Data Science

Capstone Report - Fall 2022

Mask-aware Face Recognition: An Implicit Approach

Xinhao Liu,
Zihan Shao

supervised by
Li Guo

Preface

This report is for the Data Science Capstone project at NYU Shanghai. The content in the following of this report is a joint work of Xinhao Liu and Zihan Shao under the supervision of Prof. Li Guo, in the Fall 2022 semester.

Acknowledgements

The authors would like to thank Professor Li Guo for the mentorship of the capstone capstone project. Without her help, it would be impossible for us to finish this project.
We would like to thank the CS/DS Capstone Faculty team for conducting such a great Capstone course.
We would like to acknowledge computation resource from NYU High Performance Computing. All experiments in this project is done on the partitions at NYU HPC.
Last but not least, We would like to acknowledge all NYU Shanghai Computer Science and Data Science faculties for their inspiring teaching in the past 4 years, without which it would be impossible for 2 boys who haven't wrote a single line of code before college to complete such a splendid project.

Abstract

Face recognition plays an important role in our daily life. Modern face recognition system has achieved great success in this field with the help of deep learning. However, since the outbreak of COVID-19, people's faces are usually covered with masks, which invalidates the usual face recognition pipeline. Instead of discarding the occluded part by explicitly locating the mask, we adopt an implicit approach. We propose a usual face recognition pipeline with a distillation loss during pairwise training that can implicitly extract the informative feature shared by both masked and unmasked face images, thus improving the model's performance in mask-aware face recognition tasks. We introduce two types of distillation losses together with an optimal search of the weight when combined with the usual face recognition pipeline. Discussion and analysis of their behavior are provided based on both the experiment results and their mathematical foundation. Face alignment is also conducted to improve models' performance on out-of-distribution data.

Keywords

Artificial Intelligence, Computer Vision, Face Recognition, Distillation Loss

Contents

1 Introduction

Face recognition (FR) is an important module in modern authentication systems. It recognizes the identity of a person through visual images or videos. The module is widely used in access control or act as password in personal devices. Historically, many methods are proposed to recognize faces in images [1, 2, 3]. In particular, face recognition with noise is often discussed in literature [4, 5, 6, 7] because of this is a very common case in real-world applications of face recognition systems.

Similar to other general face recognition methods, noise-robust FR systems have a feature extraction module that extracts high-dimensional feature from images or videos. In addition, they also employ modules that are designed to make the system more robust to noise. This problem has been thoroughly studied in recent years and state-of-the-art algorithms can reach a relatively equal accuracy to human beings.

However, with the continuous impact of COVID-19 since 2020, people wear face masks in daily life in order to minimize the spread of the virus. The performance of face recognition systems has experienced huge challenges because face masks occlude a significant part of human faces. Even those specially designed to handle noises do not have a satisfactory performance in this case. Without solving this problem, common face recognition systems in our daily life cannot function properly, and it causes lots of inconvenience.

We approach the problem with the inspiration from widely-used face recognition systems, where a deep network is used to extract feature representations of images, and a person's identity is recognized by comparing the distance in the feature space. We adapt this pipeline into mask-aware face recognition by adding an extra distillation loss function. The loss can guide the network to learn the same feature representation for an identity both for normal images and masked images. By adding the distillation loss between the features generated by the network for these two images, it can guide the network to learn a feature representation that is invariable of masks.

The source code for this project is available at `https://github.com/Gaaaavin/mafr`. In summary, our contributions in this project are the following:

- We discuss the influence of distillation loss on the performance of the model in terms of its weight, and weight.
- We investigate the importance of face alignment in the real-world application of face recog-

nition systems.

- We conduct extensive experiments to prove our finds, and also to understand the divergence of accuracy between in-distribution and out-of-distribution data.

2 Related Work

2.1 General face recognition

Network Architecture Because the similarity between face recognition problem and image classification problem, many successful backbone network in the ImageNet challenge [8] are widely used in face recognition, including VGGNet [9], ResNet [10], and SENet [11]. Since these network architectures are well adapted in the problem of face recognition, it is unnecessary to design an architecture that specializes in face recognition. In most of the literature in this field, these architectures are either directly used or used only with very slight modification.

