Automated Face Mask Detection and Mask Distribution System

Sheikh Munim Hussain
Dept. of Electrical and Electronic Engineering
BUET
Email: munim.sah75@gmail.com
Id: 1706008

Abdullah Al Mahmood
Dept. of Electrical and Electronic Engineering
BUET
Email: almahmudplabon@gmail.com
Id:1706010

Shafin Shadman Ahmed
Dept. of Electrical and Electronic Engineering
BUET
Email: zubeen1971@gmail.com
Id: 1706020

Md.Jahidul Hoq Emon
Dept. of Electrical and Electronic Engineering
BUET
Email:jahidulhoq7@gmail.com
Id: 1706017

Azazul Islam Razon
Dept. of Electrical and Electronic Engineering
BUET
Email:azazulislam965@gmail.com
Id: 1706033

Dr. Shaikh Anowarul Fattah
Professor
Dept. of Electrical and Electronic Engineering
BUET

Tasnima Afsana
Adjunct Lecturer
Dept. of Electrical and Electronic Engineering
BUET

***Abstract*—COVID-19 pandemic has rapidly affected our day-to-day life, disrupting the world trade and people's movements. Wearing a protective face mask has become a new normal. Therefore, face mask detection has become a crucial task to help the global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV. This paper also demonstrates real-life application of barrier up-down system and mask giving option using Arduino control system. The proposed method detects the face from real time data correctly and then identifies if the person wears a mask or not. If the person is wearing a mask, the barrier is lifted up. Otherwise, the barrier goes down and the person is given a mask by the Arduino control system. After he wears the mask, the barrier is lifted up accordingly. Different parameter tuning was done during the project to have the optimum detection and integration of hardware with software.**

***Keywords*—COVID-19 ,Machine Learning, Arduino, Python, Face Mask Detection, Barrier.**

I. Introduction

Since the end of 2019, the infectious coronavirus disease (COVID-19) was reported for the first time in Wuhan, and it has become a public fitness issue worldwide. This pandemic has devastating effects on societies and economies around the world causing a global health crisis Many shutdowns in different industries have been caused by this pandemic.

With rising numbers of cases and stretched health facilities, as well as the lack of vaccines and difficulties associated with achieving herd immunity for COVID-19, government inaction became increasingly non-viable. The application of advanced artificial intelligence (AI) techniques can lead to control the COVID-19 situation and make sure that people are wearing masks.

Face detection is a key area in the field of Computer Vision and Pattern Recognition. This proposed work can help mitigate the problems of COVID 19 situation by making sure that people are wearing masks or not.

The main points of the proposed work is stated below:

(i) A face mask detection method has been developed that accurately detects the mask in real time from video streams with transfer learning at the back end.

(ii) A red/green box is encircled on the face of the person being detected.

(iii) A control system takes feedback from the software whether the person is wearing a mask or not.

(iv) A barrier system is demonstrated to show the integration between the software and hardware through real life examples.

(v) A mask distributor is also modeled to make sure that no person without a mask has access to a certain place.

The proposed model requires less memory, making it easily usable for embedded devices used for surveillance purposes.

II. Software

A. Data Collection

We used a Kaggle dataset [1] of 11792 data in total. In the dataset, the ratio of male and female data was almost the same. In the data, there were 5883 data labeled as 'WithMask' and 5909 data labeled as 'WithoutMask'.

B. Data Splitting and Labeling

We labeled the data using LabelBinarizer. It labeled the whole data into binary encoding according to their label ('WithMask' and 'WithoutMask'). It helps to feed our data in 'softmax' layer or decision taking layer.

After that, we split the data. We split the data roughly as 80:10:10. To be more precise, we used 10000 data for training (5000 of 'WithMask' and 5000 of 'WithoutMask'), 992 data for test (483 of 'WithMask' and 509 of 'WithoutMask') and rest 800 data for validation purpose (400 of 'WithMask' and 400 of 'WithoutMask'). Before splitting, all the data were randomly shuffled to prevent over-fitting.

