Masked Face Recognition Using Convolutional Neural Networks and Similarity Analysis

Mobina Mobaraki, Mohamed Zidan, Hamid Reza Tohidypour, Yixiao Wang, Rui Zhong, Haoxiang Lei, Panos Nasiopoulos
Electrical & Computer Engineering, Univeristy of British Columbia
Vancouver, BC, Canada
{mobinamb, mzidan02}@student.ubc.ca, {htohidyp, yixiaow}@ece.ubc.ca, rzhong6819@gmail.com, hxlei@student.ubc.ca, panosn@ece.ubc.ca

***Abstract*— Nowadays, human face recognition systems have been widely used in different applications in which identity recognition is needed. The performance of current face recognition algorithms is negatively affected by occlusions, such as facial masks and various human poses. To address these challenges, we re-trained a modified version of the VGG19 deep learning model on masked and unmasked images of 62 identities to design a feature extractor that extracts deep features from the non-occluded areas of the face. This feature extractor is combined with our proposed similarity analysis network that is trained on our dataset to automatically judge whether the masked and unmasked images correspond to the same or different identities. Our final approach consists of a feature extractor from a fine-tuned VGG19 and a similarity model. It achieved an accuracy of 80 to 85 percent in recognizing the identity of test masked images with different poses.**

Keywords—— Masked Face recognition, COVID-19 masks, CNN, Feature extraction, Similarity analysis

I. INTRODUCTION

One of the most important applications of face recognition systems is to identify a masked face. Robust face recognition approaches may be used to unlock smart phones or your new car while wearing masks in public or determine if a person is following COVID-19 regulations by wearing a mask.

Current face detection models mainly extract a set of important features to learn the key facial attributes, such as eyes, nose, and mouth, and use these features to identify the face in each image. They are mainly classified into shallow representation and deep representation methods.

The former is a conventional method which uses Ada-boost algorithms [1] to extract the hand-crafted features, such as Local Binary Pattern (LBP) [2], Haar-like, scale-invariant feature transform (SIFT) [3], and Gabor [4]. These classical methods are divided into single [5] (with "Masked" and "No Masked" as labels) and multiple detectors (with "Masked", "No Masked", and "improper Masked" as labels). However, due to the shallow features in these methods, they fail to perform accurately in the presence of occlusions.

The latter is a deep learning-based method which usually uses Convolutional Neural Networks (CNNs) to extract deep features. The common algorithms are Faster Region-based Convolutional Neural Network (R-CNN), YOLO-based methods [6], and Retina Face mask-based methods [7]. Faster R-CNN is a good choice when better performance and accuracy are needed, while the YOLO-based methods are more suitable for real-time applications.

The COVID-19 pandemic and the importance of wearing face masks in public areas have made face recognition crucial for masked faces. However, some of the existing face recognition systems do not include masked faces in their dataset, while training [2,3]. Others such as [4] performed masked face recognition on 520 images of 20 identities, but they are limited to the frontal view to achieve a high accuracy (95%). The method proposed in [5] achieved an accuracy of 85.16% by considering the side views in 1274 images of 50 identities. Other recent studies could only determine if a person wears a mask and if the mask is properly used, but they cannot determine the identity of a person with a mask [7-14].

In this paper, we introduce a robust masked face identification which addresses the shortcomings of the previous studies by using a deep learning model based on convolutional layers that extracts deep features from the non-occluded areas of a masked face from different angles. To this end, VGG19 model is re-trained on masked and unmasked images [15]. The feature extractor part is combined with our proposed similarity analysis model to identify masked faces. The final model takes a masked image of a person as input and compares it with all the unmasked images in the dataset. Then, it outputs the identity of the most similar unmasked image as the person in the image. Performance evaluations demonstrated that our method achieves an accuracy of 80 to 85%.

II. DATASETS

Our proposed face detection consists of feature extractor and similarity analysis parts. We generated two training subsets from the LFW dataset [16]: one for training the feature extractor and the other for developing the similarity analysis model. We also used random subsets that consist of unseen face images of the LFW [16] and RMFD [17] datasets to test our model.

