Di-Mask : Distance Estimator and Mask Detector

Sonali Suri
sosuri@ucsc.edu
University of California, Santa Cruz

***Abstract*—Machine learning (ML) is a technology used on a large scale and helps us change the world. Machine learning focuses on developing computer programs that can access data and use them to learn for themselves. The learning process begins with observations or data, such as examples, direct experience, or instruction, to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly. On the other hand, visualization helps us in a better understanding of data and results. In today's time, COVID prevention has become an important concern for everyone. In this paper, I propose Di-Mask: A Realtime Distance Estimation and Mask Detection Pipeline, which uses Machine Learning to help prevent spread of COVID-19.**

***Index Terms*—face detection, mask detection, COVID, etc.**

I. Introduction

THe virus that causes COVID-19 spreads easily among people through the respiratory droplets released when someone with the virus coughs, sneezes or talks. Data has shown that it spreads mainly from person to person among those in close contact (within about 6 feet, or 2 meters). These droplets can be inhaled or land in the mouth, nose or eyes of a person nearby. In some situations, the virus can also spread through airborne transmission i.e. when a person is being exposed to small droplets or aerosols that stay in the air for several minutes or hours. But people nowadays are ignoring the major cause of transmission and continue to leave their houses without wearing masks and fail to maintain the social distancing knowingly or unknowingly. Thus, it is important to keep track of distance between people and identify people who are not wearing masks, especially in public places. In this paper, we present Di-Mask, Realtime Distance Estimation and Mask Detection Pipeline. Di-Mask uses Viola Jones for face detection and deep learning algorithm for mask detection and Euclidean distance in pixel space for estimating distance between two persons. There are some limitations to this approach and we use another approach for face detection for better results using Google Cloud Vision API for face detection. Though the approach using Google Cloud API provides better results, it is slower than the first approach. The rest of the paper is as follows: In Section III we go over related work on Face Detection and distance calculation. In Section IV we formulate the problem and give an overview of our method. In Section V we go over the Dataset. In Section VI we briefly describe our Di-Mask pipeline. Then we go over the results that our model achieved in Section VII. In Section VIII, we go through the proposed algorithm using Google Vision API and Section IX has the experimental results for the same.

II. Motivation

As per the data sources, Fig.1. represents the statistics representing the number of people who do not wear masks while leaving their houses. [2] More than 80% people do not wear mask while leaving their house in Denmark,around 50% people in Canada and so on. Thus, it is important to keep the track of people who do not wear masks and also detect the distance between them.

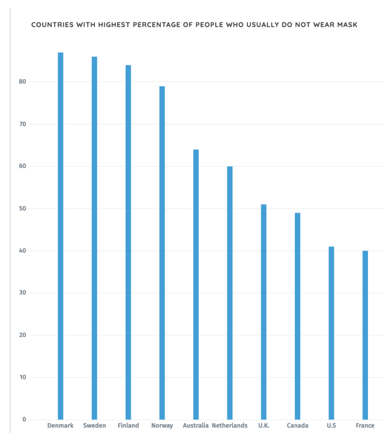

Fig.1. Illustrations of country-wise distribution of percentage of people who do not wear mask usually while leaving their houses.

III. Related Work

Mask detection problem has recently gained a lot of attention as a measure to prevent COVID-19. Mask detection problem can be broken down into face detection in the frame and mask classification in the face region. Face detection is not a new problem. In 2001 [1] Viola Jones algorithm gained ground-breaking results for real time face detection using combination of weak classifiers. After 2012, Neural networks gained popularity. Since then, many researchers started exploring Neural network solutions to solve real time face detection problem [2], [3]. All the previous research surround around face detection and face recognition. There was no research around face mask detection problem. Recently, COVID has pushed researchers to start solving face mask detection problem. Distance estimation problem can be broken in detecting objects and estimating distance between the two objects. In 90s, researchers developed techniques to determine distance of objects in images by taking reference of an object whose distance is already known. Later in 2000s, researchers did research estimation experiments with the help of hardware that would keep track of object in real world. Recently, after a breakthrough in object detection by state-of-art techniques it has been possible to have accurate distance estimation.

