Real Time Face Mask Detection System

Ding Ma
Department of Computer Science
Boston University
martin98@bu.edu

Dingjie Chen
Department of Computer Science
Boston University
djchen21@bu.edu

Abstract

In this paper, we built a model that detects whether people are wearing masks or not in real-time, which could work as a simple precautionary measure to prevent the spread of virus. A basic Convolutional Neural Network (CNN) model is developed using Pytorch and OpenCV to make the algorithm as accurate as possible. The proposed work contains three stages: (i) pre-processing, (ii) training CNN models and (iii) classification. The first stage is the pre-processing section The first stage is the pre-processing section which includes extracting faces from original images, reshaping and normalization. Then the proposed CNN, sorts faces as with or without masks. The proposed model has Validation accuracy of 96.39%. If anyone in the video stream is not wearing a protective mask, a Red colored rectangle is drawn around the face with a dialog entitled as NO MASK. A Green colored rectangle is drawn around the face of a person who is wearing a mask.

1 System Pipeline

In this project, we build a real time face mask detection system that can recognize human faces and classify whether the person is wearing a mask or not. The system's pipeline is shown in Fig. 1. The whole system contains two stages, detection and classification. The first stage is to detect human faces showing in the current frame. And then chop off the faces and resize them to predetermined dimensions. The second stage is to feed the detected faces into the model trained to sort whether the input face is wearing a mask face or not. Then, if anyone in the video stream is not wearing a protective mask, a Red coloured rectangle is drawn around the face with a dialog entitled as NO MASK In opposite, a Green colored rectangle will be drawn around the face of a person who is wearing a mask.

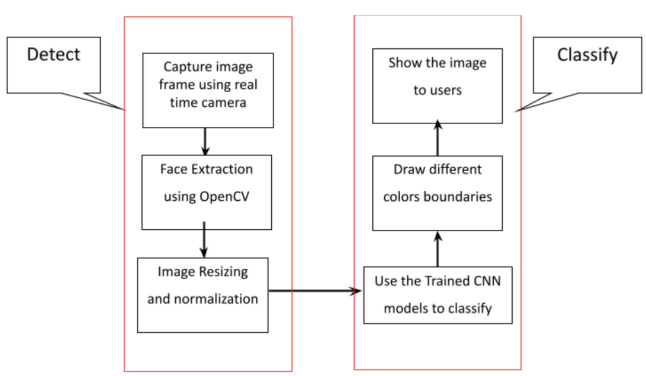


Figure 1: System pipeline

2 Preparing Data Set

The data set used for this project is face mask detection data set on Kaggle. The link is here:

`https://www.kaggle.com/wobotintelligence/face-mask-detection-dataset/tasks?taskId=1123.`

This data set contains 20 different classes. Among these classes, two of them are favored for this project are 'face_with_mask' and 'face_without_ mask'. Table 1 is showing some examples of this data set.

Table 1: The examples of training data

Name	x1	x2	y1	y2	Class Name
2756.png	69	126	294	392	face_with_mask
2756.png	505	10	723	283	face_with_mask
6098.png	360	85	728	653	face_without_mask
5525.png	262	18	484	319	face_without_mask
5525.png	49	58	191	247	face_without_mask
3911.png	49	26	389	476	face_without_mask

In filtered training data. The first column contains the name of the images. The second to fourth columns represent the bounding boxes' positions. And the last column contains the label name of the corresponding face. We first used these bounding boxes' positions to extract human faces from the training data. Fig. 2. shows the distribution of wearing masks and not wearing mask faces in the data set.

Faces without mask are far less than faces with mask, this may cause biases of the model, so we decide to do some data augmentation for no mask class, including random rotation, resizing, adding noise. Table 2. shows the examples of the augmented data.

After augmentation, the size of class 'No mask' and 'mask' are balanced. Therefore, we could move forward to the training stage. Next step is to train the deep neural network model to classify these images. Fig 2. shows the distribution of training data after data augmentation. After data augmentation, we then divide all image data by 255 to compress the data between 0 and 1.