A. Feature Extractor Training Dataset

When we try to recognize the identities of masked face images, the visible parts of the faces play the most important

979-8-3503-3959-8/23/$31.00 ©2023 IEEE"

role. This means a robust face detector requires a feature extractor to focus on the visible parts of the masked faces. Therefore, this study considers a mixed dataset that consists of unmasked and masked face images to train and validate the feature extractor network so that it teaches the feature extractor to learn key visible features of the face and assign higher weights to them. Our feature extractor dataset has 62 identities. Each identity has 16 masked and 16 unmasked images for training and 4 masked and 4 unmasked images for validation. In total, this dataset has 1984 images. The masked face images were generated by the software proposed in [18] that aligns 106 landmarks to add a mask on a face as shown in Fig.1.

B. Similarity Analysis Training Dataset

The full dataset has 38 identities. Each identity has 7 unmasked and 7 masked images. The removed dataset has the same number of identities while each identity only keeps two masked images and their unmasked versions are removed. So, each identity has 2 masked images and 5 unmasked images. The reason for using the removed dataset is to generalize the model so that it can identify a masked face even if the exact unmasked version is not included in the dataset.

III. OUR PROPOSED METHOD

Our goal is to design a robust face recognition approach that can detect the identity of masked faces. In this section, we describe the overall architecture of our approach and its two main components, including the feature extractor and the similarity analysis model.

A. Overall Architecture

Our final model consists of two parts: 1) the feature extractor which learns how to extract required information from the images to identify the person. As a feature extractor, we trained VGG19 model on the feature extractor training dataset. The details of this part including the structure, modifications, training specifications, results, and implementation details, are discussed in the feature extractor subsection. 2) the similarity analysis part which judges if the feature vectors (the output of the feature extractor part) of a masked image and an unmasked image correspond to the same or different identities. The details of our two similarity analysis approaches and their results are discussed in the similarity analysis subsection. Finally, the feature vector of a given masked image is compared with all the unmasked images and our method reports the most probable identity as the identity of the masked person. Fig. 2 shows our high-level masked face recognition architecture.

Fig. 1. Example of putting mask on unmasked images.

B. Feature Extractor

We selected VGG19 network as the feature extractor which was originally trained on the ImageNet dataset. To teach the feature extractor concentrate on the features of a face, we retrained it on a face dataset [16] using transfer learning. The last layer of VGG19 is a fully connected (FC) layer that aggregates the features that are obtained from the previous layer into scores of the classes. Since the pre-trained VGG19 has originally 1000 classes, we changed the last FC layer of VGG19 to output 62 scores, which is the number of identities in our dataset. In order to retrain the VGG19, we used our feature extractor dataset that consists of unmasked and masked face images. This allows our model to focus on face features outside the masked face region, such as eyes and forehead.

For the training phase, the images were normalized. To avoid overfitting, the training images were randomly cropped and horizontally flipped to augment the data. For the validation phase, the center crop approach was used as it is common for the ImageNet based networks. The input images were resized to 224x224 to become adapted to VGG19's network design.

The training was carried out by optimizing the cross-entropy loss function using the mini-batch Adam algorithm on an advanced computer cluster. The batch size was set to 32. The learning rate was set to 1e-4 and the weight decay was set to 0.5. The learning rate was divided into half, every time the loss vs epoch curve was plateaued. The training was also regularized by dropout regularization for the first two-fully connected layers where the dropout ratio was set to 0.5. We trained our model for 250 epochs, since no improvement was observed on loss vs epoch curve after 250 epochs. Our retrained model achieved the accuracy of 91% on the validation data of our feature extractor dataset. In order to transform our VGG19 model into a feature extractor, we considered two scenarios:1) using the entire model as a feature extractor, 2) removing the last FC layer (FC3) and using the resulting model as a feature extractor.

C. Similarity Analysis

Given our feature extractor, the feature vector of each image can be calculated. For similarity analysis, the feature vector of two images (masked and unmasked) has to be compared using a reliable approach to determine if two images belong to the same or different identities. In the quest to find the best comparison approach, we proposed two approaches, namely, the threshold-based approach and the automatic approach. The details of each approach are as follows:

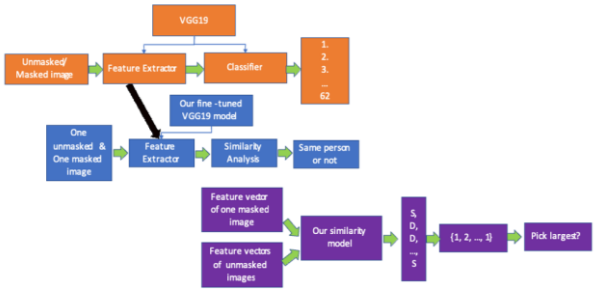

Fig. 2. High-level masked face recognition architecture.

