Development of Lightweight Facemask Detection Software using CNN and TensorFlow

Arshdeep Singh Bhatia
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT)
Vellore 632014, TamilNadu, India.
arshdeepsbhatia11@gmail.com

Arpit Khandelwal
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT)
Vellore 632014, TamilNadu, India.
arpit.khandelwal2002@gmail.com

Kartik Nahta
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT)
Vellore 632014, TamilNadu, India.
kartiknahta@gmail.com

Shivam Singhal
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology (VIT)
Vellore 632014, TamilNadu, India.
shivamsinghal24@gmail.com

Abstract

With the pandemic having such a widespread impact on the globe there is a great need for all of us to wear masks. The latest forecast from the Institute of Health Metrics and Evaluation suggests that 33,000 deaths could be avoided by October 1 if 95 percent of people wore masks in public. However, intentionally or unintentionally we seem to remove our masks while entering public places. We aim to make a tool which will be able to detect if a person is wearing a mask properly covering the vital areas of nose and mouth. Then based on the results of the AI we will be able to trigger further actions such as open the door, or send an alert message saying that the mask isn't worn properly or not worn at all. The tool shall be trained with a dataset of 1915 images of people wearing a mask and 1918 of those without a mask. Vital points of the face such as nose and mouth will be classified and the output shall be presented using the results obtained from the processing of the data.

Keywords *— Classification, CNN, Keras, COVID-19, TensorFlow.*

I. INTRODUCTION

The overview of the system shall be covered in this section. The system software we have developed is developed in python programming language. We have also used python libraries like OpenCV and mobilnetV2 architecture. These libraries enable us to perform image processing and analysis features that further help in the prediction of whether the face in the video is covered by a mask or not. The system refers to a model which is trained using about 4000 images classified as with mask and without mask respectively. Upon model training a particular plot is generated which helps us understand the accuracy of the model and what levels of improvements can be made.

Now that all the prerequisites are satisfied the python system is started which takes in the video stream and compares it with the trained model and determines the percentage of the face covered with the mask. Above a threshold value of 50 % the system shall conclude a mask is worn and the frame surrounding the subject will be green else the system shall show as red and certain actions can be taken based on this.

A. Objective

- Through this proposed Face mask recognition software, we aim to track and precisely compute the percentage of face covered by mask and classify it as safe or unsafe, among some other options.
- Then based on the results of the AI we will be able to trigger further actions such as open the door, or send an alert message saying that the mask either isn't worn properly or not worn at all.
- We are going to use video streams in real-time and will use photos as the data set.
- The tool will be able to generate the output (with mask/without mask) without the need of storage of the video stream.

B. Limitations

- This system may need more training of dataset as it can definitely work better and longer than humans can but its accuracy may not be as good as that of human monitoring system
- The system can only analyze subjects if the face is visible in the video stream and thus a particular hardware may limit the system's performance.
- The system only detects people wearing masks and may fail to detect in case facial features like eyes and forehead are covered.

II. SYSTEM ANALYSIS

A. Proposed System

- The work we propose to do refers to a research paper that proposes a real-time face mask detection system by applying computer vision and machine learning concepts like convolution neural networks (CNN) and refined Mobile

Net V2 architecture to ease the deployment of the proposed model in embedded devices with limited computational capability.

- The data set utilized here contains about 12000 images (6000 with mask 6000 without mask). The model is trained using Adam optimizer algorithm which is best suited for deep learning models and is built using keras, TensorFlow and open CV.

- The proposed model touches 99% accuracy under various training to testing ratios like 70% training and 30% testing 50% training and 50% testing etc. Precision, recall, F1 score support are calculated for all trials.

- This means that the system is computationally effective and could potentially be used in places like railway stations, airports, ATMs and other public places to detect people not wearing face mask and ensure the safety to certain extent in this pandemic.

- If our system is able to detect that the person does not wear a mask a small speaker or other such hardware devices could be used and an alert message could be sent or an alert audio could be played in front of the person to alert or notify` the person to wear the mask.