Classification loss. Because of its essential as a few-shot learning problem, a well-designed loss function during the training process has attracted much attention. An intuitive idea is to use the loss functions in the classification problem. DeepFace [12] is the first to use softmax loss in face recognition. Liu et al. [13] propose a large margin loss with the desire for a more discriminative classification. Recently, angular-margin-based loss functions are adopted as an improvement of large margin loss and has many variants [14, 15, 16, 17]. These methods have achieved very good performance on public datasets. One of the problems with these loss functions, though, is they are very difficult to optimize. As these loss functions are all based on cross entropy, they all have a term summing over all classes on the denominator. This is usually impossible to train in a very large dataset that contains millions of identities because of limited memory.

Contrastive loss. It comes very intuitively from the objective to minimize inter-class distance and maximize intra-class distance. The use of contrastive loss also emerges very early in the development of face recognition [18, 19]. Compared to classification loss, it doesn't prone to the difficulty in optimization. However, it also has a problem of instability due to the selection of training tuples. Wen et al. [20] approaches the problem by leveraging the idea of a clustering center. However, Sun et al. [21] points out that one of the problems with contrastive based loss is the ambiguity between the positive term and negative term. They propose a loss function named Circle Loss that gives weights on the positive and negative terms based on there distance to the optimum. Their method has achieved a higher accuracy than some classification-loss-based methods [16] in many public datasets.

2.2 Mask-aware Face Recognition

The performance of general face recognition models degenerate in case of masked face [22]. To alleviate the degeneration, two opposite methods are proposed in general: 1) to remove the masked part of the face and do recognition with only the uncovered part, 2) to learn a feature representation that is invariant to masks and try to recover the face from the representation.

Mask removing. While non-learning algorithms could leverage hand-crafted feature descriptors such as SIFT [23] to localize the masks and remove them, most of the state-of-the-art methods use deep neural networks to localize corrupted feature (i.e. masked part of the face) and give a lower weight for these features in recognition. Song et al. [24] proposed an end-to-end model that generates a feature discarding mask for input feature to eliminate corrupted feature in the recognition process by matching between masked and unmasked face of one person. Similarly, Qiu et al. [25] proposed an end-to-end pipeline which first detects the corrupted features by an encoder-decoder structure and then cleans them by dynamically learned masks.

Metric learning. Though it is intuitively more challenging compared to mask removing because a successful latent space learning requires large dataset and carefully designed learning method, there are still several works in this direction. Zhao et al. [26] proposed an LSTM-Autoencoders model to effectively restore partially occluded faces. After restoring the masked features, the recovered images can be directly passed to a general face recognition pipeline. Though not directly aimed for face recognition, He at el. [27] also provides an insight to learn the latent space using masked auto-encoders, and have very good recovering performance.

2.3 Dataset

Because of the nature of face recognition as a zero-shot learning problem, usually the model or pipeline is first trained on a large dataset, then tested on a separate, usually much smaller testing set. This also mimics the case of real-world implementation where it is impossible to train the model on the collected database. We follow this convention and divede our discussion into training dataset and testing dataset.

Training. MS-Celeb-1M [28] is a widely used training dataset that contains images of 1 million celebrities. The identities are manually labeled and with careful evaluation protocols. Recently, WebFace42M, as a cleaned subset of WebFace260M [29], is the largest public dataset for face recognition so far. It contains 42 million images of about 2 million identities. However, such large dataset is usually impractical to use conventional high performance computing partitions, because of the I/O bottleneck.

Testing. Testing datasets often contains a small amount of faces and identities that is similar to real applications. LFW [30] is a widely used dataset for evaluation and bechmarking. It contains more than 13,000 labeled images of faces collected from the web over 5,000 identities. CFP-FP [31] is another testing dataset that features for frontal and profile views of celebrities.

Data Augmentation. No matter the purpose is to adapt the algorithms in general face recognition to the masked cases, or to do metric learning and recover the masked part, data augmentation is a crucial part in data pre-processing, because most large-scale training datasets do not contain masked faces. Recently, a few techniques to add mask to a normal face are proposed and have been proved to have performance [32, 33].

3 Method

3.1 Overview

By treating the identity of a person as a class, the problem of face recognition can be generally formulated as a multi-class classification problem. Given an image, the system is expected to find the corresponding identity of the image. Following the general classification pipelines, a common practice is to use the deep network as a feature extractor and use multi-layer perceptrons after the network to produce a classification probability distribution. Finally, the cross-entropy loss is used to evaluate the distribution and update the parameters in the network.

