Building An Accurate Detector for Masked Faces in the Pandemic Era

Zishen Li
li.zis@northeastern.edu

Junmei Luo
luo.jun@northeastern.edu

Tony Yukuang Zhang
zhang.yuk@northeastern.edu

Github:
https://github.com/Tonyz4516/detector-for-masked-faces

1 SUMMARY

Since we have seen the first Covid case in the U.S. back in March, it has already been eight months. In a time of deep uncertainty, people learned to cope with coronavirus by wearing masks, washing hands, and keeping social distances. However, with cooler weather and pandemic fatigue, we have seen spikes in both cases and hospitalization worldwide.

To help society to mitigate the spread of Covid-19 and to raise people's awareness of the importance of wearing masks, we proposed a method of using an algorithm at the entrances of public places, such as markets, grocery stores, and offices, to monitor whether people wear masks. The algorithm is also designed to remind them if they did not follow the experts' mask-wearing guidelines, that is, not wear masks correctly. Furthermore, the face recognition model can recognize people's faces and give their names as well.

In the first phase, we have already reached 99.6% accuracy of predicting whether people wear a mask or not from static images. In the second phase, we searched for two new datasets with better quality for our purposes, the MAFA (MAsked FAces) dataset for wearing correctness check and the MFR2 (Masked faces in Real world) dataset for masked face recognition.

We also further utilized the algorithm that we raised in phase one to build and train three new classifiers, including wearing correctness, unmasked face recognition and masked face recognition. In the end, we tried to deploy the models in a near real-world scenario to detect multiple faces from a surveillance camera, suggest whether each of the people captured in the images wears the mask and wears it correctly and bring out their names.

2 METHODS

In this project, followed by the face mask detector we built previously, there is a wearing correctness check part to detect whether a person is wearing a mask correctly, for those who don't we further built a face recognition to find out who they are. If the person is in our dataset, the system will provide his/her name, otherwise, an unknown tag will be given. [3]

2.1 Data processing

2.1.1 MAFA Dataset.

As the Real-World-Masked-Face-Dataset[10] we used for training face mask detection does not contain enough label information or high quality images, we found another dataset MAFA[4]. The MAFA consists of 30,813 Internet images, in which 35,806 masked faces are annotated.

The annotation in the dataset includes locations of faces, eyes and masks, face orientation, occlusion degree, and mask type. To measure the occlusion degree, we divide a face into four major regions, including eyes, nose, mouth and chin. According to the number of regions occluded by masks and/or glasses, we define three occlusion degrees, including Weak Occlusion (one or two regions), Medium Occlusion (three regions), and Heavy Occlusion (four regions)(see Fig.1). We define four categories of masks that can be frequently found in Internet images, including: Simple Mask (man-made objects with pure color), Complex Mask (man-made objects with complex textures or logos) and Human Body (face covered by hand, hair, etc.). Figure 2 shows an example of the annotated face.

Among these annotated faces, there is a small proportion of images that are too small to be detected especially in the real world scenarios(shown in Fig.3). In that case, we skip all these small faces whose proportion of the whole photo is less than 5%. We aligned faced based on the face location in the label and resize

Figure 1: Occlusion Degrees

Figure 2: Annotated Face Example

them to 160*160 in order to pass them to FaceNet in the next stage. After data processing we have 25,877 images and 25,039 faces in the training set and 4936 images, 5720 faces in the test set which composed 80%, 20% of the whole dataset respectively.

In the wearing correctness task, we found the dataset is highly imbalanced with 21K wearing corrected images and 3K not correct ones. To overcome this issue, we augmented the minor class as what we did in phase 1. The process flips the images and changes the tune of them to reach a 10 times augmentation.

2.1.2 MFR2 Dataset.

Due to the poor image quality in the original dataset we used in phase 1, we introduced a new dataset: Masked faces in real world for face recognition (MFR2)[1][2] to be used in masked face recognition. MFR2 is a small dataset with 53 identities of celebrities and politicians

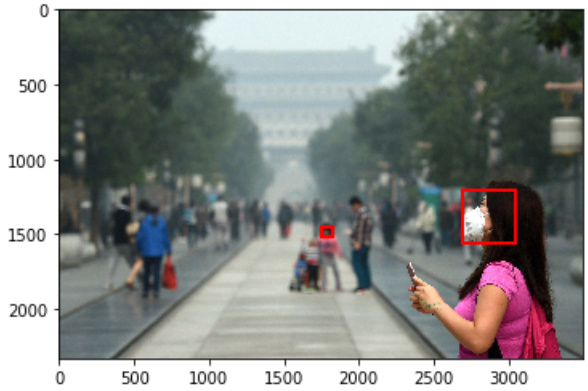

Figure 3: Tiny Face Example

with a total of 269 images that are collected from the internet. Each image has a dimension of (160x160x3). We skipped people with less than 5 images (with/without mask) in the dataset and got 33 people to be recognized totally. We randomly pick one masked face image for each person as a test image and train the model with the rest images containing both masked and unmasked faces.