IV. Proposed algorithm

The Algorithm shown in Fig.4. uses Viola Jones algorithm for face detection.

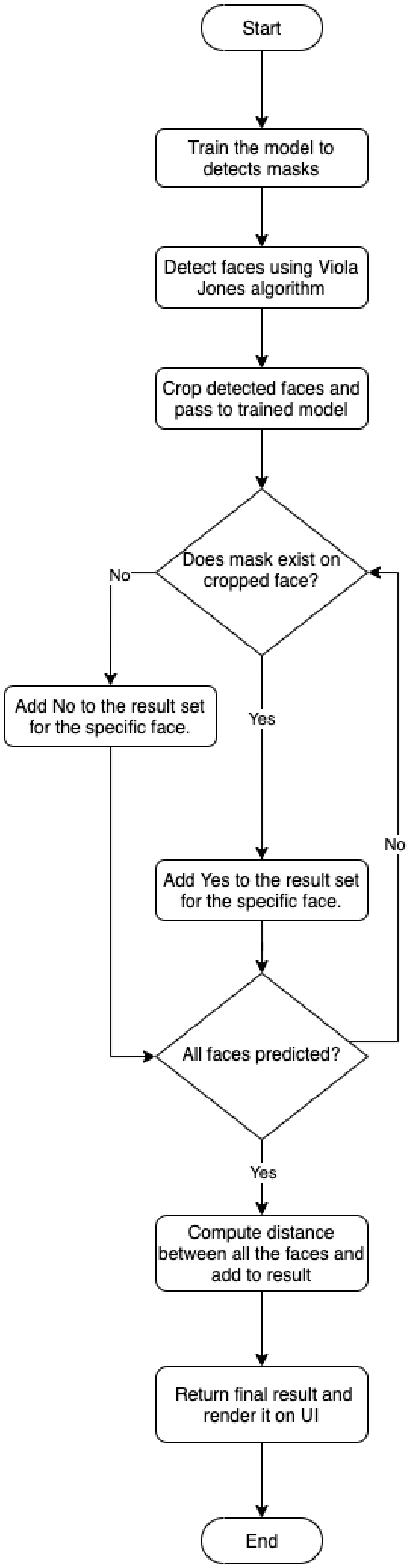

Fig.4. Proposed Algorithm : Using Viola Jones for Face detection

Di-Mask works on four important principles i.e. training the model for mask detection, face detection using Viola Jones, mask detection for the detected face and calculating the distance between the detected faces. This can help the government track, people without masks at public places and help the countries keep a check on the percentage of people not wearing masks. If the rate is observed to be increasing above some certain threshold, then specific actions should be taken to avoid the future spread of the COVID virus.

V. Dataset

Fig.2. Subset of training data with face mask

The dataset used is from [7] which is artificially created face mask images. The data set consists of 1376 images. 690 images out of 1376 images are with mask images and 686 images out of 1376 images are without mask images. Fig.2. and Fig.3. shows some example of training images which has both with face mask and without mask images respectively.

Fig.3. Subset of training data without face mask

VI. Approach

Creation of Di-Mask pipeline is a four-step approach:

1) Training a face mask classification technique : In training step, we use Google Cloud AutoML service to train the classifier. AutoML technique includes raw dataset as input and deployable ML model as output. Fig.5. shows the pipeline for Di-Mask.