B. Benefits of Proposed System – Shivam

- Light-Weight and Quick Face Detection and Analysis.
- Can Be Integrated with Security Cameras and Microphones to Give Alerts.
- Real-Time Monitoring and Multi Face Analysis

III. REQUIREMENT SPECIFICATION

A. Hardware Requirements

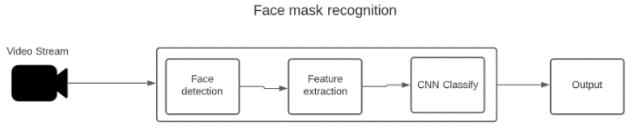
1. Configured GPU (optional)
2. Webcam

B. Software Requirements

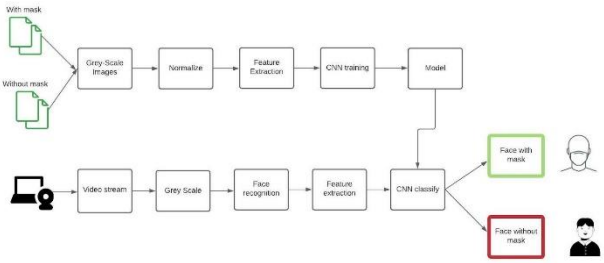
- Python
- Tensorflow
- Keras
- Numpy
- MobilenetV2
- OpenCV
- Frontal Face Classifier API

IV. SYSTEM DESIGN SPECIFICATION

A. System Architecture



B. Detailed Design



V. MODULE DESCRIPTION

A. Dataset

Our dataset contains two directories:

a. With Mask
b. Without Mask

Both these directories contain 500 PNGs and JPEGs of respective categories.

The “with mask” directory has photos of people wearing mask correctly, which fulfils the following requirements:

a. Nose is correctly and fully covered with mask
b. Mouth is covered with mask
c. Mask covers whole nose-mouth-chin region and is worn over the ears.

The “without mask” directory has normal images of people from different countries and ethnicities.

B. Data Preprocessing

a. Initialize the initial learning rate, number of epochs to train for and batch size
b. Grab the list of images in our dataset directory, then initialize the list of data (i.e., images) and class images
c. Perform one-hot encoding on the labels

C. Face Detection essentials

a. Construct the training image generator for data augmentation
b. Load the MobileNetV2 network, ensuring the head FC layer sets are left off

D. Train model

a. Construct the head of the model that will be placed on top of the base model
b. Place the head FC model on top of the base model (this will be the actual model we will train)
c. Loop over all layers in the base model and freeze them so they will not be updated during the first training process

E. Model

a. Compile our model
b. Train the head of the network
c. Make predictions on the testing set
d. For each image in the testing set we need to find the index of the label with corresponding largest predicted probability
e. Serialize the model to disk
f. Plot the training loss and accuracy

F. Face mask detect

a. Grab the dimensions of the frame and then construct a blob from it
b. Initialize our list of faces, their corresponding locations, and the list of predictions from our face mask network
c. Loop over the detections and extract the confidence (i.e., probability) associated with the detection
d. Compute the (x, y)-coordinates of the bounding box for the object and ensure the bounding boxes fall within the dimensions of the frame
e. Add the face and bounding boxes to their respective lists
f. Return a 2-tuple of the face locations and their corresponding locations
g. Load our serialized face detector and mask detector model from disk
h. Initialize the video stream and loop over the frames from the video stream
i. Grab the frame from the threaded video stream and resize it to have a maximum width of 400 pixels
j. Detect faces in the frame and determine if they are wearing a face mask or not
k. Loop over the detected face locations and their corresponding locations
l. Determine the class label and color we'll use to draw the bounding box and text
m. Include the probability in the label and display the label and bounding box rectangle on the output frame
n. Show the output frame

VI. Conclusion and Future Work

We aim to make a tool which will be able to detect if a person is wearing a mask properly covering the vital areas of nose and mouth. Then based of the results of the AI we will be able to trigger further actions such as open the door, or send an alert message saying that the either mask isn't worn properly or not worn at all.

This real-time mask detection module can be further used in the following cases:

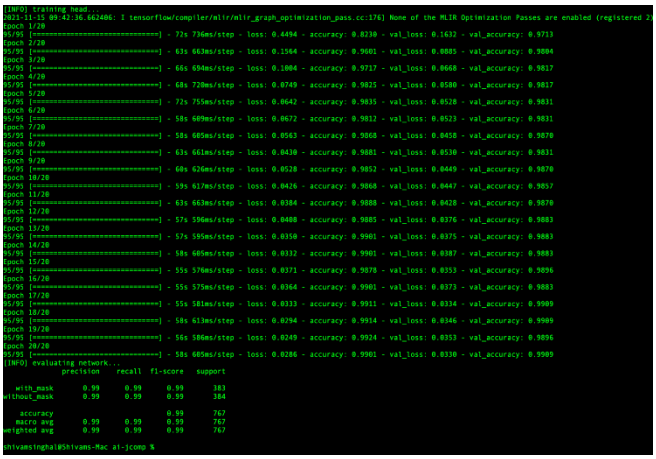
a. Public places like malls, railway stations, airports, parks and anywhere with CCTVs and raspberry Pi with a monitor to show who is not wearing a mask properly.
b. At corporations, factories and places where people may want to remove mask but pandemic isn't over yet.
c. This technology can also be used with thermal screening modules with required IoT devices to track who may possess the Corona virus and segregate him from public for further inspection to for preventing spread of virus.
d. The face mask detection module also has capabilities to capture the faces and their mask on recorded videos, so the healthcare services can use this to track epicenters and further track the people who originally were infected and didn't wear their masks properly which led to spike in covid cases in particular regions/neighborhoods.