3 MODELING

3.1 Face Alignment with MTCNN

Multi-task Cascaded Convolutional Neural Networks (MTCNN) is a framework developed as a solution for both face detection and face alignment with very good results. It uses cascaded CNN with three stages including the proposal network (P-Net), the refine network (R-Net), and the output network (O-Net). After each stage, the model can gain more information and focus on more details of faces in the image.

We used the Python/Tensorflow implementation of MTCNN provided by David Sandberg from GitHub[8].

3.2 FaceNet

Embedder is used to transfer images to embeddings. For this part, we decided to use the FaceNet model as the embedder to accomplish our tasks, since this model is widely used in face recognition and performs very well.

The FaceNet[9] model is a system originally presented from research by Schroff, Kalenichenko, and

Philbin. It can directly learn a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. In the FaceNet model, they also introduce triplet loss that is compatible with each other and allow for direct comparison between each other. The triplet loss is designed to minimize the distance between an anchor and a positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity. For the performance, they achieve a very high accuracy of 99.63% on the widely used Labeled Faces in the Wild (LFW) dataset.

We used the pre-trained FaceNet model called 20180402-114759, which is trained on VGGFace2 dataset with Inception ResNet v1 architecture by Google[6]. The architecture of the FaceNet model we used consists of an input layer and a deep convolutional neural network followed by L2 normalization, which results in the face embedding.

After the FaceNet model, the output embeddings are normalized and regarded as input into the classifiers for different purposes. All our classifiers share the same embeddings.

3.3 Classifier

In the face recognition part, since we adopted FaceNet as our embedding generator, we simply used the SVM(linear kernel) classifier inside the whole package. For the wearing correctness part, we use the Neural Network model with 2 hidden layers(5,2). Here, we simply called the sklearn package MLPClassifier to avoid two tensorflow sessions in the live streaming application. For the same reason, we adopted logistic regression as the classifier for face mask detector in the webcam version. As we discussed in the phase 1, the performance of logistic regression is quite close to that of the Neural Network.

3.4 Webcam

For real-world scenarios, to apply our model in a live streaming, we combined all the classifiers mentioned above and deployed them with the Webcam, shown in Figure 4.

In the Webcam, we utilized OpenCV[7] to capture the image and do face alignment and then resized images to 160 by 160. The first classifier is the mask detector. If the person is wearing a mask, then we will decide if he/she is wearing it correctly, that is, following experts' wearing guidelines to cover mouth and nose, using wearing correctness classifier. If the person is wearing it correctly, we will display the text "wearing correctly", or if not, "please wear your mask correctly". On the other hand, if the person does not wear the mask correctly, we will then apply the face recognition classifier to recognize the person's face and display the text "person's name: please wear your mask".

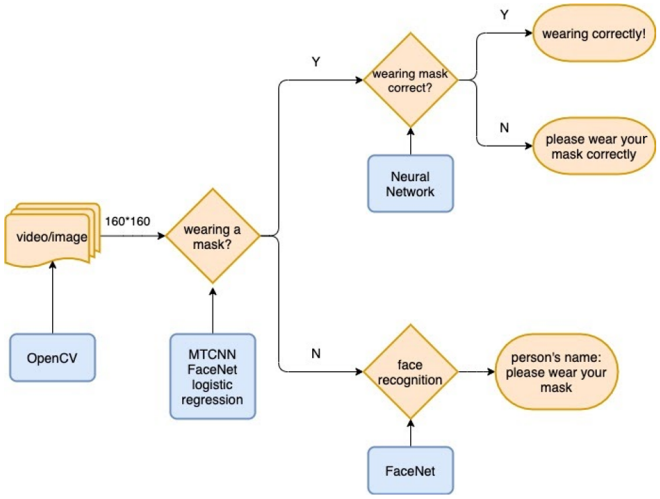

Figure 4: Webcam Flow Chart

4 RESULTS

We added a face alignment component in the system to extract faces from the whole image. The mask wearing correctness achieved 84% accuracy, unmasked face accuracy is 93% and masked face accuracy is 82%. Combined all components, we also made the system applicable to live streaming.