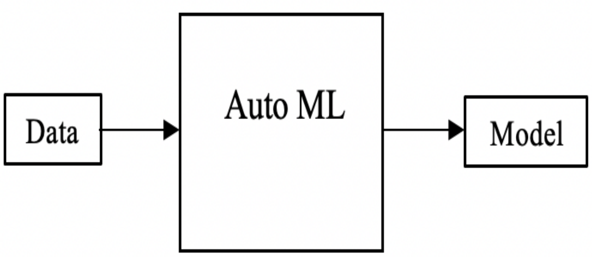

Fig.5. Di-Mask pipeline

2) Using Viola Jones for face detection :
   a) Haar Feature Selection : All human faces share some similar properties and these regularities can be matched using Haar Features. Some of the common properties include: a. The nose bridge region is brighter than the eyes. b. The eye region is darker than the upper-cheeks. Location and size: eyes, mouth, bridge of nose Value: oriented gradients of pixel intensities
   Fig.6. [1] represents some of the Haar features where bright area may correspond to adreas like nose bridge and dark areas may correspond to features like eye region. So, each feature is related to a special location in the sub-window of the face.

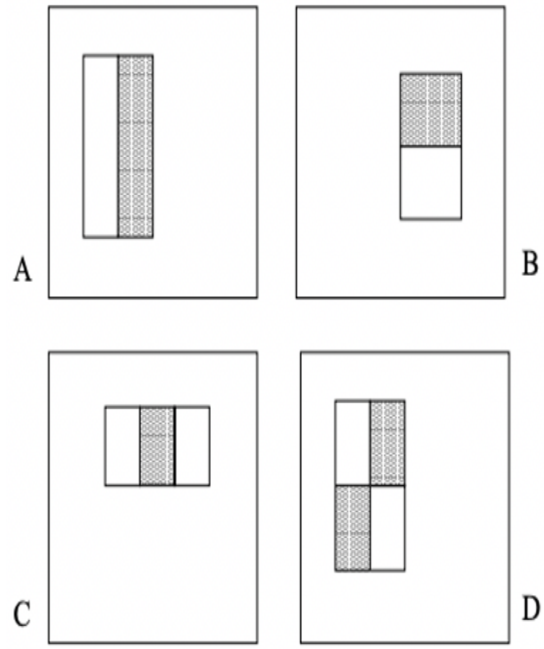

Fig.6. Haar features

   b) Creating an Integral Image : An integral image is the image representation which evaluates the rectangular features in constant time. Since each feature's rectangular area is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, three-rectangle feature in eight, and any four-rectangle feature in nine.
   c) Adaboost Training : The object detection framework employs a variant of the learning algorithm AdaBoost that selects the best features and trains classifiers to use them.
   d) Cascading Classifiers : In cascading, each stage consists of a strong classifier such that all the features are grouped into several stages where each stage has a certain number of features. The job of each stage is to determine whether a given sub-window is definitely not a face or maybe a face.

   When Viola Jones is used to detect a face with mask, it is still able to recognize it as a face though with low probability since it makes a decisions through the summation of the values returned by all of the features weighted by their prominence.
3) Using trained face mask classifier for detecting mask in the detect face : The cropped face is passed to the model which detects or determines if the face has mask or not. Fig.7. represents the mask classifier on the live stream data which is the part of our actual testing with new data.
4) Distance calculation : Determine the distance between all the detected faces. We get the midpoint co-ordinates

of detected faces. After detecting the midpoint co-ordinates, we calculate the distance between the mid-point co-ordinates of each detected face in pixel space. We approximate the distance in real world by translating the pixel in space with the PPI (Pixels per Inch) of the camera taken relative to image dimensions, in our experiments we have MAC Book 2019 Pro camera with 227 PPI. For our experiments, we have set the image dimension 300 * 225 where 300 is the width and 225 is the height, and we compute the value for PPI as 113 for this setup.

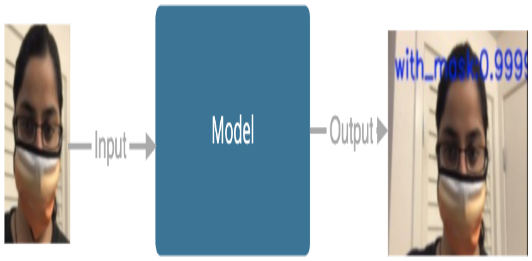

Fig.7. Mask Classifier

VII. Experiment Results : By using Viola Jones for face detection

Viola-Jones is able to detect the faces without masks with great accuracy of about 93.5% since it used about 200 features for face detection. But in scenarios where the face is covered with a mask and the person is looking sideways, the detection is not accurate as most of the features are then not visible for detection. Thus, the accuracy for scenarios with masks reduces around 72.8% accuracy for faces with masks. In the below images status N means no mask and Y means mask detected.