Most of the related works' approach is to discard the masked (occluded) part of the face. To realize this goal, they either use an object detection pipeline or an auto-encoder structure to explicitly locate the masked part. Our approach, however, is to discard these occluded parts implicitly. **We propose that a usual face recognition pipeline with a distillation loss during pairwise training can implicitly extract the informative feature shared by both masked and unmasked face images, thus improving the model's performance in mask-aware face recognition.**

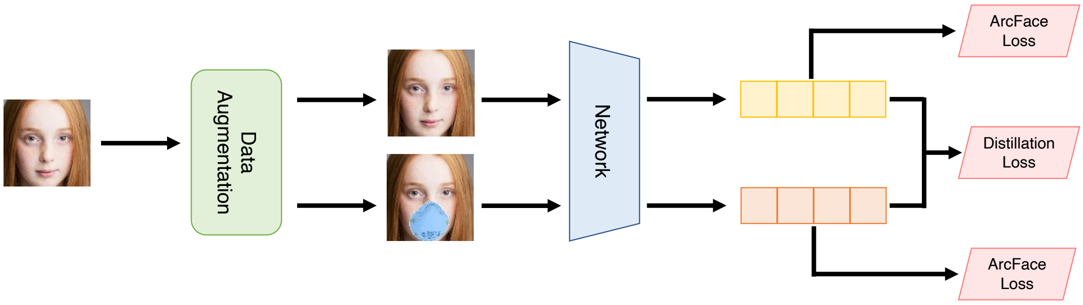


Figure 1: Training pipeline.

3.2 Pipeline

We generally follow the pipeline design in [16]. As shown in Fig. 1, the pipeline of our method has three main components. Before the image is fed into the network, we first augment the data because the faces in the training dataset do not wear masks. After adding masks to the images, a pair of images are input into the network. Two separate cross-entropy loss are calculated according to the features. The distillation loss is computed between the two features. We arrange the rest of this section as the following: Sec. 3.3 introduces the data augmentation and other pre-processing method in our pipeline; Sec. 3.4 reviews the loss function in ArcFace [16]; and Sec. 3.5 explains in detail our proposed distillation loss function.

3.3 Image Pre-Processing

Data augmentation. We employ the method described in [33] (`MaskTheFace`) to add masks to faces in the dataset. The method firstly find conventional facial keypoint descriptors from the images. It has a database of different kinds of masks (surgical, N95, etc.) and they are stored in a way that can be matched to the keypoint descriptors. After the keypoints are detected, a direct map is computed to map the masks onto the image. In our implementation, we find that the computation time for this data augmentation is comparable or larger than the training time on an `A100` GPU. Hence, we perform the augmentation step offline, i.e., we store the masked image in a separate dataset and directly load it from disk during traning in order to avoid the bottleneck.

Face alignment. In order to minimize the performance loss occurred when implementing the trained model in a separate dataset that might not have the same data distribution as that in the training dataset, we align the face in the image before feeding it into the network. This is down by using the trivial cascade classifier [34]. Similar to data augmentation, we align the images in an offline manner.

3.4 ArcFace Loss

ArcFace [16] is state-of-the-art algorithm in modern face recognition systems. Eq. 1 describes the loss function proposed in the paper. Here, θ denotes the angle between the extracted feature and the trained feature centroids, y_i denotes the ground truth label of the input, m is a margin hyperparameter for penalty, and s is a scaling factor for more stable training.

$$\mathcal{L}_{\text{ArcFace}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}} \tag{1}$$

Compared to more frequently used cross-entropy loss in classification tasks, the ArcFace loss is able to learn a feature representation that can be better distinguished. This is especially significant in face recognition tasks because the images are more similar to each other than those in classification tasks. It also helps a better incorporation of the margin m. However, since this loss function is designed for general face recognition, it cannot well handle the case of face masks. Hence, an additional loss function is needed to achieve a better performance in our task.

