Face Mask Detection and Crowd Estimation Using MobileNetV2

By Jobin P. Jose[1], Joshua Jacob Chacko[2], Neethu Maria Mathew[3], Prof. Tom Kurian[4]

[1-4]Students & Guide, Computer Science and Engineering, Amal Jyothi College of Engineering, Kottayam

---***---

Abstract: - During this pandemic WHO has made it compulsory to wear masks, to maintain social distancing and to avoid social gathering. But there are people out there who consider it as hoax and refuse to wear mask, and follow the COVID-19 protocols. It is not an easy job for the security officials to keep a check on everybody in a crowd for masks and to get a count of the crowd. To make things easier we came up with a simple solution to identify if a person is wearing a mask or not and also counts the number of people in a crowd. In our study we develop a program using deep learning to count and detect the masked faces. To detect the mask, we are going to implement in 2 steps: -

1. Training the model
2. Deploying the trained model

We use MobileNetV2 network to train the model OpenCV to implement face detection module.

Index terms – MobileNetV2

I. INTRODUCTION

In the situation of COVID-19, wearing a mask and not over-crowding in places has turned out to become a life style. In order to stop the spreading of the virus we must wear our masks properly covering our nose and mouth. Many people around us don't wear their masks properly which in turn escalates the spread of virus. According to the updated COVID-19 protocols, our government has also strictly prohibited the gathering of more than 5 people in a public place and not more than 20 people in a It consists of 2 types of blocks. Block 1 is a residual block with stride 1. Block 2 is used for downsizing with stride 2. For both the blocks there are 3 layers.

- The first layer is with ReLU6 1x1 convolution.
- Depth wise convolution is the second layer.
- Another 1x1 convolution without any non-linearity is the last layer.

MobileNet is also called the lightweight deep neural network because of the reduced number of parameters w.r.t the network with convolutions with the same depth in nets. Due to a smaller number of parameters, it is faster than other neural networks.

The speed and power consumption of the MobileNetV2 is correlative to the number of MACs i.e., *Multiply-Accumulates* which is a measure of the number of fused Multiplication and Addition operations.

function. So, to keep a check on the count of people we have developed a program which also allows us to detect whether they have a mask or not.

II. METHODOLOGY

To implement our model, we have built a face mask detector and crowd estimation in Python using TensorFlow, Keras, OpenCV and MobileNetV2. We can apply our model on a live video camera and with a bit more adjustments our model can be integrated with CCTV cameras, to identify and detect the number of people in the crowd and to examine if they have worn a mask or not.

Our system can be in real time applications to ensure safety and to keep a check on those who violate the protocols.

We have used MobileNetV2 architecture which makes the model accurate, efficient and easier to deploy to embedded systems.

MobileNetV2:- MobilenetV2 was released as a part of TensorFlow- slim image classification library. It is an efficient feature extractor for segmentation and object detection. It is a Convolutional Neural Network architecture based on inverted residual structure where the residual connections are between bottleneck layers.[1]

By using MobileNet we are generating 2 types of models: -

- Base Model, whose output will be passed into the normal model.
- The normal model also called the Head Model.

In our program, the Head model is set to 128 neurons and the pool size is set to 7x7 and finally setting ReLU as the go to activation function.

ReLUu:- ReLU or Rectified Linear Unit Activation Function is a piece wise linear function which will give output 0 if input negative and if input is positive will output the input directly, hence it gives an output that has a range from 0 to ∞ [4]. It is the most commonly used activation function in deep learning models. ReLU uses the simple formula:

f(x)=max (0, x)

Adam Optimiser:- Adam or Adaptive Moment Estimation is an optimisation algorithm used to train deep learning models. It can handle sparse gradients on noisy problems. It is an adaptive learning rate optimisation algorithm which is a combination of RMSprop and Stochastic Gradient Descent with momentum. For different parameters it can compute individual learning rate.[2]

It is the usual optimiser for image prediction methods.

In our program we have defined *Adam optimiser* by setting the parameters, learning rate and decay *(learning rate / epochs)*.

SoftMax Activation:- Softmax is an activation function that scales a vector of numbers into vector of probabilities. It is great for dealing with multiple class classification problem and as it gives back the confidence score for each class. Probabilities returned by softmax will add up to 1. The class that is predicted will be the item where the probability is the highest.[3]

With respect to our program, this activation function recognises if the person is with or without a mask by taking both probabilities and considering the highest among them.

CAFFE Model:- *Caffe* is based on *Single Shot Multibox Detection (SSD).* Convolutional Architecture for Fast Feature Embedding (CAFFE) is a deep learning framework that allows us to create image classification and segmentation models. CAFFE defines a net layer by layer in its own model schema. The network defines the entire model bottom to top from input data to loss.

Initially, the users create a model and save that model as plaintext and after training it returns the model using the Caffe, which is saved as a *CAFFEMODEL* file.

