Table 3, we can find that the application of WebFace42M improve the performance of model obviously, from 0.1332 to 0.1036. There is no doubt that richer data can train a more robust network. And we speculate that if the network is trained on Webface260M, the performance of the network can be further improved.

4. Conclusion

In this paper, we proposed a framework which combines the ArcFace and pairwise loss to improve the performance of the masked face recognition. Although the masked area causes difficulties for the traditional face recognition algorithm, pairwise loss can effectively map the images of the same identity with and without wearing a mask into a restricted region in the feature space. At the same time, it will improve the distinguishability of the features of images from different identities. The values of hyper-parameters in pairwise loss are also discussed. Finally, our algorithm is performed on the currently biggest face recognition dataset, WebFace42M and we got the second place in the competition. We believe that the publication of WebFace will promote the development of face recognition as a new benchmark.

References

[1] Dan Zeng, Raymond Veldhuis, and Luuk Spreeuwers. A survey of face recognition techniques under occlu-

sion. *arXiv preprint arXiv:2006.11366*, 2020. 1

[2] Matthew Turk and Alex Pentland. Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 01 1991. 1

[3] Chen Wang, Jianfei Yang, Lihua Xie, and Junsong Yuan. Kervolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 31–40, 2019. 1

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[5] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014. 1

[6] Jianfei Yang, Kai Wang, Xiaojiang Peng, and Yu Qiao. Deep recurrent multi-instance learning with spatio-temporal features for engagement intensity prediction. In *Proceedings of the 20th ACM international conference on multimodal interaction*, pages 594–598, 2018. 1

[7] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 1

[8] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 1

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1

[10] Sijie Ji, Kai Wang, Xiaojiang Peng, Jianfei Yang, Zhaoyang Zeng, and Yu Qiao. Multiple transfer learning and multi-label balanced training strategies for facial au detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 1

[11] Shiming Ge, Jia Li, Qiting Ye, and Zhao Luo. Detecting masked faces in the wild with lle-cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2682–2690, 2017. 2

[12] Jianfeng Wang, Ye Yuan, and Gang Yu. Face attention network: An effective face detector for the occluded faces. *arXiv preprint arXiv:1711.07246*, 2017. 2

[13] Kai Wang, Xiaojiang Peng, Jianfei Yang, Debin Meng, and Yu Qiao. Region attention networks for pose and occlusion robust facial expression recognition. *IEEE Transactions on Image Processing*, 29:4057–4069, 2020. 2

[14] Kai Wang, Xiaojiang Peng, Jianfei Yang, Shijian Lu, and Yu Qiao. Suppressing uncertainties for large-scale facial expression recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6897–6906, 2020. 2

[15] Feifei Ding, Peixi Peng, Yangru Huang, Mengyue Geng, and Yonghong Tian. Masked face recognition with latent part detection. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2281–2289, 2020. 2

[16] Yande Li, Kun Guo, Yonggang Lu, and Li Liu. Cropping and attention based approach for masked face recognition. *Applied Intelligence*, 51(5):3012–3025, 2021. 2

[17] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. Webface260m: A benchmark unveiling the power of million-scale deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10492–10502, 2021. 2

[18] Ya Li, Xinmei Tian, Xu Shen, and Dacheng Tao. Classification and representation joint learning via deep networks. In *IJCAI*, volume 2017, page 67, 2017. 2

[19] Jianfei Yang, Han Zou, Yuxun Zhou, and Lihua Xie. Learning gestures from wifi: A siamese recurrent convolutional architecture. *IEEE Internet of Things Journal*, 6(6):10763–10772, 2019. 3

[20] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 3

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer*

vision and pattern recognition, pages 770–778, 2016. 3

[22] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012. 3