3.5 Pairwise Distillation

For now, the loss function only consists of the independent cross-entropy losses of the raw and masked face image, and no correlation between them is considered. The goal of our model, however, is to extract the feature of the face shared by masked and unmasked faces. Besides having good classification accuracy, the masked face and raw face should have similar feature vectors after being processed by the model. Therefore, we input paired masked and raw face images during training, and the distillation loss between their extracted features (denoted γ_r and γ_m) is added to the loss.

3.5.1 Options for Distillation Loss

There are several options for distillation loss. The first one is the most commonly used MSE loss

$$\mathcal{L}_{dist}^{MSE}(\gamma_r, \gamma_m) = \|\gamma_r - \gamma_m\|_2^2 \tag{2}$$

Besides the usual MSE loss, we can measure the cosine similarity between 2 vectors, namely

$$\mathcal{L}_{dist}^{\cos}(\gamma_r, \gamma_m) = 1 - \cos\left(\text{angle between } \gamma_r \text{ and } \gamma_u\right) = 1 - \frac{\langle \gamma_r, \gamma_m \rangle}{\|\gamma_r\|_2 \|\gamma_m\|_2} \tag{3}$$

Also, to be fully consistent with the ArcFace loss, we can add the same margin to the loss function as how we do in ArcFace loss. The loss function then reads

$$\mathcal{L}_{dist}^{\cos}(\gamma_r, \gamma_m) = 1 - \cos\left(\arccos(\frac{\langle \gamma_r, \gamma_m \rangle}{\|\gamma_r\|_2 \|\gamma_m\|_2}) + m\right) \tag{4}$$

This might be a better choice in the context as it is consistent with ArcFace loss. Experiments will be conducted to determine which type of distillation is better.

With this construction, the loss function is the (weighted) sum of the cross-entropy loss and the distillation loss.

$$\mathcal{L} = \mathcal{L}_{ArcFace} + \beta \cdot \mathcal{L}_{dist}, \quad \beta \in \mathbb{R}^+ \tag{5}$$

3.5.2 Balancing $\nabla\mathcal{L}_{ArcFace}$ and $\nabla\mathcal{L}_{dist}$

One big prerequisite for this method to work is the $\nabla\mathcal{L}_{dist}$ and $\nabla\mathcal{L}_{ArcFace}$ should have comparable magnitude. If $\|\nabla\mathcal{L}_{dist}\| \ll \|\nabla\mathcal{L}_{ArcFace}\|$, then the distillation loss is not making contributions to the model. If $\|\nabla\mathcal{L}_{dist}\| \gg \|\nabla\mathcal{L}_{ArcFace}\|$, then the model will fail to distinguish the identity of the faces. (e.g. sending β to ∞, then a constant function will be optimal). This is why the weight β is important here. The weight β is to make sure that $\nabla\mathcal{L}_{dist}$ and $\nabla\mathcal{L}_{ArcFace}$ have reasonable magnitude.

Unfortunately, we don't have a good way to determine the weights, which implies a grid search of β is likely to be necessary. What's more, the importance of distillation loss might vary during training, which suggests a dynamic weight β (see next section). One intuitive approach is to perturb β based on the quotient of $\mathcal{L}_{dist}$ and $\mathcal{L}_{ArcFace}$. However, this approach has an invalid mathematical foundation, as the magnitude of the function has no direct relation with its gradient.

Instead of changing β, there's an alternative approach. At each iteration, the distillation loss is only taken for those correctly classified data. This means, we only require those correct classified images to have the same feature vectors, those incorrect should first be correct.

4 Results and Discussion

4.1 Overview

Dataset. We mainly use two different datasets in our training and test process. The WebFace260 dataset [29] is used during training, and the LFW dataset [30] is used in the testing process. This setting is chose particularly because of the real application of face recognition systems. Usually, the model is pre-trained on a large dataset to learn a robust feature representation. It is then implemented in real-world systems where the testing data and identity has no overlap with the training data. WebFace260M is a dataset that contains 42 million images of 2 million identities and is known for the largest open face dataset up-to-date. LFW is a dataset that contains about 5000 identities and 13,000 images of celebriti faces. Due to its relatively smaller size, it is usually used for testing.