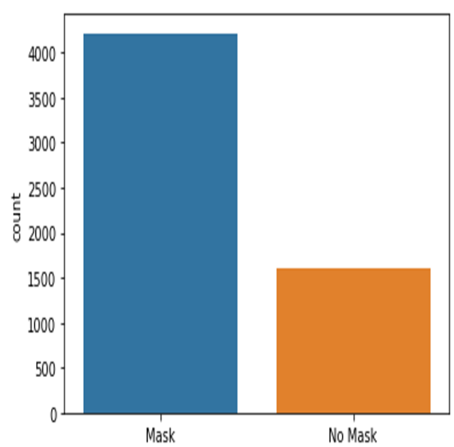


Figure 2: Distribution of training data before data augmentation

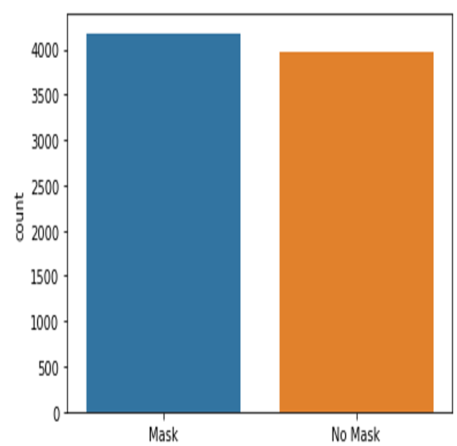
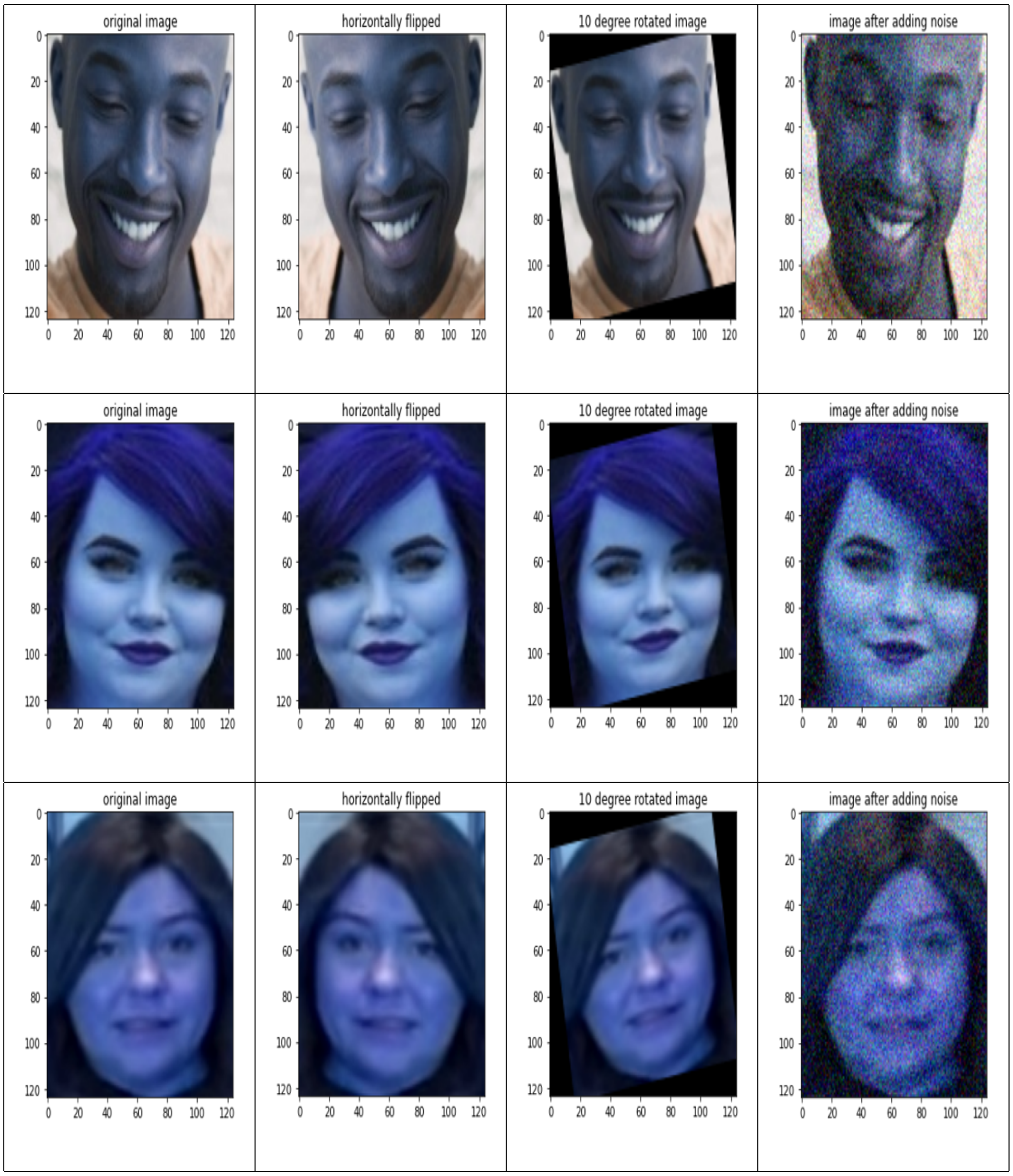