Below are the scenarios tested with this approach:

1) Face without mask: Fig.7. shows the results for an image without mask.

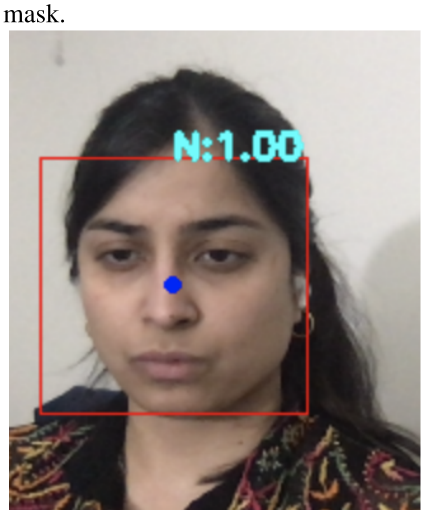

Fig.7. Without mask face : shows status as N

2) Face without mask, looking sideways and with glasses: Fig.8. shows the results for a face without mask, looking sideways and wearing glasses.

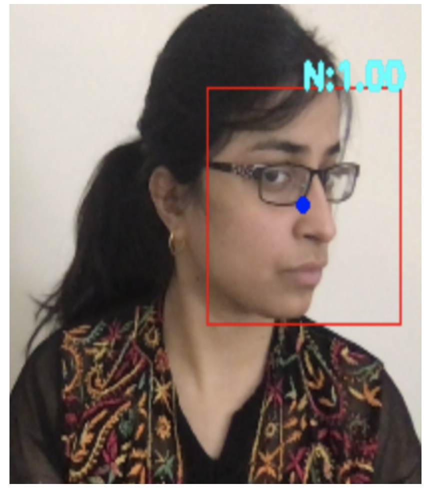

Fig.8. Without mask ,sideways and with glasses: shows status as N

3) Face with mask, and with glasses: Fig.9. shows the results for a face with mask and wearing glasses.

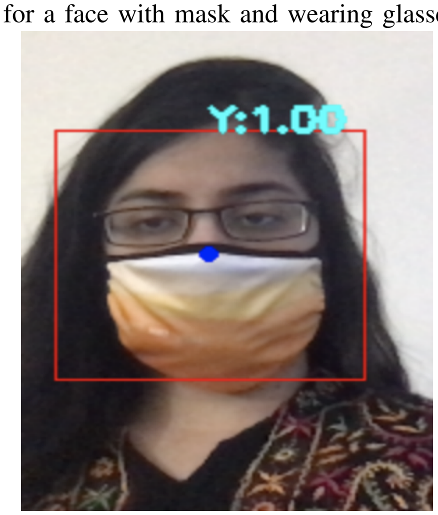

Fig.9. With mask: shows status as Y

4) Mask detection for two faces without masks: Fig.10. shows the results for two faces without mask. Individual mask status is shows for each face and it also shows the distance between the two faces.

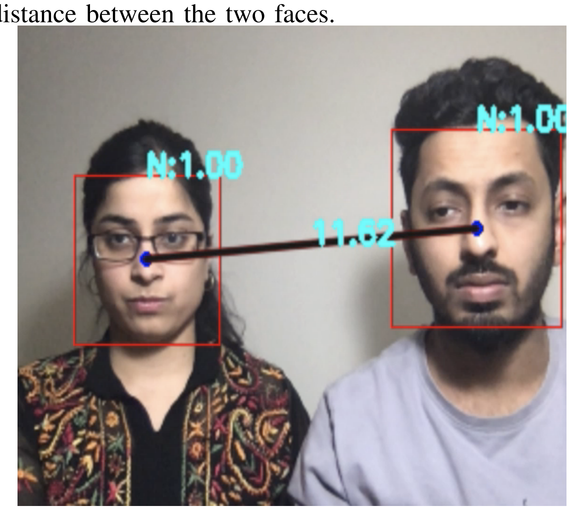

Fig.10. Mask detection for two faces, without mask showing statuses as N and N respectively with distance of 11.02 inches between the faces.