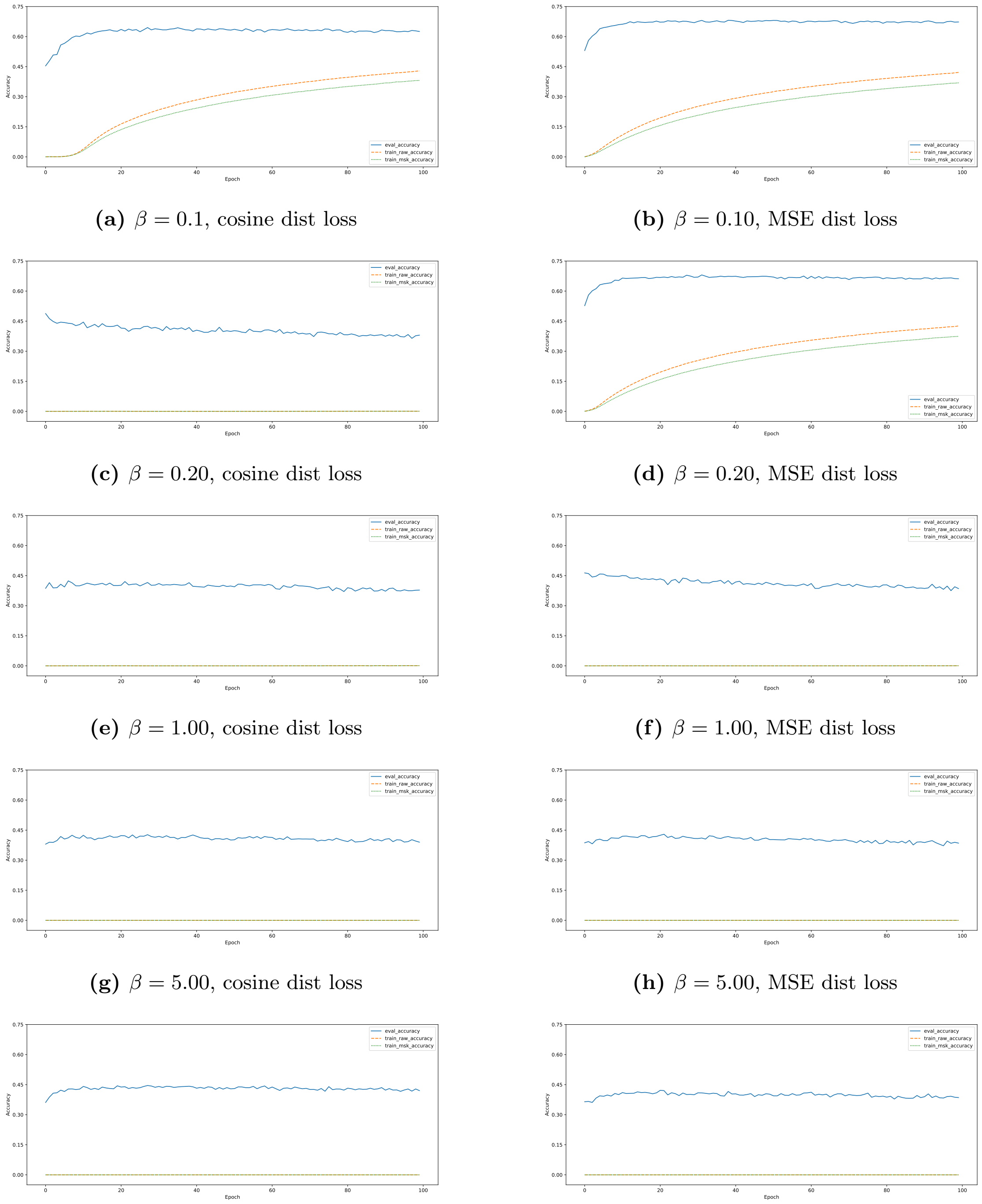
Implementation details. We use a subset of WebFace260M (30,000 identities) for training. The backbone network is the pretrained ResNet18 [10]. The batch size is set to 512. The optimizer is [35] with a learning rate of 2×10^{-4}. All training in this section is run on a single NVIDIA A100 GPU.

Evaluation protocol. There are several evaluation protocols to use in the area of face recognition. For our task, we choose the one that best resembles an access control system. With a trained model, we first input a single image for each identity into the network and get a feature vector. Then, the feature vectors of all identities are gathered to build a database. In the actual testing phase, an input image is fed into the network. The output feature is compared with all features in the database to find the closest one. The corresponding identity is compared with the ground truth identity of the input to calculate prediction accuracy.