Figure 3: Distribution of training data before data augmentation

3 Network Structure

Training a complex deep neural network is very expensive because it requires high computing power and memory. It is also very time-consuming. In order to save training costs, this article chooses to

Table 2: The examples of data augmentation results



finetune the trained model. Even when training on a small data set, this method can improve the performance of the new model to a certain extent and reduce the training time. The neural network in this system is targeted to classify whether or not the input face is wearing a mask. We implemented the four most popular CNN models including: Alex Net, VGG-16, Dense Net 201 and GoogLeNet in addition to the pre-trained parameters. Discard all the models' classifiers structure in order to fit these models to our goal: two classes: 'No Mask' and 'Mask', classification. The pre-trained models are playing the role of feature extractors in this system. Fig 3. is showing the structure of the whole network structure.

As shown in Fig. 4, in order to improve our recognition accuracy, we use model merging to combine these four models together to make better decisions. In order to increase the diversity between models, the training data of each model is randomly selected from 80% of the total training data, and then trained. During merging, the voting mechanism is adopted, and the prediction results of each model

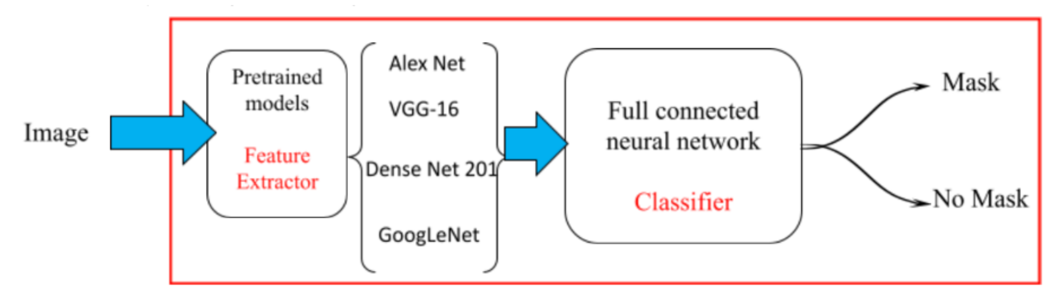


Figure 4: Network structure

are used for voting. The prediction results with more than half of the scores are taken as the final classification results.