Dataset:- Dataset is used to determine or learn the optimal combinations in order to generate a trained and good predictive model.

For our project, we have 2 datasets: -

- With Mask
- Without Mask

We have taken the images from Kaggle, few open-source libraries and Google images. We train our model using these datasets. The dataset that we used of masked images were not morphed. We collected around 1900+ images.

Image Augmentation:- It is done using the image data generator using Keras and it helps in augmenting the images in dataset in real-time while the model is training. Random transformations can be applied on the images when it is given as the input to a model. It helps in reducing the memory used and also makes the model robust.

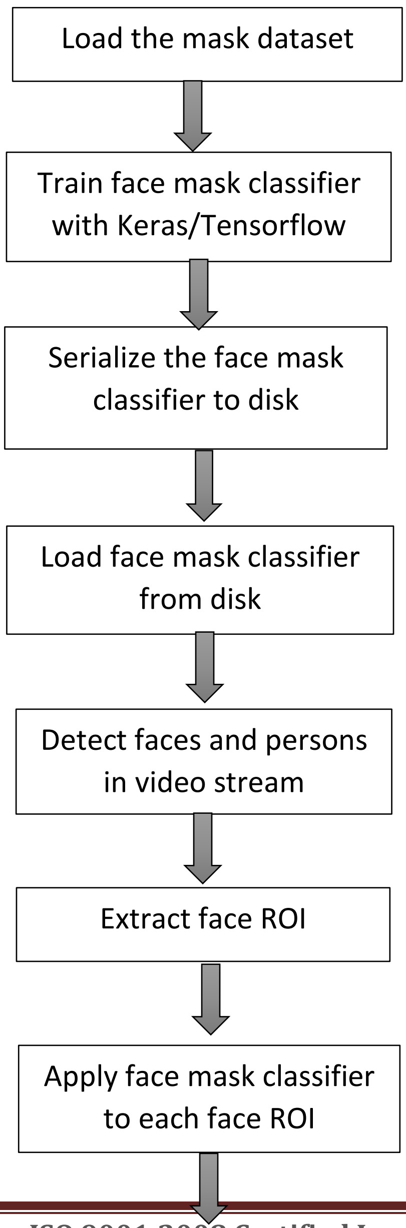
It can artificially expand the size of a dataset by modifying the already present images.

We have the used the parameters like *rotation, zoom, width_shift, height_shift, shear and flip.*

III. PROPOSED SYSTEM

Our program consists of 2 phases:

1. Training
2. Deploying the trained model


Load the mask dataset
Train face mask classifier with Keras/Tensorflow
Serialize the face mask classifier to disk
Load face mask classifier from disk
Detect faces and persons in video stream
Extract face ROI
Apply face mask classifier to each face ROI

Determine "Mask"/"Without Mask" and count the number of people

For training the model we have 2 datasets:

a. Mask
b. Without Mask

The "Mask" dataset contains images of people wearing masks. And the "Without Mask" dataset contains images of people without mask.

We convert the image into an array for the pre-processing and input that pre-processed images into the 'baseModel' of the MobileNetV2.

And for the *train_test_split*, we took *validation_size=0.20 (20%)* and the remaining *0.80 (80%)* for the *train_size*.

We initialized the initial learning rate as ***e^{-4}*** and number of epochs as ***20***. Initial learning rate should be less so that the loss can be calculated properly and better accuracy is obtained.

We load the MobileNetV2 network ensuring the head layers' sets are left off and initialize the weight as "*imagenet*" which is a pre-defined weight for the images. The head of the model will be placed on top of the *baseModel*. The *headModels*' input will the *baseModel.* Then placing the *headModel* on top of the *baseModel* we will get our actual model that we will train.

After training we display a nicely formatted classification report as shown in *figure 6.*

After training we will deploy the trained model. For that we will grab the dimensions of the frame and then construct a blob from it and then pass the blob through the network and obtain the face detections. Now we will initialize our list of faces, their corresponding locations, and the list of predictions from our face mask network. Then we extract the confidence associated with the detection and filter out the weak detections by ensuring the confidence is greater than the minimum confidence. We then extract the face ROI, convert it from BGR to RGB channel ordering and pre-process it. Then we load the face mask detector model from the disk and grab the frame from the threaded video stream. We detect the faces in the frame and determine if they are wearing face mask or not and also count the number of people in the same frame under the category "*Live Count*". For counting we initialize an array called the *object_id_list,* which is updated and the count is increased whenever a person is detected in the frame. For each person an *id* is provided.