4.2 Distillation Loss

Preliminary results show that it is generally difficult to make the distillation loss function well. The model is generally very sensitive to added distillation losses. Note that, as the results with margined Cosine Similarity distillation loss given by Eq. 4 are too bad, we omit it in the following part.

From the grid search for β, one may observe the model is very sensitive to the distillation loss. (See Fig. 2). In cases of β, we observed the case where a big weight for distillation loss will invalidate the cross-entropy loss, thus being not able to detect different identities. Also, even a



(a) $\beta = 0.1$, cosine dist loss

(b) $\beta = 0.10$, MSE dist loss

(c) $\beta = 0.20$, cosine dist loss

(d) $\beta = 0.20$, MSE dist loss

(e) $\beta = 1.00$, cosine dist loss

(f) $\beta = 1.00$, MSE dist loss

(g) $\beta = 5.00$, cosine dist loss

(h) $\beta = 5.00$, MSE dist loss

(i) $\beta = 10.00$, cosine dist loss

(j) $\beta = 10.00$, MSE dist loss

Figure 2: Grid Search of β

small change in β might invalidate the model (see Fig. 2b and Fig. 2d).

4.2.1 Dynamic balancing of the gradient

Given the difficulty of setting a constant weight, we consider dynamically balancing 2 gradients. The scheme is only adding the distillation loss of the correctly classified samples in each iteration.

It has the following advantages:

- **It indeed changes the 'weight' of the distillation loss dynamically.** In the first several epochs, the model doesn't have good accuracy and priority should be given to the ArcFace loss. In this case, the weight of distillation loss is small since only a few samples have contributed to it. However, in later epochs when the model already has a good accuracy due to small ArcFace loss, the distillation loss will have a bigger weight, thus further optimizing the model.

- **The correctly classified sample will give more useful distillation loss** since when both the feature vector are away from the ground truth, their distillation loss is of little importance and might also be misleading.

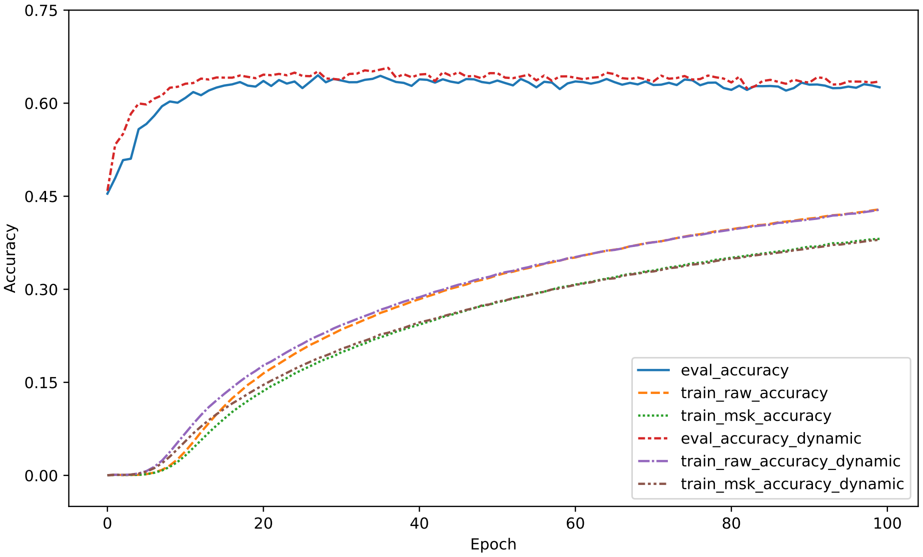


Figure 3: The model with distillation loss only from correctly classified samples (dynamical balancing) has a better performance.

With this training strategy, we get a better result (see fig.3). One good thing about this training strategy is it can be used for all kinds of distillation losses function.

4.2.2 Discussion

Despite our excessive trials and hyper-parameters tuning, our models haven't significantly outperformed the model with no distillation loss at all. While our approach is conceptually correct, we can further improve the results in the following ways.

- Different distillation loss should be considered. The Cosine similarity are overlapped with the ArcFace loss. Since the Cosine similarity makes sure the feature vectors, both from masked and raw images, have a small angle with the target. This automatically ensures that the angle between feature vectors from masked and raw images is small.