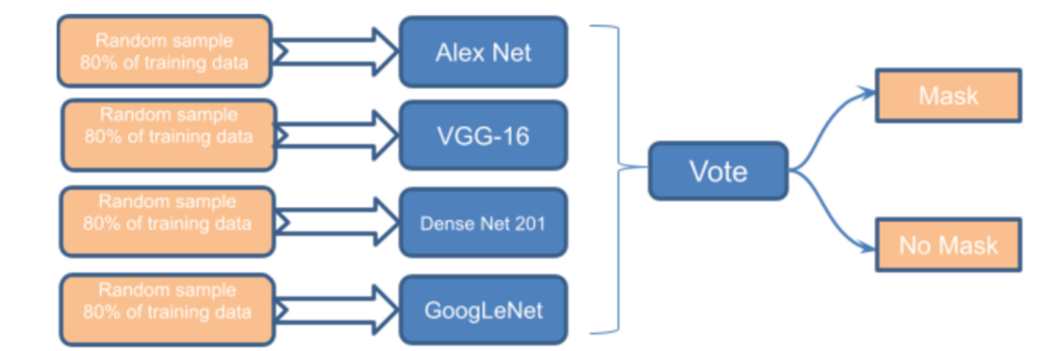


Figure 5: Model merging pipeline

4 Result Analysis

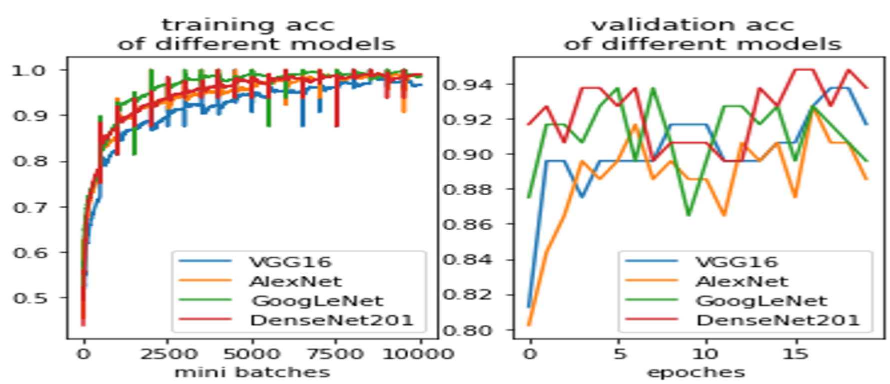


Figure 6: The training and validation accuracy of different models

A successful real-time face detection system should not only have high detection accuracy, but also efficient detection efficiency, and can detect the frame image transmitted by the camera with low delay. So, to analyze the result of different models, we use three metrics to measure the quality of the models: training accuracy, validation accuracy and response time (second/ per frame). We set 10% of training data to be the validation set to test the model's generalization on images which are not seen by model. Fig 4. shows the training accuracy and the validation accuracy curve of different models during the training step. Table 3 is the training results of different models using the same hyper parameters and training data. Besides the 4 pre-trained models used above, we also merged all 4 models together using voting and called this model "Merged Model". We can see from the table that the Merged Model has the highest accuracy but needs longer time to respond. Different models below show different qualities and can be implemented in different environments. Merged model has the highest accuracy and response time, it can be used in scenes that do not need real-time detection but need high accuracy. Alex Net has the lowest response delay, so it is most suitable for real-time detection. VGG-16 has the highest comprehensive ranking and can be used in different occasions.

Table 3: The examples of training data

	Training Accuracy	Validation Accuracy	Response Time (s/frame)
Alex Net	0.9892	0.8645	0.0666
VGG 16	0.9847	0.9270	0.0817
Dense Net 201	0.98914	0.9479	0.1056
Google Net	0.99264	0.9062	0.0895
Alex Net	———	0.9639	0.3434

So far, the whole system runs perfectly. We will show some detection results of our system on the test pictures in appendix.

5 Conclusion

In this article, we trained 4 most popular CNN models, and selected the best one for the research situation in this article. Using deep learning target detection algorithms, we have proposed a high-performance face mask recognition and epidemic prevention system, which can ensure that people who enter public places and those who are already in public places to wear masks. The model is trained on the face mask detection dataset on Kaggle, and the recognition rate can reach 96%. This system meet the actual epidemic prevention needs at a low cost, and has a wide range of application value and strong practical significance.

In the future, in order to we could increase the model's accuracy, it maybe helpful to use more diversity and complex data set to train the detection model. And use object detection pre-trained modern neural networks structures like: YOLO 5, Faster RCNN, SSD. change the classification stages' parameters to classify 'No mask' and 'mask' is also an interesting approach.

References

[1] Arora, R. & J. Dhingra & A. Sharma (2021) Face Mask Detection Using Deep Learning.

[2] Garg, P. S. (2020) "Face Mask Detection System using Deep Learning." International Journal for Modern Trends in Science and Technology 6.12(2020):161-164.

[3] Dhaya, R. (2021) "Efficient Two Stage Identification for Face mask detection using Multiclass Deep Learning Approach." Journal of Ubiquitous Computing and Communication Technologies 3.2(2021):107-121.