VII. Appendix

A. Appendix 1 – Source code

Source code can be found at this link

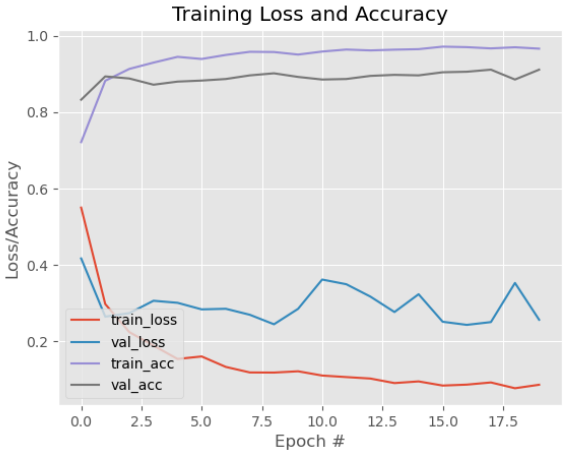
B. Outputs –



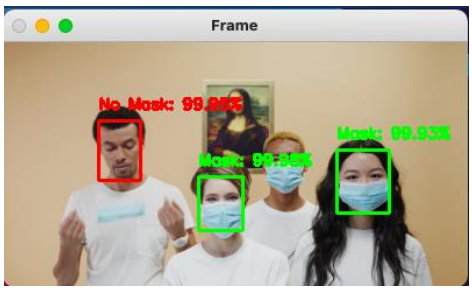
Training Screenshot

```
[INFO] evaluating network...
              precision    recall  f1-score   support

   with_mask       0.99      0.99      0.99       383
without_mask       0.99      0.99      0.99       384

    accuracy                           0.99       767
   macro avg       0.99      0.99      0.99       767
weighted avg       0.99      0.99      0.99       767
```

Training statistics



Plot generated after training



Detection of no mask amongst multiple people

ACKNOWLEDGMENT

We would like to thank our professor Prof. Lokesh Kumar R (Associate Professor Grade 1) for providing us the necessary guidance to prepare the prototype of the model and successfully achieve our learning objectives. We would also like to thank Vellore Institute of Technology for providing us with the resources and support to present this work of ours to the research community.

REFERENCES

[1] R. N. S and M. N, "Computer-Vision based Face Mask Detection using CNN," 2021 6th International Conference on Communication and Electronics Systems (ICCES), 2021, pp. 1780-1786, doi: 10.1109/ICCES51350.2021.9489098.

[2] "Coronavirus Disease 2019 (COVID-19) – Symptoms", Centers for Disease Control and Prevention, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019ncov/symptoms-testing/symptoms.html. 2020.

[3] "Coronavirus — Human Coronavirus Types — CDC", Cdc.gov, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/types.html. 2020.

[4] W.H.O., "Advice on the use of masks in the context of COVID-19: interim guidance", 2020.

[5] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector", arXiv.org, 2020. [Online]. Available: https://arxiv.org/abs/2005.03950. 2020.

[6] Suvarnamukhi and M. Seshashayee, "Big Data Concepts and Techniques in Data Processing", International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 712-714, 2018. Available: 10.26438/ijcse/v6i10.712714.

[7] F. Hohman, M. Kahng, R. Pienta and D. H. Chau, "Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers," in IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 8, pp. 2674-2693, 1 Aug. 2019, doi: 10.1109/TVCG.2018.2843369.

[8] Kanan and G. Cottrell, "Color-to-Grayscale: Does the Method Matter in Image Recognition?", PLoS ONE, vol. 7, no. 1, p. e29740, 2012. Available: 10.1371/journal.pone.0029740.

[9] Opencv-python-tutroals.readthedocs.io.2020. Changing Colorspaces — Opencv-Python Tutorials 1 Documentation. [online] Available at:https://opencv-python-tutroals.readthedocs.io/en/latest/py tutorials/py imgproc/py colorspaces/py colorspaces.html. 2020.

[10] M. Hashemi, "Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation", Journal of Big Data, vol. 6, no. 1, 2019. Available: 10.1186/s40537-019-0263-7 . 2020.

[11] S. Ghosh, N. Das and M. Nasipuri, "Reshaping inputs for convolutional neural network: Some common and uncommon methods", Pattern Recognition, vol. 93, pp. 79-94, 2019. Available: 10.1016/j.patcog.2019.04.009.