- In this sense, MSE is a slightly better option, which is also proved by the experiment (see Table.). However, it is essentially the same. With (2) and (3), we have

$$\begin{aligned}
\mathcal{L}_{dist}^{MSE}(\gamma_r, \gamma_u) &= \|\gamma_r - \gamma_m\|_2^2 && (6)\\
&= \|\gamma_r\|_2^2 + \|\gamma_m\|_2^2 - 2 \cdot \langle \gamma_r, \gamma_m \rangle && (7)\\
&= \|\gamma_r\|_2^2 + \|\gamma_m\|_2^2 - 2 \cdot \frac{\langle \gamma_r, \gamma_m \rangle}{\|\gamma_r\|_2 \|\gamma_m\|_2} \cdot \|\gamma_r\|_2 \|\gamma_m\|_2 && (8)\\
&= \|\gamma_r\|_2^2 + \|\gamma_m\|_2^2 - 2 \cdot (1 - \mathcal{L}_{dist}^{\cos}(\gamma_r, \gamma_u)) \cdot \|\gamma_r\|_2 \|\gamma_m\|_2 && (9)\\
&= (\|\gamma_r\|_2 - \|\gamma_m\|_2)^2 + 2 \cdot \mathcal{L}_{dist}^{\cos}(\gamma_r, \gamma_u) \|\gamma_r\|_2 \|\gamma_m\|_2 && (10)
\end{aligned}$$

- To avoid the overlap between the ArcFace losses and distillation loss, The distillation loss should also be taken from shallow layers.

4.3 Evaluation Data

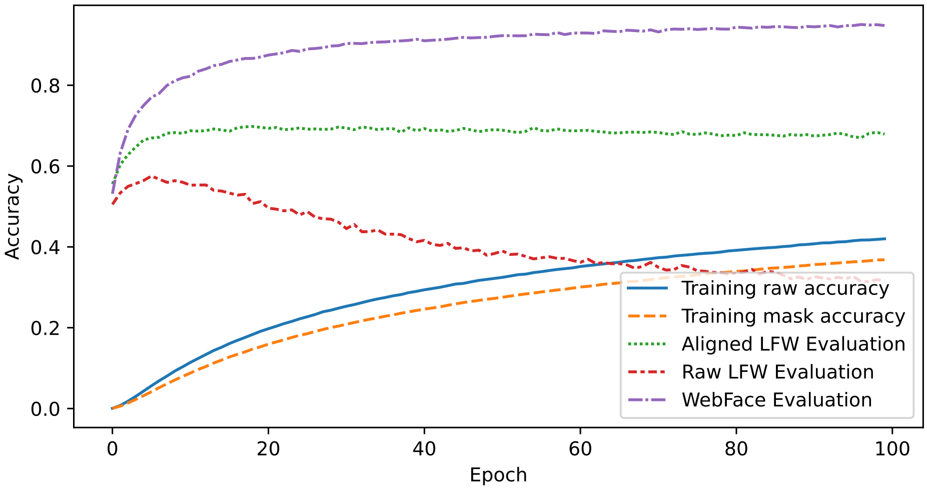


Figure 4: raining curve of different evaluation data

As mentioned in Sec. 2.3, the implementation case of face recognition system makes it very different from common classification problems. Although it is convention to use a different dataset for evaluation, we also do experiment to compare the performance of our method on different evaluation dataset. The evaluation accuracy along the training epochs on LFW [30] and WebFace260M [29] is shown in Fig. 4. It is worth noting that the evaluation dataset from WebFace260M does not contain same identities that are in the training dataset. Thus, although using the same dataset for evaluation, this is still different from the train-and-test procedure in image classification.

It can be seen from the figure that the evaluation accuracy on WebFace260M can achieve a nearly perfect accuracy. On the other hand, although the accuracy on LFW can achieve a relatively satisfactory level, there is a large margin between them. This is because the WebFace260M evaluation data is exptected to have the same distribution with the training set, but is assumption is not true for LFW. This shows how in-distribution and out-of-distribution data can affect the performance of the system. This result also motivates future research in minding the gap between the two types of data.