A Appendix

A.1 Sample result for model

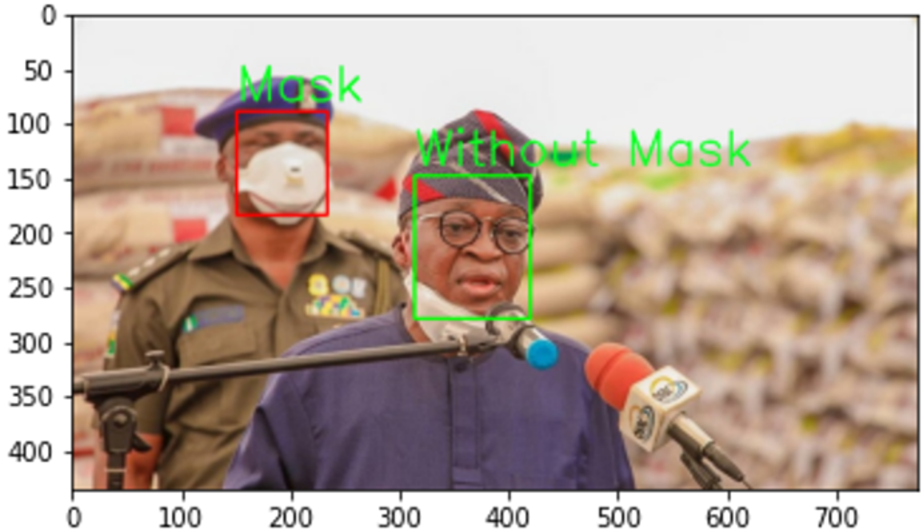


Figure 7: multiple people in the photo

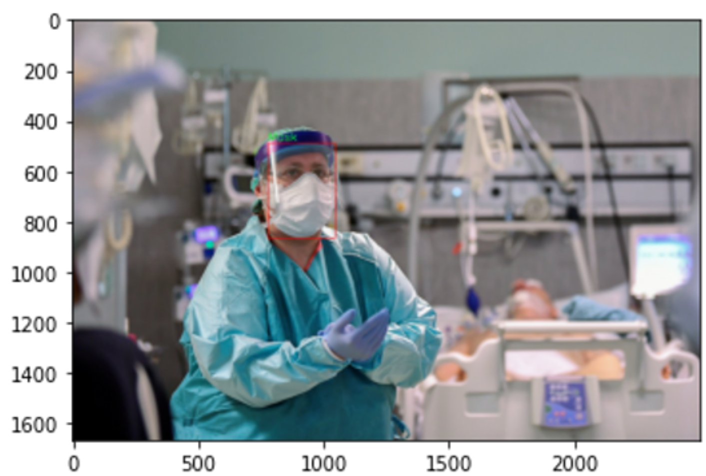


Figure 8: people with mask

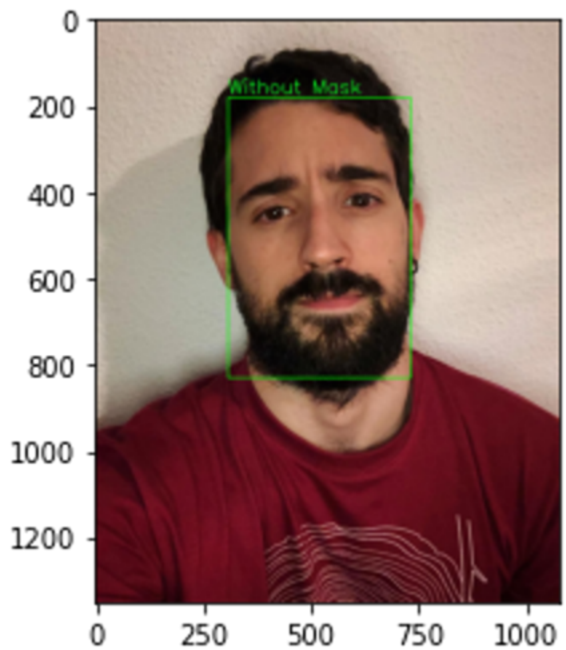


Figure 9: people without mask