C. Building The Model

In this project a variety of models were used. At first, we used 'ImageDataGenerator' for data augmentation. Then we used MobileNetV2 [2] as base model. After that we used different deep learning layers to train the model.

a) Image Augmentation with ImageDataGenerator: From Keras, the image preprocessing tool, ImageDataGenerator was used to augment data. Different parameters were changed to augment data, namely rotation range, zoom range, width shift range, height shift range, shear range with suitable values for our project. We also kept horizontal flip feature for augmentation. These augmentations increased the amount of data and as a result our training accuracy increased. Fill mode was set at nearest position.

b) MobileNetV2 as base model: MobileNets are small, low-latency, low-power models parameterized to meet the resource constraints of a variety of use cases. MobileNet V2 improves the state-of-the-art performance of mobile models. This model is based on inverted residuals and linear bottlenecks [3]. We used pre-trained model 'imagenet' to start training of the base model.

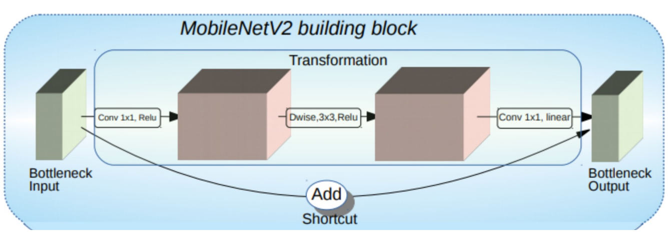

Fig. 1. Architecture of MobileNet V2

c) Deep Learning Models: AveragePooling2D, Conv2D, MaxPooling2D, Flattening layer, Dense and Dropout layers were used for further training. We used 'relu' as activation function and padding was set at same for all the time. At the last layer, we used 'softmax' activation for binary classification on 'Mask' and 'No Mask'.

D. Training The Model

After building the previous models, we trained our model in Google Colab using the data we got and the data we augmented. We used 'adam' optimizer. We used 'binary cross entropy' as loss function.

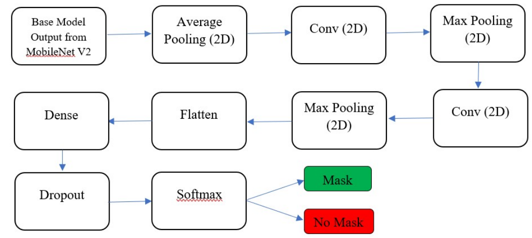

Fig. 2. Architecture of Proposed Deep Learning Network

a) Optimizer: We used 'Adam' optimizer for compiling action. It helps to adjust weights in the trainable layers. We set the training capability of the MobileNet V2 network to false. It also calculates the learning rates required for different epochs and assign them in the respective training epochs. We used adaptive learning rate instead of constant learning rate.

b) Loss Function: We used 'binary cross entropy' as loss function in this project. It calculates the discrepancy between the actual label and predicted label. It also shows how our model is trained up. If our model is well trained, then loss is very small, or negligible, close to 0. The mathematical formula of binary cross entropy is,

$$L_{binary}(x) = -\frac{1}{N}\sum_{i=1} N x_i \cdot \log(p_i(x_i)) + (1-x_i) \cdot \log(1-p_i(x_i)) \quad (1)$$

c) Model Training: We trained the model for 20 epochs. Batch size was taken as 32. Through the whole period accuracy increased and the loss was decreased. The lowest training loss was recorded as 0.0163 and the highest training accuracy was recorded as 99.51%

E. Model Results

After running the model, the following data were obtained:

```
63/63 [========]-8s 133ms/step - loss:0.0214 - acc:0.9925
acc:99.25%
```

	precision	recall	f1-score	support
with_mask	0.99	0.99	0.99	1002
without_mask	0.99	0.99	0.99	1000
accuracy			0.99	2002
macro avg	0.99	0.99	0.99	2002
weighted avg	0.99	0.99	0.99	2002

Fig. 3. Accuracy and Precision Value

The accuracy of this model is 99.25%, which means that the model can almost accurately detect whether a person is using a mask or not. The macro-average and weighted-average accuracy values were also found to be around 0.99, supporting the validity of the previous accuracy mentioned and indicating that the model works fine.