Moreover, it can also be observed that face alignment place a significant role in our method. This is also because of the different data distribution in the training and evaluation dataset. In LFW dataset, the images contain a larger part of background. Hence, by face alignment, the unnecessary and noisy information in the image input can be discarded enables the network to achieve a higher accuracy.

4.4 Influence of Mask

Table 1: Evaluation accuracy with different mask portion

Loss function	Mask portion	Eval accuracy
ArcFace	0	69.47%
ArcFace+Dist	0	69.67%
ArcFace	0.5	67.42%
ArcFace+Dist	0.5	67.72%
ArcFace	1.0	65.93%
ArcFace+Dist	1.0	65.63%

We also discover how different masking portion in the evaluation dataset affects the evaluation accuracy. Here, masking portion means the portion of images that are added mask in the preprocessing. Tab. 1 shows the results in this experiment, where the distillation loss used is the cosine similarity loss. It can be observed from the table that the influence of mask is marginal compared to other factors discussed above in this section. This can be explained by the data augmentation in the training process. Because both raw and masked image are fed into the network to learn an invariant feature representation, the model is able to make retaliative accurate predictions regardless of the existence of mask in the evaluation.

5 Personal Contributions

The two authors have equal contributions to this work.

6 Conclusion

6.1 Limitation

Although our proposed method improves the identification accuracy for masked face images, there is still a large margin for improvement in order for the system to be valid for actual implementation.

In future works, we are motivated to close the gap between in-distribution and out-of-distribution data. Following this line of thought, it is promising to significantly improve the performance of this method in real applications.

6.2 Summary

Our proposed method adds a distillation loss function to the conventional face recognition pipeline. This method improves the performance of face recognition systems on masked faces. We also explain the failure mode of certain types of distillation function. We believe our work motivates future research in mask-aware face recognition.

References

[1] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

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

[3] W. Deng, J. Hu, J. Lu, and J. Guo, “Transform-invariant pca: A unified approach to fully automatic facealignment, representation, and recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 6, pp. 1275–1284, 2013.

[4] A. Martinez and R. Benavente, “The ar face database: Cvc technical report, 24,” 1998.

[5] F. Wang, L. Chen, C. Li, S. Huang, Y. Chen, C. Qian, and C. C. Loy, “The devil of face recognition is in the noise,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 765–780.

[6] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, “Robust face recognition via adaptive sparse representation,” *IEEE transactions on cybernetics*, vol. 44, no. 12, pp. 2368–2378, 2014.

[7] W. Ou, X. You, D. Tao, P. Zhang, Y. Tang, and Z. Zhu, “Robust face recognition via occlusion dictionary learning,” *Pattern Recognition*, vol. 47, no. 4, pp. 1559–1572, 2014.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, “Imagenet large scale visual recognition challenge,” *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

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

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

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

[12] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[13] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, 2016, p. 507–516.

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

[15] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.

[16] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699.

[17] I. Kim, S. Han, S.-J. Park, J.-W. Baek, J. Shin, J.-J. Han, and C. Choi, "Discface: Minimum discrepancy learning for deep face recognition," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

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

[20] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision.* Springer, 2016, pp. 499–515.

[21] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6398–6407.

[22] D. Zeng, R. Veldhuis, and L. Spreeuwers, "A survey of face recognition techniques under occlusion," *IET biometrics*, vol. 10, no. 6, pp. 581–606, 2021.

[23] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

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

[25] H. Qiu, D. Gong, Z. Li, W. Liu, and D. Tao, "End2end occluded face recognition by masking corrupted features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[26] F. Zhao, J. Feng, J. Zhao, W. Yang, and S. Yan, "Robust lstm-autoencoders for face de-occlusion in the wild," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 778–790, 2017.

[27] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

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

[29] Z. Zhu, G. Huang, J. Deng, Y. Ye, J. Huang, X. Chen, J. Zhu, T. Yang, J. Lu, D. 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*, 2021, pp. 10 492–10 502.

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

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

[32] J. Wang, Y. Liu, Y. Hu, H. Shi, and T. Mei, “Facex-zoo: A pytorch toolbox for face recognition,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3779–3782.

[33] A. Anwar and A. Raychowdhury, “Masked face recognition for secure authentication,” 2020.

[34] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, vol. 1. Ieee, 2001, pp. I–I.

[35] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.