1) Threshold-based approach: To define a threshold value which separates the feature vectors that belong to the same and different identities, first, the features are extracted using the feature extractor from the two face images that are supposed to be compared. Then, three similarity functions (Mean Absolute Error (MAE), Dot Product (Dot), and Normalized Dot Product (Normalized Dot)) were considered to find the best similarity function for our application. The Normalized Dot is obtained by normalizing the feature vectors before applying the dot product operator.

$f_1 = [x_1, x_2, ..., x_{4096}]$ and $f_2 = [y_1, y_2, ..., y_{4096}]$ are the feature vectors of one masked and one unmasked image in the dataset and $f'_1 = [x'_1, x'_2, ..., x'_{4096}]$ and $f'_2 = [y'_1, y'_2, ..., y'_{4096}]$ are the normalized feature vectors of the images by dividing each component of the feature vectors by the magnitude of the vector. The MAE, Dot, and Normalized Dot can be formulated as follows:

$$MAE = (|y_1 - x_1| + \ldots + |y_{4096} - x_{4096}|)/|f_1| \quad (1)$$
$$Dot = x_1 \cdot y_1 + \ldots + x_{4096} \cdot y_{4096}$$
$$\text{Normalized } Dot = x'_1 \cdot y'_1 + \ldots + x'_{4096} \cdot y'_{4096}$$

The similarity values obtained from each of the similarity functions have to be compared with an appropriate threshold value to determine if two face images belong to the same identity. For the case of MAE, if the MAE similarity value is less than the MAE threshold, the two feature vectors correspond to the same identity, otherwise they are related to different identities. For the case of the Dot/Normalized Dot, if the value of Dot/Normalized Dot similarity is greater than the Dot/Normalized Dot threshold, the two feature vectors correspond to the same identity, otherwise they are related to different identities.

In order to calculate the threshold value for each similarity function, the similarity values between every masked and unmasked image of the same identity in the training (similarity) dataset are calculated and averaged to find a threshold value.

Although the threshold approach can potentially work, it relies on a simple averaging function that may not be able to generalize well in real scenarios, in which the pose of a person's masked face image look different from the unmasked face images of the same person. To address this issue, we propose an automatic similarity detection approach that is explained in the following subsection.

2) Automatic similarity detection approach (weighting model): In this approach, the L1 norm between the normalized version of the two feature vectors is calculated and fed into a classification model. Here, to use the retrained VGG19 as a feature extractor, its last fully connected layer was removed. The classification (weighting) network, which receives the feature vector from the feature extractor, consists of three linear layers with Rectified Linear Unit (ReLU) as the activation function, two batch normalization (BN) layers, and a Sigmoid layer as a classifier layer. Fig. 3 shows the architecture of our proposed weighting network. The model is trained to find the optimal weights of each component of the L1 norm vector so the components around the non-occluded areas of the face should have greater weights. The learning rate, number of epochs and batch size are set to 0.001, 20 and 64, respectively. We used Adam as the optimizer function and the Binary Cross Entropy loss as a loss function. The output of the model is either "same" or "different".

The final model is a combination of the feature extractor and the weighting model which determines if the two given masked and unmasked images correspond to the same or different identities. It compares the masked image with all the unmasked images in the dataset and takes the most probable identity as the identity of the given masked image.

IV. RESULTS AND DISCUSSION

In order to find the accuracy of the threshold-based similarity functions we considered two conditions shown in Fig. 4, where the orange line is the horizontal axis, A and B are the masked images, A' and B' are the unmasked images of the first identity. B and C are the masked images, and B' and C' are the masked images of the second identity:

First condition: the MAE value between the masked image and at least one of the unmasked images with the same identity should be less than the MAE threshold (for the case of Dot/ Normalized Dot values should be greater than their corresponding threshold). This happens when the minimum of the MAE values between the current masked image and all the unmasked images of the same identity as the current masked image (green area in Fig. 4) is less than the MAE threshold (or the maximum of Dot/Normalized Dot values should be greater than their corresponding thresholds).

Second condition: the MAE value between the masked image and all the unmasked images with different identities than

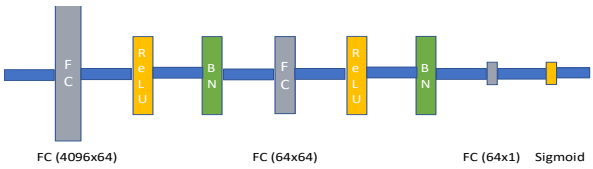

Fig. 3. Architecture of our proposed similarity detection network.