We also see that the precision, recall and F1 scores of the model we used always have a value of 0.99, which is fairly consistent for both masked and unmasked conditions. A high precision means that the model can understand the masked conditions fairly accurately, and a high recall means the model can quickly detect a mask. The F1 score is a weighted average of both precision and recall, and in this case the F1 score is high as expected.

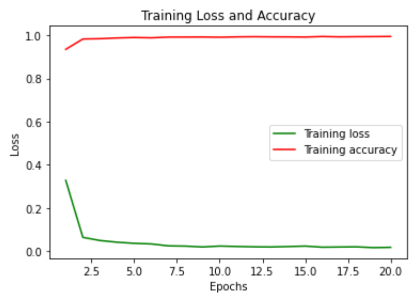

Fig. 4. Loss Function and Accuracy Graphs for Training Set

From the loss function and accuracy graph, it is seen that the accuracy throughout the entire training period has been quite stable. Although the initial loss was higher, it decreased appreciably to around 0.0152 after 2 epochs. The training accuracy never became a solid 1, which means that there was no over-fitting.

Now accuracy of the reference model for that approach was almost 96.85% for train set and 90.52% for validation set.[2] So almost 3% improvement was achieved over the existing model.

III. Hardware

A. Connecting Software With Arduino

Arduino is a hardware-cum-software project which makes micro-controller boards for making digital devices and provides codes to work on the boards. In this project, Arduino was used to implement the barrier and mask-providing mechanisms. The micro-controller board used in the project was

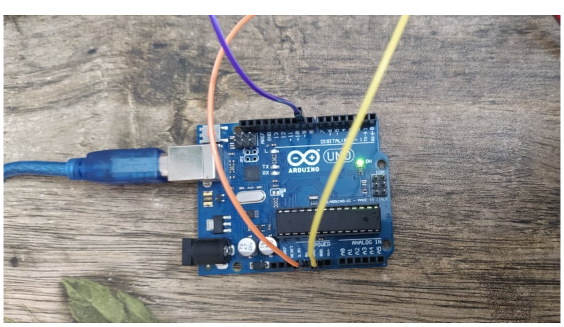

Fig. 5. Microcontroller

Arduino Uno. It has separate ports for output, power supply and ground. The servo motors which were used to make the barrier and the mask-providing mechanisms were connected to the Arduino via jumper wires. For completing the circuit, a breadboard was used. The Arduino board was then connected to the laptop via a USB cable.

An Arduino code was also used in this project. This code, called StandardFirmata, was called at first, and then the main code in Python was run. A Python code was also written to call the Arduino code. By using the said Arduino code, the servo motors were run accordingly.

B. Barrier

A stick and a servo motor were used in the barrier system. This barrier is then controlled using Arduino. The output given by the model decides whether to keep the barrier down or to lift it.

When the main model detects that a person without wearing a mask has appeared in front of the camera, it sets the output at 1. This output is then sent to the Arduino board. Pin 10 in the Arduino board controls this servo. When the input to Arduino is 1, indicating that a person without a mask has appeared, Arduino sets the servo at an angle of zero degree. It means that the barrier is now kept down and the person cannot pass.

Again when a person wearing a mask comes, the model gives an output of 0. When Arduino gets this output, it rotates the servo by an angle of 120 degrees. This means the barrier is now lifted.

Fig. 6. Barrier is Down

In this picture the barrier is shown. The barrier is now at down position, which means the person is not wearing the mask.

C. Mask Distribution System

Another stick and another servo motor were used as the mask distribution system, which was again controlled by the same Arduino. The output given by the model decides whether to provide the mask or not.

As previously mentioned, when a person without a mask appears, the Arduino gets an input of 1. Pin 9 in the Arduino board controls this second servo motor. When the input to Arduino is 1, it sets the servo to an angle of 75 degrees. As a result, the stick attached to this servo pushes the mask to the person, and the person gets this mask.