4.1 Multi-face detection

Employing the MTCNN we were able to align faces from the whole photos as well as doing a multi-face alignment with unmasked face and masked face. Even for some blurred faces, our model could extract the face bounding box correctly(see Fig.5 and Fig.6). With this component, our system could process any images containing faces instead of cropped faces only, which

Table 1: Wearing Correctness: Details for More Metrics

	Precision	Recall	F1-score	Support
Correct	0.96	0.85	0.90	4722
Not Correct	0.52	0.82	0.64	997
Accuracy	–	–	0.84	5719
Macro Average	0.74	0.83	0.77	5719
Weighted Average	0.88	0.84	0.85	5719

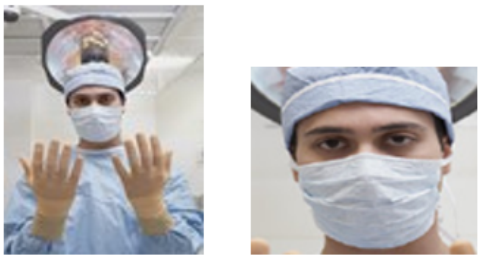

Figure 5: Single Face Alignment

Figure 6: Multi-face Alignment

makes it more applicable to real-world scenarios and live streaming applications.

4.2 Mask wearing correctness check

We explored several traditional classifiers that may suit our task. The best performance was achieved by Neural Network(see Table 2). The details of the final results are shown in the Table 1. The accuracy is 84% though the not correct class's precision is still not good enough, which may be due to less image diversity or wrong labeled photos in the dataset, the final model performs quite well in the live streaming version as shown in the webcam section.

4.3 Unmasked face recognition

Table 2: Wearing Correctness: Accuracy for Different Models

	Test Accuracy	Train Accuracy
Logistic Regression(L2)	0.76	0.82
AdaBoost	0.72	0.80
Random Forest	0.81	1
Neural Network	0.84	0.86

In this section, we explored different numbers of images for training per person and a variety of numbers of people to be recognized. As shown in Table 3 and Table 4, we achieved higher accuracy with more training images per person. The accuracy is 79% when we train the model with 95 images per person. On the other hand, a larger number of people to be recognized leads to lower accuracy, which is consistent with our intuition. The accuracy increased from 79% to 93% as we lowered the number of people in the database from 430 to 30. The best performance of the model is 93% under 30 people to be recognized and 95 training images per person. Compared with the 99% accuracy of FaceNet on LFW dataset under 19 people to be recognized and 6 training images per person, our accuracy could be further improved by improving the quality of training images and increasing the diversity of ethnicity.

4.4 Masked face recognition

We adopted the same model to do a masked face recognition initiative and achieved 81.8% accuracy. Since the dataset is quite small, we could not verify the relationship between accuracy and number of training images and number of people in the database as we did in unmasked face recognition. In the future, we can

Table 3: Accuracy of Face Recognition with Fixed Number of Persons

	Face Recognition for Uncovered Face on Our Own Dataset			Face Recognition for Uncovered Face on LFW
# images for training per person	6	35	95	6
Accuracy	0.54	0.75	0.79	0.99
# person		430		19

Table 4: Accuracy of Face Recognition with Fixed number of Images per Person

	Face Recognition for Uncovered Face on Our Own Dataset				Face Recognition for Uncovered Face on LFW
# images for training per person			95		6
Accuracy	0.79	0.85	0.88	0.93	0.99
# person	430	227	100	30	19

certainly improve the model performance with a better dataset with more high quality images and other techniques discussed in the next section.

4.5 Webcam

As a deliverable, we combined all the classifiers we talked about before including a mask detector which we trained in phase 1, a wearing correctness check classifier and a face recognition model.

The webcam will capture people's real-time face and used our classifier models to first detect if the person is wearing a mask, and then determine if the person is wearing it correctly, and if the person is not wearing a mask, the person's face will be recognized and his or her name will be displayed on the capture. Our webcam works well on a single person as well as multiple people.

There are in total three scenarios. We have recorded a video example with all our team members, and Figure 7 shows all these situations. Green means the person is wearing the mask correctly, for which the text is "wearing correctly". If the person is wearing a mask but not wearing it correctly, that is, not covering his mouth and nose at the same time, the box will turn to yellow and the text will be "please wear your mask correctly". For the last situation, if the person is not wearing a mask, the box will be red and our model will also recognize the person, for which the text will be the person's name, followed by "please wear your mask". For example, for our team members, our names are shown as Zishen, Junmei, and Tony.