5) Mask detection for two faces, one with mask and one without masks: Fig.11. and Fig.12. shows the results for two faces where one person is wearing the person and the other person is not wearing the mask. Individual mask status is shows for each face and it also shows the distance between the two faces.

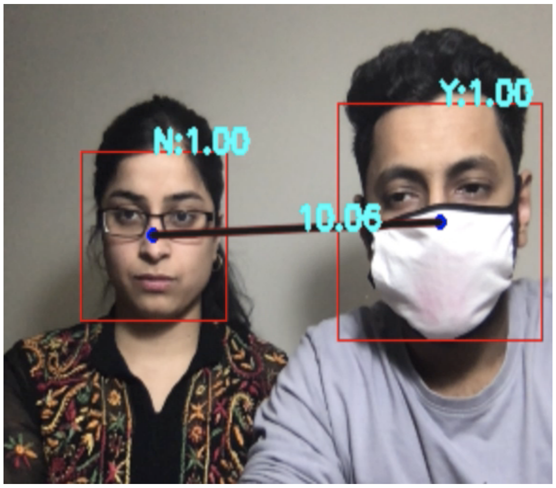

Fig.11. Mask detection for two faces, one with mask and one without mask showing statuses as N and Y respectively with distance of 10 inches between the faces.

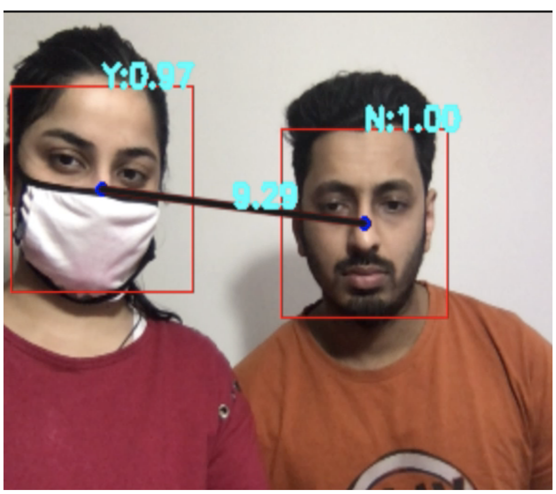

Fig.12. Mask detection for two faces, one with mask and one without mask showing statuses as Y and N respectively with distance of 9.29 inches between the faces.

6) Mask detection for faces with features similar to a face

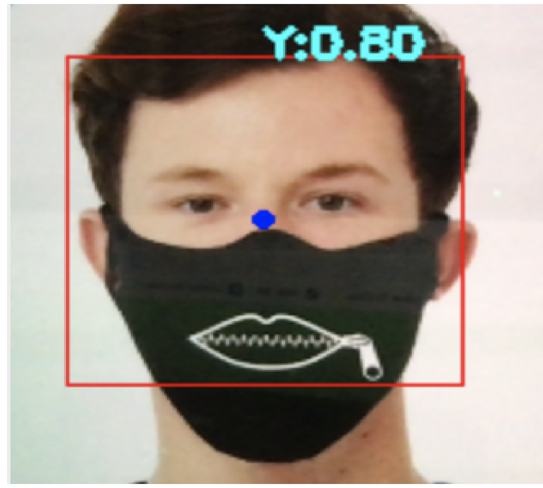

Fig.13. Result shows Y

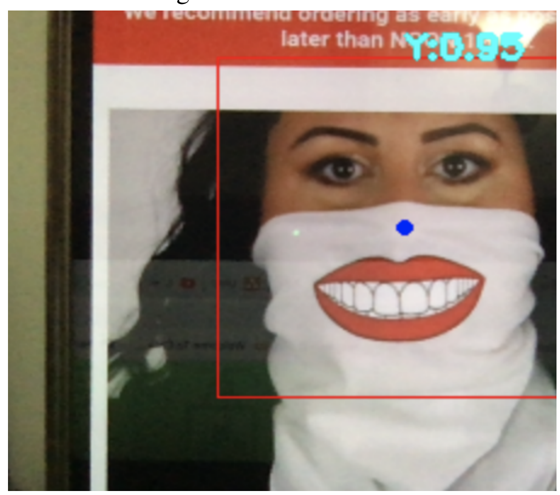

Fig.14. Result shows Y

7) Mask detection multiple people wearing mask : Fig.13. represent multiple people wearing mask with the statuses showing as Y for each of the face detected.

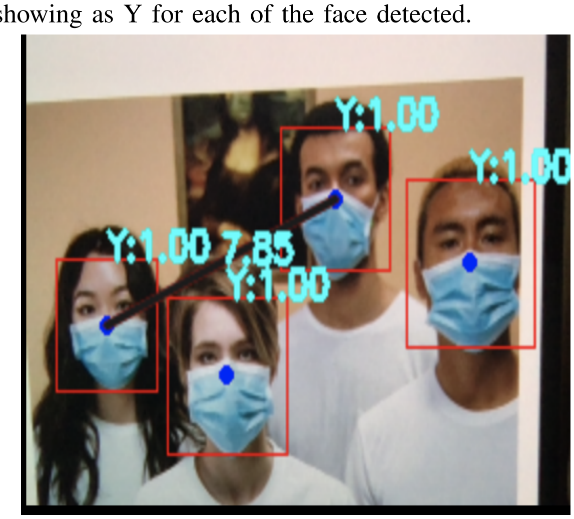

Fig.15. Mask detection for two faces, one with mask and one without mask showing statuses as N and Y respectively with distance of 10 inches between the faces.

8) Mask detection for a crowd : Fig.14. represents the mask detection for a crowd.

Fig.16. Mask detection for crowd

9) Results for distance: Table.1. represents the actual distance between the two faces and the calculated distance by Di-Mask.

Sr.No	Actual Distance	Di-Mask Distance
1	5	4.78
2	6	6.118
3	7	7.10
4	8	7.92
5	9	8.94
6	10	10.01

Table.1. Actual distance between faces vs computed distance

The limitations of this approach is that, the face detection is not accurate when a person is looking sideways and is wearing a mask as the prominent features are then not visible. Thus, we can use the approach mentioned in the next section.

VIII. Proposed Algorithm : Using Google Vision API

Since the face detection doesn't work accurately when a person wearing mask is looking sideways, we can use Google Vision API for face detection and pass the detected faces to our model and compute the distance between the faces. Fig.15. helps us to understand the steps for this approach.

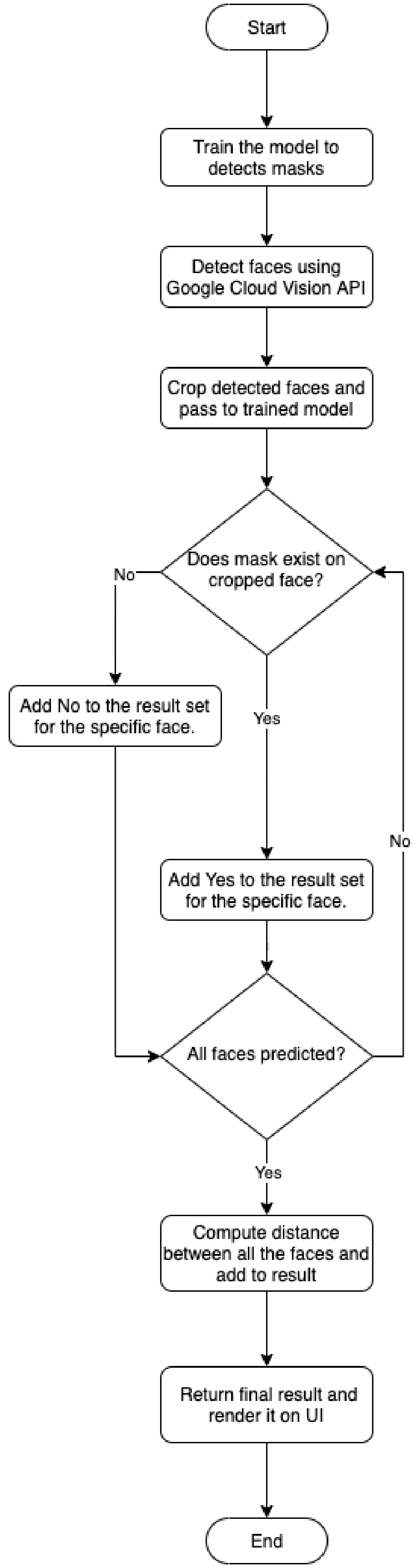

Fig.17. Proposed Algorithm : Using Google Vision API for face detection

Following are the steps for this approach:

1) Training a face mask classification technique : Similar to Section VI,in training step, we use Google Cloud AutoML service to train the classifier. AutoML technique includes raw dataset as input and deployable ML model as output.
2) Use Google Vision API for face detection : This API can detect multiple faces within an image along with the associated key facial attributes such as emotional state. We pass the live video frame to the Vision API and for each image frame, the response includes a bunch of metadata about the detected faces, which include coordinates of a polygon encompassing the face.
3) Mask detection : Once the faces are detected by the Google Vision API, these faces are passed to the Machine learning model trained for detecting masks. For each face, the model check if there is a mask on the face or not and accordingly adds it to the result.
4) Distance calculation : Similar to Section VI, we get the midpoint co-ordinates of detected faces. After detecting the midpoint co-ordinates, we calculate the distance between the midpoint co-ordinates of each detected face in pixel space. We approximate the distance in real world by translating the pixel in space with the PPI (Pixels per Inch) of the camera taken relative to image dimensions and add the distance to the result.

IX. Experiment Results : By using Google Vision API for face detection

1) Face with mask, sideways and with glasses: Fig.18. and Fig.19 shows the results for a face with mask, looking sideways and wearing glasses.

Fig.18. With mask ,sideways and with glasses: shows status as Y

Fig.19. With mask ,sideways and without glasses: shows status as Y

2) Fig.20 shows results for two people, one with mask and one without mask

Fig.20. Status shows N and Y respectively and the distance of approximately 10.27 inches

This approach also provide accurate results for all of the tests mentioned in Section VIII for mask detection. The results for distance calculation is similar to the details mentioned in Table.1.

X. Conclusion and Future Work

Through this paper, Di-Mask was able to detect masks and calculate the approximate distance between people. This paper implemented two approaches: detecting faces through the Viola-Jones algorithm and the other approach using the Google Vision API. In terms of performance, the approach that uses Viola-Jones for face detection is faster than the latter. But in terms of accuracy, the approach using Google Vision API for face detection provides better results for scenarios wherein the face is looking sideways. Depending upon the use case, either of the approaches can be chosen. Di-Mask can be used at public places, or entrances of social places do determine if social distancing is being followed. As a part of future work, we can enhance the algorithm for calculating the distance between two persons by exploring various approaches.

XI. References

[1] https://en.wikipedia.org/wiki/Viola%E2%80%93Jones _object_detection_framework

[2] https://www.nytimes.com/interactive/2020/07/17/upshot /coronavirus-face-mask-map.html

[3] https://www.base64-image.de/

[4] https://cloud.google.com/automl

[5] https://d3js.org/

[6] M. N. Chaudhari, M. Deshmukh, G. Ramrakhiani and R. Parvatikar, ”Face Detection Using Viola Jones Algorithm and Neural Networks,” 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697768.

[7] https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/