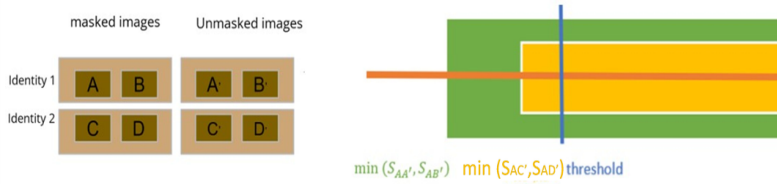

Fig. 4. Conditions to make a true positive prediction by comparing the error with the MAE threshold.

the masked image should be greater than the minimum MAE value between the masked and unmasked images of the same identity. For the Dot/Normalized Dot approach, the maximum values of Dot/Normalized Dot for different identities should be less than the maximum Dot/ Normalized Dot values between the same identities. In the case that several identities can fulfill condition 1, the final prediction for the MAE is the identity with minimum MAE value. If the predicted identity is the expected one, we consider it as a correct prediction. This happens when the minimum of the yellow area (the MAE between the current masked image and all the unmasked images with identities different than the current masked image) in Fig. 3 is greater than the minimum of the green area. For the Dot/Normalized Dot approach the predicted identity is the one with maximum Dot/Normalized Dot value. If the two conditions (1 and 2) are met, the threshold approach predicts the identity of the masked person correctly, otherwise it is considered as wrong prediction.

Table I shows the final accuracy based on the threshold approach for the three similarity functions and the two feature extractors (with the last FC layer and without the last FC). The results are for the full dataset. We observe that for the case of removing FC3, the accuracy is higher than the case of using the full VGG19 model as feature extractor. In addition, the MAE similarity function has the best performance followed by Normalized Dot and the Dot.

Although the accuracy of the MAE and Normalized Dot were above 96.99% for the full dataset, the performance dropped to 15% for the removed dataset in which the unmasked version of the masked images does not exist in the dataset. This means the model may not work properly when the masked image of the person is not exactly the same as his/her unmasked image in the dataset due to the change in the pose, clothes, background, etc.

The problem is that the average of the similarity values would not be a good representative threshold value to separate the same and different identities based on the two following reasons: 1) The histogram of the error (MAE) values does not have a normal distribution, so the mean value is not the same as the peak of the histogram which is the most frequent error value. 2) Each component of the feature vector may have different effect on the final decision which is not considered in the threshold-based similarity approach

Finally, we examined the performance of our automatic (weighting) method on the LFW [16] test dataset consisting of 20 identities with two masked images and five unmasked images for each identity. We also evaluated the generalization ability of this method using six random cross-validation subsets of the RMFD [17] dataset. These are all larger than the LFW [16] test dataset. The results are reported in Table II. We observe that for LFW [16] test data, our model achieved 100% accuracy. In addition, for the random subset data, our model can identify masked faces with 80 to 85% accuracy and standard deviation of 1.66.

TABLE I. FINALACCURACY BASED ON THE MANUAL SIMILARITY ANALYSIS MODEL

Similarity function	Accuracy without FC (%)	Accuracy with FC (%)
MAE	97.74	75.56
Dot	13.50	3.00
Normalized Dot	96.99	90.22

TABLE II. FINAL ACCURACY OF OUR MODEL MEASURED ON OUR TEST CROSS-VALIDATION DATASET

Test dataset	Masked images	Unmasked images	IDs	Total number	Accuracy (%)
LFW	40	100	20	140	100
Random subset1	124	1150	40	1274	82.2
Random subset2	124	1150	40	1274	85.16
Random subset3	132	410	50	542	83.88
Random subset4	132	410	50	542	84.86
Random subset5	132	410	50	542	80.36
Random subset6	193	720	50	913	84.11

V. CONCLUSION

In this paper, we proposed a novel masked face recognition scheme using a convolutional neural network. First, we created a dataset that contains masked and unmasked face identities. Using this dataset, we trained a VGG-based model to create a feature extractor for face images. We examined three similarity functions including mean absolute error (MAE), Dot Product, and Normalized Dot Product to determine the identity of the masked image by comparing the features of the masked image and unmasked images with known identities. Although the MAE and Normalized Dot achieved the accuracy of equal and/or greater than 96.99% for the case that the unmasked version of the examined masked image existing in the dataset, they did not achieve a promising accuracy for the case that the unmasked face version did not exist in the test dataset. To address this challenge, we designed a neural network (weighting model) that compares the features of the masked and unmasked images to automatically determine if they belong to the same identity. Our final approach consists of the feature extractor part from the re-trained VGG19 and the weighting model and is capable of predicting the identity of the given masked image with 80 to 85 % accuracy on 6 random cross-validation subsets from RMFD [17] dataset.

ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC–PG 11R12450), and TELUS (PG 11R10321). This research was enabled in part by support provided by Digital Research Alliance of Canada (https://alliancecan.ca/en).

REFERENCES

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features." Proceedings of the 2001 IEEE computer society conference on computer

vision and pattern recognition. CVPR 2001, pp. I-I, 2001, doi: 10.1109/CVPR.2001.990517.

[2] T. Ahonen, A. Hadid and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, Dec. 2006, doi: 10.1109/TPAMI.2006.244.

[3] C. Geng and X. Jiang, "Face recognition using sift features," 2009 16th IEEE International Conference on Image Processing (ICIP), pp. 3313-3316, 2009, doi: 10.1109/ICIP.2009.5413956.

[4] B. S. Bayu Dewantara and D. Twinda Rhamadhaningrum, "Detecting Multi-Pose Masked Face Using Adaptive Boosting and Cascade Classifier," 2020 International Electronics Symposium (IES), pp. 436-441, 2020, doi: 10.1109/IES50839.2020.9231934.

[5] I. Cheheb, N. Al-Maadeed, S. Al-Madeed, A. Bouridane and R. Jiang, "Random sampling for patch-based face recognition," 2017 5th International Workshop on Biometrics and Forensics (IWBF), pp. 1-5, 2017, doi: 10.1109/IWBF.2017.7935104.

[6] V. Sharma, "Face mask detection using yolov5 for COVID-19", Diss. California State University San Marcos, 2020.

[7] M. Jiang, X. Fan, and H. Yan, "Retinamask: A face mask detector." arXiv preprint arXiv:2005.03950, 2020.

[8] A. Alguzo, A. Alzu'bi and F. Albalas, "Masked Face Detection using Multi-Graph Convolutional Networks", 2021 12th International Conference on Information and Communication Systems (ICICS), pp. 385-391, 2021, doi: 10.1109/ICICS52457.2021.9464553.

[9] J. Zhang, F. Han, Y. Chun and W. Chen, "A Novel Detection Framework About Conditions of Wearing Face Mask for Helping Control the Spread of COVID-19," in IEEE Access, vol. 9, pp. 42975-42984, 2021, doi: 10.1109/ACCESS.2021.3066538.

[10] J. Wang, Y. Yuan, and G. Yu, "Face attention network: An effective face detector for the occluded faces." arXiv preprint arXiv:1711.07246, 2017.

[11] S. Ge, J. Li, Q. Ye and Z. Luo, "Detecting Masked Faces in the Wild with LLE-CNNs," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 426-434, 2017, doi: 10.1109/CVPR.2017.53.

[12] S. Lin, L. Cai, X. Lin, and R. Ji "Masked face detection via a modified LeNet." Neurocomputing 218, pp. 197-202, 2016.

[13] Y. Chen et al., "Face Mask Assistant: Detection of Face Mask Service Stage Based on Mobile Phone," in IEEE Sensors Journal, vol. 21, no. 9, pp. 11084-11093, 2021, doi: 10.1109/JSEN.2021.3061178.

[14] X. Fan, M. Jiang and H. Yan, "A Deep Learning Based Light-Weight Face Mask DetectorWith Residual Context Attention and Gaussian Heatmap to Fight Against COVID-19," in IEEE Access, vol. 9, pp. 96964-96974, 2021, doi: 10.1109/ACCESS.2021.3095191.

[15] M. Shaha and M. Pawar, "Transfer Learning for Image Classification", 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 656-660, 2018, doi: 10.1109/ICECA.2018.8474802.

[16] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments", pp. 7-49, 2007, [Online]. Available: http://vis-www.cs.umass.edu/lfw/

[17] Real-World-Masked-Face-Dataset, [Online]. Available: https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset.git

[18] FMA-3D, face mask adding, [Online]. Available: https://github.com/JDAI-CV/FaceX-Zoo/tree/main/addition_module/face_mask_adding/FMA-3D. Accessed: August 2022.