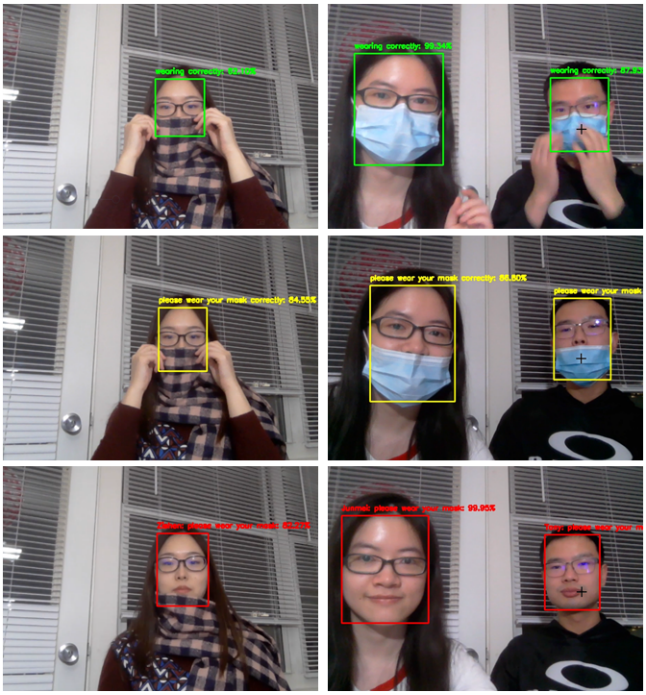

Figure 7: Webcam Sceenshots

5 DISCUSSION

5.1 Human Body

In our wearing correctness model, we did not take the human body occlusion into consideration so that anyone who occludes their mouth and nose with anything might be detected as wearing a mask correctly as shown in our demo video. Since this live streaming application is to remind people to protect themselves as well as others in a public place, so if it could let people notice they need to occlude their mouth and nose, even with hands, should be counted as effective. But in the future, we may add the human body component to the system so that it will give more accurate and detailed instruction. Based on the label of our dataset, it could be done by simply adding another classifier after the embedding extraction part.

5.2 Unmasked Face Recognition

In the face recognition task, the relationship between accuracy and number of training images and number of people in the database is consistent with our intuition that the model trained with more images and fewer classes will provide better performance. Our best accuracy is still 6% lower than what FaceNet performs on LFW.

The possible reasons for this may be: 1. The less ethnicity diversity in our dataset compared with the LFW. In our dataset, all faces are Asian people while in LFW the ethnicity is more broadly spread.2. As mentioned before, our dataset contains some low quality images which may impact both the training and test process. Since the masked face recognition using the same model on a high quality dataset reached 82% accuracy, we believe a better dataset will certainly improve the system performance. Additionally, in the real-world scenarios, it is much easier to capture multiple high quality photos for each person in the database. So, collecting a high quality dataset is feasible and reasonable for our application.

5.3 Masked Face Recognition

Our 82% accuracy of masked face recognition may be due to the mixed training dataset with masked and unmasked faces, which could be easily implemented in the real-world scenario. For further improvement, there is a work[5] showing that cropping uncovered part of a face, like eyes and forehead etc. could largely improve the performance. The cropping process may help exclude some noise from the whole photos. Since we only trained the model by average 6 images per person, if the relationship we found in the unmasked face recognition is applicable to this task, we may also improve the performance a lot by expanding the training set for each person.

6 STATEMENT OF CONTRIBUTIONS

Zishen Li: data processing, mask wearing correctness, masked face recognition

Junmei Luo: FaceNet embedder, Webcam

Tony Yukuang Zhang: face alignment, unmasked face recognition

REFERENCES

[1] Aqeel Anwa. [n. d.]. MaskTheFace — CV based tool to mask face dataset. https://towardsdatascience.com/masktheface-cv-based-tool-to-mask-face-dataset-1a71d5b68703

[2] A. Anwar and A. Raychowdhury. [n. d.]. Masked Face Recognition for Secure Authentication. https://arxiv.org/pdf/2008.11104.pdf

[3] N. Damer, J. H. Grebe, C. Chen, F. Boutros, F. Kirchbuchner, and A. Kuijper. 2020. The Effect of Wearing a Mask on Face Recognition Performance: an Exploratory Study. In *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 1–6.

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

[5] Walid Hariri. 2020. Efficient Masked Face Recognition Method during the COVID-19 Pandemic. https://doi.org/10.21203/rs.3.rs-39289/v1

[6] Athul P. [n. d.]. Building a face recognition system with FaceNet. https://medium.com/@athul929/building-a-facial-recognition-system-with-facenet-b9c249c2388a

[7] Adrian Rosebrock. [n. d.]. OpenCV Face Recognition. https://www.pyimagesearch.com/2018/09/24/opencv-face-recognition/

[8] David Sandberg. [n. d.]. Facenet Implementation. https://github.com/davidsandberg/facenet

[9] F. Schroff, D. Kalenichenko, and J. Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 815–823. https://doi.org/10.1109/CVPR.2015.7298682

[10] X-zhangyang. [n. d.]. Real World Masked Face Dataset. https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset