Masked Face Recognition with deep learning
Project Milestone

Tsui Ka Hei
khtsuiac@connect.ust.hk
20599112

NG Chun Fung
cfngai@connect.ust.hk
20696453

1. Abstract

Wearing masks has become the new normal during the Covid-19 pandemic. However, this also makes facial recognition, a widely used biometrics authentication, very difficult with important features like mouth and nose hidden. In this project, we want to propose a method to conduct masked face recognition. The first step is to remove the face region occluded by the mask. Next, we will apply three pre-trained Convolutional Neural Networks (CNN), that are FaceNet, VGGFace, and AlexNet for face embedding with ArcFace as loss function. The embedding will passed into two classifiers, namely the K-nearest-neighbors classifier (KNN), and the Multilayer Perceptron classifier (MLP). We will experiment using the Real World Masked Face Dataset and evaluate their performance. Our project successfully implements a face recognition model with 80.33% accuracy.

2. Introduction

Covid-19 is one of the most severe pandemics in recent history, causing disruption to the economy and deaths. Wearing masks has been proven to be an effective way to prevent airborne transmission of the disease, so many governments including Hong Kong has required their citizen to wear masks during the outbreak. The virus can also be spread through human-to-human contact and contaminated surfaces; Therefore, the contactless nature of face recognition makes it safer compared to traditional biometrics authentications such as fingerprints. However, wearing masks might reduce the performance of face recognition systems, because a large area of the face including the noses and lips are covered by the masks. Many important features cannot be extracted for recognition. Therefore, it is important to develop a masked face recognition system during the pandemic. Our project adopts an occlusion removal approach. We experiment with different models and our best model using InceptionResNetV1, ArcFace and MLP achieved an 80.33% accuracy.

3. Related Work

Occlusion has always been a major challenge for real-world face recognition. Objects including hats, eyeglasses, hands, and masks can block some parts of the face. Masks, however, are one of the most challenging occlusion among them as it occludes a big part of the face, including key features such as the nose, mouth, and chin. Researchers have proposed many approaches to this problem and those can be generalized into three categories.

Restoration method: The restoration method aims to restore the unoccluded face from the occluded face. Then we can use the restored face and apply conventional face recognition. . However, the quality of the restoration is very important, but that is often unpredictable. [8] proposed to Principal Component Analysis (PCA) to reconstruct the eyes of people wearing glasses. This paper [17], use robust LSTM-Autoencoders (RLA) to restore occluded face based on spatial and temporal characteristics. The author of [2] uses a recent image inpainting model based on a Generative Adversarial Network (GAN) to restore the face. Fig 1 presents some restored faces of the research.

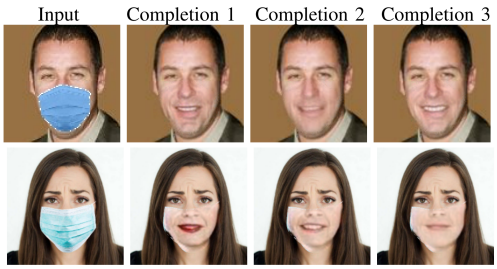

Figure 1. Restored face based on GAN

Matching method: This method aims to extract features from the unoccluded areas and then use them to compare similarities between faces. This paper [6] sampled the faces into patches and extract features from each patch, and match them by similarity. The author of [12] use statistical learning including the Scale Invariant Feature Transform (SIFT)

descriptor to extract the feature.

Occlusion removal method: This method aims to discard the area that is covered by the mask and use the remainder for feature extraction and matching process. [1] identify the facial area as an occluded area with a trained improved SVM classifier. [13] used a CNN model to learn the correspondence between occluded facial areas and corrupted feature elements, which is named Feature Discarding Mask (FDM) between the occluded area and corrupted features.

4. Dataset to be used

Real World Masked Face Dataset [14] (RMFD) is a dataset with masked face images to improve the performance of the existing face recognition technology for masked faces. It consists of three types of datasets, namely the Masked Face Detection Dataset (MFDD), the Real-world Masked Face Recognition Dataset (RMFRD), and Simulated Masked Face Recognition Dataset (SMFRD). In our project, we will focus on RMFRD and SMFRD.

a) RMFRD is one of the largest real-world masked face datasets in the world. It includes 5,000 pictures of 525 people with their masks on, and 90,000 images of the same 525 subjects without masks. Fig. 2 presents some pairs of facial image samples from the dataset.

Figure 2. Examples of a pair of face images. (a) and (b) are face images without mask. (c) and (d) are masked face images.

b) SMFRD contains 500,000 simulated masked faces of 10,000 subjects from existing public large-scale face datasets, including Labeled Faces in the Wild (LFW) [3], AgeDB-30 [7], Celebrities in Frontal-Profile in the Wild [10] and Webface [16]. The simulation is developed using Dlib library. We selectes the dataset that is generated from Webface. Fig. 3 presents some simulated facial image samples from the dataset.

Figure 3. Examples of some simulated facial image

In our progress report, we mentioned the images in the dataset are already cropped around the faces, so we will not do face detection to crop out the face.

We also discarded the masked area and only use the area without the mask (i.e. area with the eyes and forehead). In practice, we normalized all face images into a standard dimension. Then, we will discard the lower half of the picture (i.e height / 2), that mostly occluded by the mask. Fig 6 shows an example of a face after cropping.

Figure 4. Examples of cropped face

5. Method

5.1. Occlusion removal approach

We attempted an occlusion removal approach as it seems to achieve more reliable results in our literature review, compared with the restoration approach which relies on unpredictable restored images for face recognition. The details of the occlusion removal are explained in Section 4. However, we discovered that such a simple removal method may not be that reliable and we also compare the result for a model without discarding the lower half.

5.2. Face embedding

The pre-processed images are fed into DCNN for face embedding. To allow more efficient training, we used transfer learning. We have selected the following pre-trained models that are well-trained for non-masked face recognition for experimentation.

5.2.1 FaceNet

[11] ia a face recognition model developed by Google. It achieved more than 95% accuracy when performing unmasked face recognition. They uses a deep convolutional network with architectures such as ZF-Net and Inception Network. It also proposed triplet loss as a loss function for training. Fig. 6 presents the architecture of FaceNet. We found "facenet-pytorch", a python package, based on InceptionRes-NetV1, pre-trained on two datasets, VGGFace2 [9] and CASIA-Webface [16]. This pre-trained network can achieve a 99.65% accuracy when tested with a face image dataset called Labelled Face in the Wild. We used PyTorch for training this pre-trained model with our own dataset. We note that the output size is 512 in the last

layer, and we use the output as a 512-dimension feature vector that contains the feature of the face.

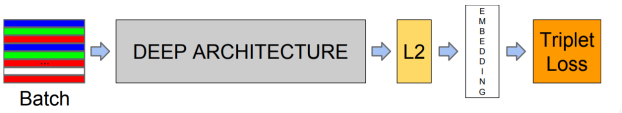

Figure 5. Overview of the FaceNet's Architecture

5.2.2 VGGface

[9] is a face recognition model developed by researchers from the University of Oxford. They created a very large face dataset with 2.6 million images. They also use a deep convolutional network and train the network with the newly created dataset. It achieved more than 97% accuracy when performing unmasked face recognition. We used the source code and model published by Visual Geometry Group for their paper. The model is pre-trained on Labeled Faces in the Wild (LFW) [3] and the YouTube Faces [15] dataset. Fig. 7 presents the architecture of VGGface.

5.2.3 AlexNet

[4] is a successful image classification model introduced in Lecture 9 of our class. This model is pre-trained based on millions of images from ImageNet. Its architecture is shallower than FaceNet [11] and VGGFace [9]. We would like to experiment with its compatibility to face recognition tasks. Fig. 8 presents the architecture of AlexNet. We found that AlexNet is included in the model zoo of PyTorch. We used the "torchvision.models" subpackage for simplicity. In our experiment, we extract The fifth layer is used as a deep feature for classification.

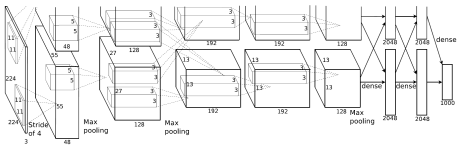

Figure 7. Overview of the AlexNet's Architecture

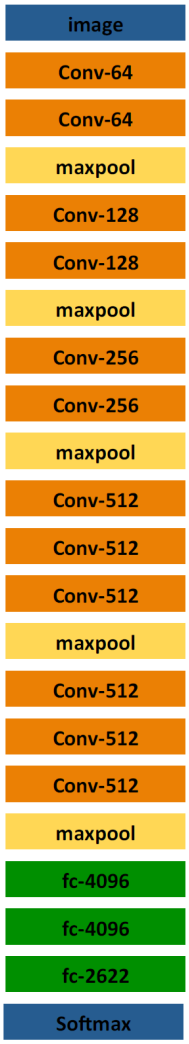

Figure 6. Overview of the VGGface's Architecture

$$L_7 = -\log \frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{j=1,j\neq y_i}^{N} e^{s\cos\theta_j}},$$
$$\text{where } \theta_j = arccos\left(\max_k \left(W_{jk}^T \mathbf{x}_i\right)\right), k \in \{1, \cdots, K\}.$$

Figure 8. ArcFace loss formula

5.3. Loss function

In our progress report, we proposed to experiment with the triplet loss function and ArcFace function. However, given the proposed numbers of networks, loss function, and classifiers, we have to experiment 3 x 2 x 3 = 18 different networks and we might not have enough time for such implementation. With that in mind, We will only experiment with ArcFace which is published more recent;y and can produce better result (99.53%) than triplet loss (98.98%) based on LFW. We therefore only use ArcFace for the computation of loss and for optimization during training. Fig 9 below shows the ArcFace loss formula.

5.4. Classification

The embedded vector will pass through a classification layer. In our progress report, we proposed to experiment with linear discriminant analysis (LDA), K-nearest-neighbors classifier (KNN), Multilayer Perceptron classifier (MLP) and see which model did the best job. Because of the

time constraint, we only experiment with KNN and MLP which we are more familiar with.

5.5. Evaluation

To evaluate the performance of our models, we will separate the dataset into training, validation set, and testing set in the ratio of 7:1:2 (as recommended in COMP4471 Lecture 2). The validation and test set will contain pairs of images from the same person, called a "same-person pair" or from different people, called a "different-person pair". We will measure the following metrics:

• True positive: correctly classified a "same-person pair"

• False positive: a "different-person pair" wrongly classified as "same-person pair"

• True negative: correctly classified a "different-person pair"

• False negative: a "same-person pair" wrongly classified as "different-person pair"

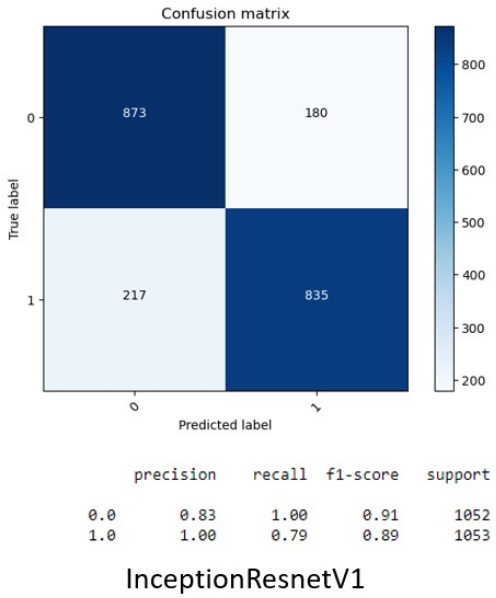

Figure 9. confusion matrix for InceptionResNetV1, arcface and MLP

6. Experimentation and discussion

For the training and testing of the models in this project, we used a local machine with an NVIDIA GeForce RTX3060.

6.1. Model experimentation

We experiment with different combinations of architectures, loss functions and classifiers. The best model is obtained with InceptionResNetV1, ArcFace and MLP with 80.33% accuracy. Fig. 9 and Fig. 10 below showed the confusion matrix shows confusion matrix with different combinations.

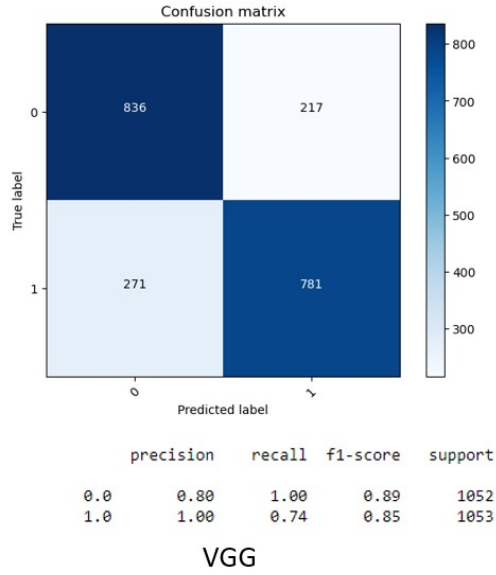

Figure 10. confusion matrix for VGGFace, arcface and MLP

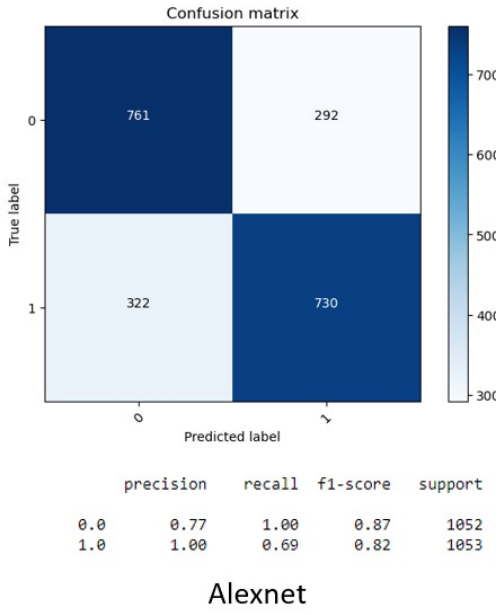

Figure 11. confusion matrix for AlexNet, arcface and MLP

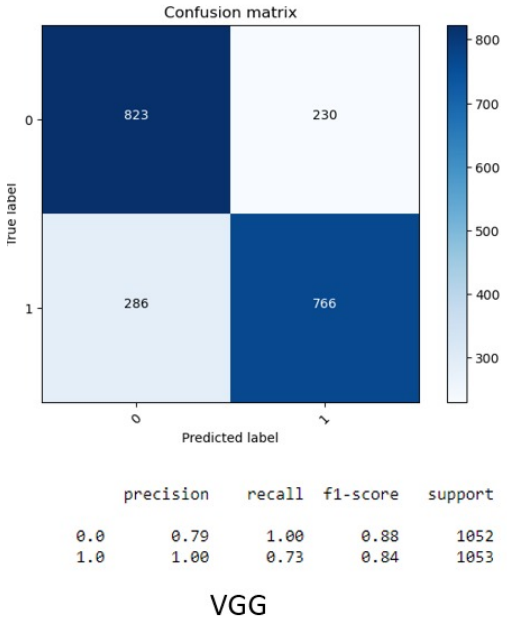

Figure 13. confusion matrix for VGGFace, arcface and KNN

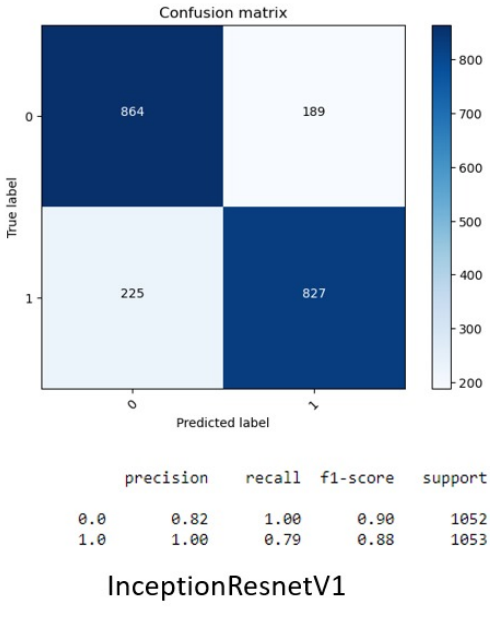

Figure 12. confusion matrix for InceptionResNetV1, arcface and KNN

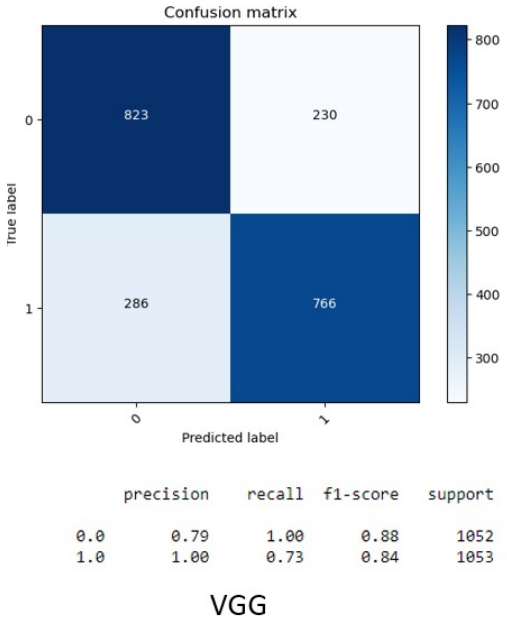

Figure 14. confusion matrix for AlexNet, arcface and KNN

6.2. Failed attempt to remove other occlusion

When we look at the incorrect classfied data, we discovered the data include other types of occlusion (e.g. hats, hands, glasses), making the face completely blocked. Fig 15 shows some examples of completely occluded faces.

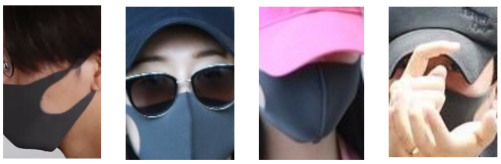

Figure 15. Examples of image with other occlusion

Our thought is to conduct face detection. If a face is completely occluded, it will not be detected as face. We tried to use MediaPipe [5], an open-sourced ML solution to conduct face detection. However, it seems to be unable to detect faces, even if not completely occluded in our dataset. Fig 16 demonstrates our attempted result using RMFRD for face detection. On top of the mask occlusion affecting the accuracy, we observed our dataset already has the face cropped, i.e. the images have conducted face detection before. This may be the reason for our failure as images with a background seem to be more successful in face detection. Fig 17 shows a test of us using a masked face with a background that was not in our dataset.

	Face Detected	Face Not detected
MediaPipe	258	8741

Figure 16. attempted result of face detection with RMFRD

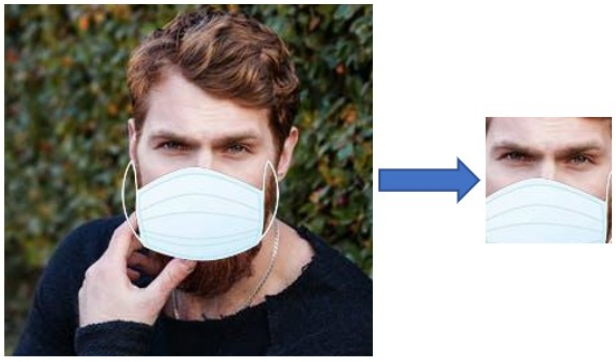

Figure 17. face detection using masked face with background

6.3. Failed case for occlusion removal

We also discovered our occlusion removal method (i.e. simply discarding the lower half) may sometime discard important features. This is because the faces in our dataset are oriented in a different way such that their eyes may be in the lower half of the image. Fig 18 shows an example of a badly cropped face where his eyes is cropped away.

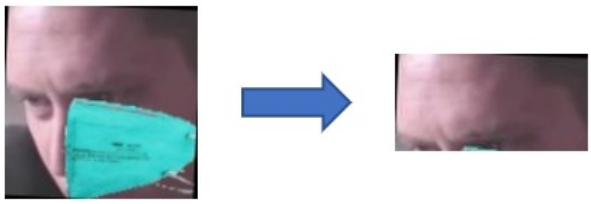

Figure 18. Comparison of cropped model and the uncropped model

It is questionable whether our method to remove mask enhance the quality of face recognition. We, therefore, experiment to retrain our best model with uncropped faces. It shows that the accuracy is similar or even slightly better. Our speculation is that the uncropped model learns to extract features from the upper half of the image while avoiding the accidental removal of features. Fig 19 shows a comparison of cropped model and the uncropped model.