IV. EXPERIMENTAL RESULT AND ANALYSIS

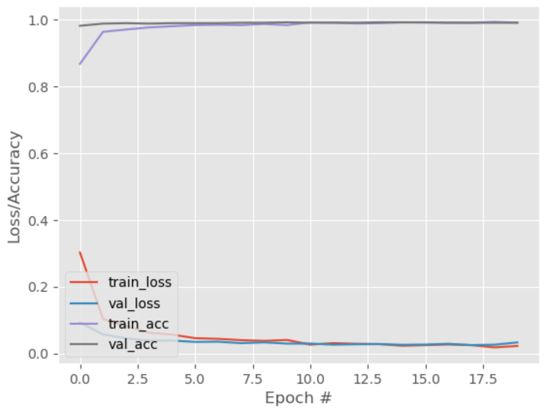


Figure 1: Training loss and Accuracy Plot

The above plot shows the result after training our model. The *x-axis* represents the number of epochs in our training and the *y-axis* depicts the loss and accuracy of our model.

The purple line in the graph illustrates the training accuracy, which is noticed to increase with the number of epochs whereas the grey line denotes the validation accuracy, which remains constant on the trained model.

The red line shows the training loss, which show that as the loss decreases accuracy increases, whereas the blue line shows the validation loss and a slight variation is noticed.

	Precision	**Recall**
With_Mask	0.99	1.00
Without_Mask	1.00	0.99
Accuracy	1.00	1.00

Table 1. Training Results

The above image shows the result that we obtained after training the model. Here, after training, in the case of ***with_mask*** we obtain the precision as 0.99 and the recall as 1.00. In the case of ***without_mask,*** we obtain the precision as 1.00 and recall as 0.99. The overall accuracy obtained after training is 1.00.

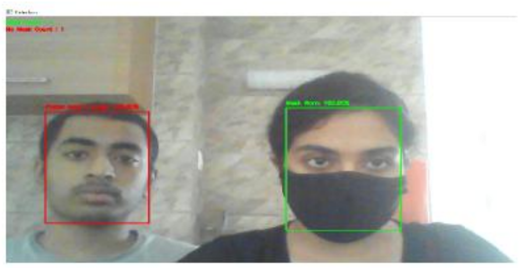


Figure 2: With and without mask

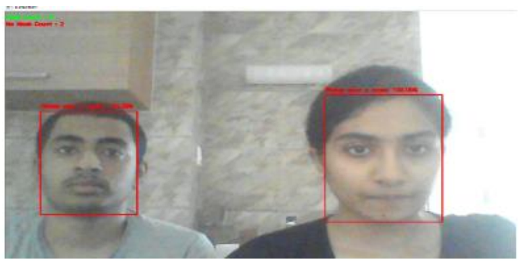


Figure 3: Without mask

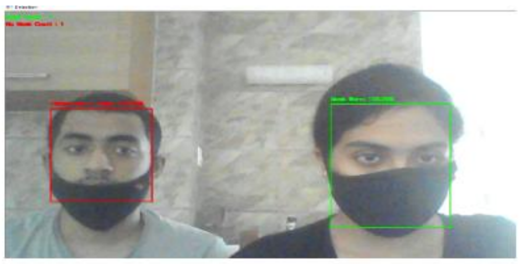


Figure 4: Mask not worn properly

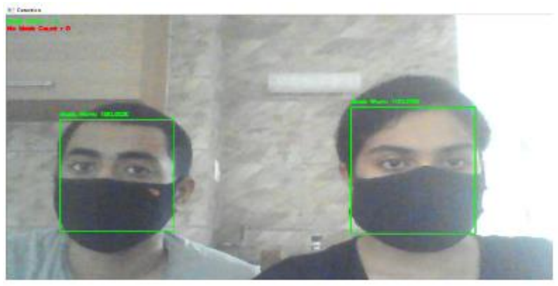


Figure 5: Mask worn properly

V. CONCLUSION

In this paper, we have detected if people in a crowd are wearing a mask or not and the number of people in the crowd. We have done this using MobileNetV2 as it faster and requires less parameters than other neural networks. The counting of number of people in a crowd is done using MobileNetSSD. This model can be used by government authorities, shopkeepers, restaurants etc so
314, doi: 10.1109/ICEMI.2017.8265801.

as to keep a check on those who are violating the protocols.

REFERENCES

[1] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[2] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 2018, pp. 1-2, doi: 10.1109/IWQoS.2018.8624183.

[3] M. Bhuvaneshwari and E. G. M. Kanaga, "Convolutional Neural Network for Addiction Detection using Improved Activation Function," 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 996-1000, doi: 10.1109/ICCMC51019.2021.9418022.

[4] Y. Zhang and X. Chen, "Lightweight semantic segmentation algorithm based on MobileNetV3 network," 2020 International Conference on Intelligent Computing, Automation and Systems (ICICAS), 2020, pp. 429-433, doi: 10.1109/ICICAS51530.2020.0009.

[5] L. Qiming, H. Ligang, X. Qiuyun, Y. Tongyang, G. Shuqin and W. Jinhui, "The design of intelligent crowd attention detection system based on face detection technology," 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), 2017,pp.31