On the contrary, when a person wearing a mask appears, it sends an signal 0 as input to Arduino. Then Arduino resets this servo, or in other words, sets the servo at an angle of 0 degrees. This causes the stick to go back to its original position or the lifted position, thus allowing the person to pass.

Fig. 7. Mask Distribution System

This is the mask distribution system. Detecting whether a person is wearing a mask or not, it pushes the mask forward.

IV. Result

In summary, the deep learning model analyzes a person's face and detects whether he is wearing a mask or not. The output was generated for different qualities of images and different positions of the mask. And then on the basis of the model output, Arduino controls the servo motors of the barrier and the distributor system and allows a person with a mask to pass and provides a mask to a person who doesn't wear it properly.

Fig. 8. Blurry Image Detection

Fig. 9. Clear Image Detection

V. Limitation

Major limitations could appear if we try to detect a person from a much larger distance. In our analysis, a person up to 4 feet distance can be detected by the model. But beyond this distance, it couldn't recognize the mask. The problem exacerbates if the person tilts his face or moves aside. In that case, the model can detect the face up to 3 feet distance.

Fig. 10. Person at a distance lower and greater than 4 feet

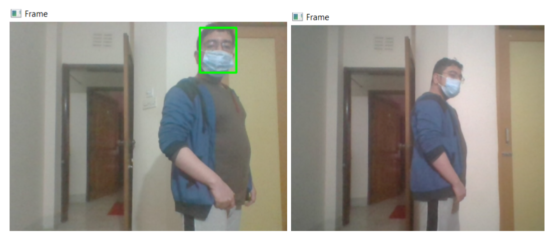

Fig. 11. Person at a distance lower and greater than 3 feet tilting his face

Another problem that could appear in the presence of broad daylight as it could affect the detection of the laptop's webcam. In that case servo motor's movement could be shaky and the barrier could oscillate instead of remaining still in a position. A better camera setup could solve these problems.

VI. Cost Analysis

As the target was to minimize the cost as far as possible, apart from the software setup, an additional Arduino Nano and two servo motor of SG90 models were used. In total they cost around 1000 Taka. Now further improvement, smoothness and clear detection could require a better camera, motor of higher load capability and raspberry pi for better operation. In this case cost could increase much higher than the required one here.

VII. Further Improvement

The project could be further improved and made eligible for industrial use by using better camera to detect multiple persons from a greater distance. Also, instead of a pushing system a robotic arm could be added. The model could be improved for the face detection of particular persons to maintain a dedicated database about their awareness.

VIII. Conclusion

In this project, a model face mask detector and distributor system was implemented. The primary objectives were to develop a simple model to maintain the COVID norms and protocols without coming into contact. MobileNetV2 model was used to develop a deep learning model for the detection of the person from distance. Later a hardware demonstration was implemented ad the efficacy of the system was analyzed in different lighting system and from different distance. As a person can be detected and provided mask through this system, further improvement can make it much more useful in industrial level and in interaction without causing physical contact.

Acknowledgment

Throughout the project we have been supported by the Department of EEE, Bangladesh University of Engineering and Technology. We would like to offer our sincere gratitude to Dr. Shaikh Anowarul Fattah, Professor, Department of EEE, BUET and Tasnima Afsana, Adjunct Lecturer, Department of EEE, BUET since without their directions and guidance, we couldn't complete the whole project. Their advice and suggestions were much valuable for us to complete the project properly.

References

[1] Ashish Jangra, "Face Mask Detection 12K Images Dataset", https://www.kaggle.com/ashishjangra27/face-mask-12k-images-dataset

[2] Samuel Ady Sanjaya and Suryo Adi Rakhmawan, "Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic", 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-5

[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks"

[4] Florian Schroff, Dmitry Kalenichenko, James Philbin, "FaceNet: A unified embedding for face recognition and clustering"

[5] Bosheng Qin, Dongxiao Li, "Identifying Facemask-wearing ConditionUsing Image Super-Resolution with Classification Network toPrevent COVID-19"

[6] Mingjie Jiang, Xinqi Fan, "RetinaMask: A Face Mask detector"