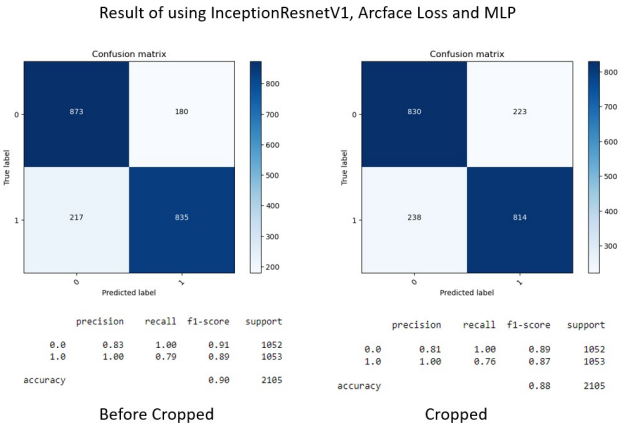

Figure 19. Badly cropped face

6.4. Conclusion/Future Work

For our project, we have experimented with three pretrained models (InceptionResNetV1, VGGFace, AlexNet), with ArcFace as loss function and two classifiers (KNN, MLP) for masked face recognition. The best model is obtained with InceptionResNetV1, ArcFace and MLP with 80.33% accuracy. We also have a failed attempt to remove other kinds of occlusions with face detection. Last but not least, we learnt that our simple cropping method to remove occlusion may not be that helpful in increasing accuracy. A more sophisticated way that removes the mask but retains the non-masked region is necessary.

For future extensions, we might try other open-source solutions such as OpenCV for occlusion detection and remove the images with other kinds of occlusion (e.g. glasses) to enhance accuracy. For a better way to remove masks, We

can do mask detection and identify areas that are covered by masks, and remove those areas only for better accuracy

References

[1] Zhaohua Chen, Tingrong Xu, and Zhiyuan Han. Occluded face recognition based on the improved svm and block weighted lbp. In *2011 International Conference on Image Analysis and Signal Processing*, pages 118–122. IEEE, 2011. 2

[2] Md Imran Hosen and Md Baharul Islam. Himfr: A hybrid masked face recognition through face inpainting, 2022. 1

[3] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 2, 3

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 3

[5] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. Mediapipe: A framework for building perception pipelines. *CoRR*, abs/1906.08172, 2019. 6

[6] A.M. Martinez. Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):748–763, 2002. 1

[7] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. Agedb: the first manually collected, in-the-wild age database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, volume 2, page 5, 2017. 2

[8] Jeong-Seon Park, You Hwa Oh, Sang Chul Ahn, and Seong-Whan Lee. Glasses removal from facial image using recursive error compensation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):805–811, 2005. 1

[9] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *British Machine Vision Conference*, 2015. 2, 3

[10] C.D. Castillo V.M. Patel R. Chellappa D.W. Jacobs S. Sengupta, J.C. Cheng. Frontal to profile face verification in the wild. In *IEEE Conference on Applications of Computer Vision*, February 2016. 2

[11] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. 2, 3

[12] Jeongin Seo and Hyeyoung Park. A robust face recognition through statistical learning of local features. In *Neural Information Processing*, Lecture Notes in Computer Science, pages 335–341, Berlin, Heidelberg. Springer Berlin Heidelberg. 1

[13] Lingxue Song, Dihong Gong, Zhifeng Li, Changsong Liu, and Wei Liu. Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In

2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 773–782, 2019. 2

[14] Zhongyuan Wang, Guangcheng Wang, Baojin Huang, Zhangyang Xiong, Qi Hong, Hao Wu, Peng Yi, Kui Jiang, Nanxi Wang, Yingjiao Pei, Heling Chen, Yu Miao, Zhibing Huang, and Jinbi Liang. Masked face recognition dataset and application, 2020. 2

[15] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534, 2011. 3

[16] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. Learning face representation from scratch, 2014. 2

[17] Fang Zhao, Jiashi Feng, Jian Zhao, Wenhan Yang, and Shuicheng Yan. Robust lstm-autoencoders for face de-occlusion in the wild. *IEEE Transactions on Image Processing*, 27(2):778–790, 2018. 1