

RESEARCH ARTICLE OPEN ACCESS

Covid-19 Face Mask Detector for Institutions

Prof. Dnyaneshwar Choudhari*, Mane Yash Gautam**, Ajinkya raje***, Shriraj kanase****, Siddhi Dhamne*****

* (Pimpri Chinchwad College of Engineering and Research, Pune

Email -dnyaneshwar.chaudhari@pccoer.in)

** (Pimpri Chinchwad College of Engineering and Research, Pune

Email- yashmanslp979@gmail.com)

*** (Pimpri Chinchwad College of Engineering and Research, Pune

Email- araje424@gmail.com)

**** (Pimpri Chinchwad College of Engineering and Research, Pune

Email - shrirajkanase09@gmail.com)

***** (Pimpri Chinchwad College of Engineering and Research, Pune

Email- syv6103@gmail.com)

Abstract

As the spread of Covid-19 is increasing rapidly due to which the restrictions on the people are increasing but keeping restrictions on people’s daily routine is not a permanent solution as we cannot stop the growth and progress of our society. We have to adapt or change ourselves according to the situation. As people will go back to their workplace and institutions we need to ensure the safety of people. To ensure the safety new plans and policies are implemented daily in which Covid Tracking tools are mainly used. As wearing a face mask in a day to day life has become compulsory so it is important to keep a close watch on people wearing a face mask in public places and workplace surrounding to ensure the safety and avoid the spread of Covid-19. To do this Covid tracking tools such as face mask detection system are used. This system identifies whether a person is wearing a mask or not. This system uses existing surveillance systems with Neural network algorithms to track people without a face mask with the help of Artificial intelligence which mainly includes machine learning, deep learning frame works and Image processing

Keywords - Artificial intelligence, Machine learning, Neural network, Surveillance system, Face mask detection system, Image processing.

I.INTRODUCTION

To detect whether people are wearing mask or not? We are developing a face mask detector system that is able to distinguish between faces with masks and faces with no masks through surveillance systems (CCTV). This system first detects the face and then detects the mask on the face. Which is done by using SSD[1] for face detection and the neural network to detect the face mask .This implementation is done on Images and Live stream videos. To ensure the accuracy of the system it is trained and tested with the help of Data sets which contains images of people with mask and without mask.

II.PROPOSED SYSTEM

Now for creating the model we need to divide the developing process in to 2 parts. First Training and Testing the model and then applying model in open cv.

A) Datasets

We have nearly 1900 images of people with masks and without masks.. These images are taken from Kaggle and various open source datasets. And the number of images with mask and without mask are equal. We have split our dataset into two parts. Where 85% of data is used in training and 15% of data is used in testing. We split the dataset to avoid overfitting and thus it helps in optimisation of training dataset accuracy.

B) Training, Testing and Working

To create and train a model, we need to use a dataset which contains images of people with mask and without mask. To work with a deep learning model, we need to convert these images into arrays. Then we need split data into two parts for training and testing. For training, we follow CNN (Convolutional neural networks) [2]. We have used Keras along with Tensorflow [3] to train our model. We give the initial learning rate as 0.0004. We keep the learning rate lower because the loss will be calculated properly. Which gives better accuracy. We give 22 epochs and keep the batch size 32. We created two models. The Base model and the Head model. We use

MobileNet to create a base model. First we take the input and convert the images into Numpy arrays [4] and corresponding images are reshaped into (224,224,3). Now this image augmentation is done with the help of an Image Data Generator. An Image data generator creates many images from a single image by rotating, shifting, zooming and flipping. It increases the accuracy significantly, then we send it through Mobilenet. Here we use MobileNet because MobileNet is very faster in process and they use fewer parameters and the further we do Maxpooling and then Flattened it and create a fully connected layer and give the output. To train the model after creating a base model, we are creating a head model. In the head model, we are passing base model output as the first parameter or input. Then we create a pool of pool size (7,7). Then we flattened this and then we added a dense layer of 128 neurons and the activation layer here is Relu. Then we use dropout just to avoid overfitting of the model. And finally, we get the output which is of 2 layers, that is with mask and without mask. Softmax activation is used as it takes the input and gives probability based values between 0 and 1. And finally, we are saving the model in h5 format. So we can run and check the accuracy.



Figure 1- Training and Loss Accuracy

Now we have a model trained to detect masks. For face detection, we have used the OpenCV library. The OpenCV includes a Deep Neural Network (DNN) module [5]. The DNN comes with pretrained Face detection CNN. To load the camera we use video stream function. With the help of this video stream, we can load multiple cameras. After implementing it will return the prediction and location. Now we store the prediction and location in a tuple. And unpack the tuple and we get the coordinates and with the help of coordinates we mark a rectangle around the face and we add a label of percentage which gives us the prediction whether the person is wearing a mask or not.

Now below we can see the working of the algorithm.

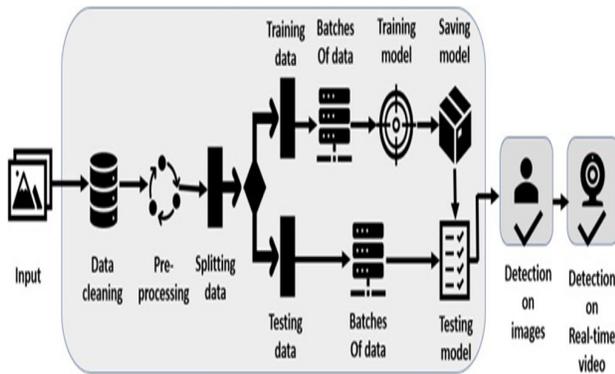


Figure 2- Working Algorithm



Figure 3 -With Mask

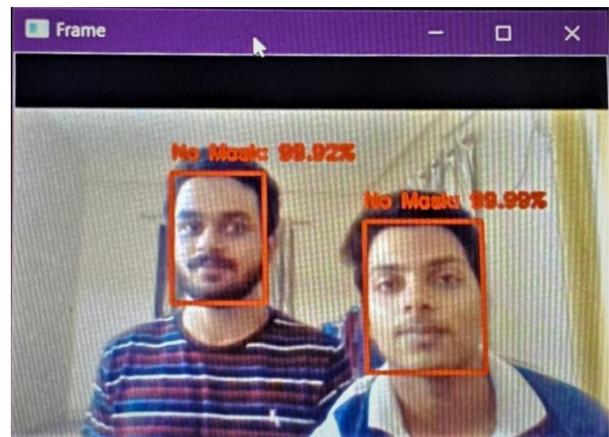


Figure 4- Without Mask

C) Results

We implemented it on livestream videos and videos with mask and without mask. we got accuracy around 99%. Which are more than enough for Face Mask detection system. Following results are shown below

III.Requirements

A)Software and Tools

- Windows 10(operating system)
- Python package Manager
- Python 3.6

B)Hardware

- 4 GB RAM (Minimum)
- 8 GB HDD (Minimum)
- Dual Core processor
- Surveillance Camera

IV.TOOLS AND TECHNIQUES

A) Keras ImageDataGenerator

Keras ImageDataGenerator[6] is used during training of model. While training the model it aguments the images in real time. Which helps to increase the accuracy of model. It's a very useful technique when limited no of dataset is available As it transforms images from available dataset. The augmentation techniques are

- a) Horizontal and Vertical Shift Augmentation
- b) Horizontal and Vertical Flip Augmentation
- c) Random Rotation Augmentation
- d) Random Brightness Augmentation
- e) Random Zoom Augmentation

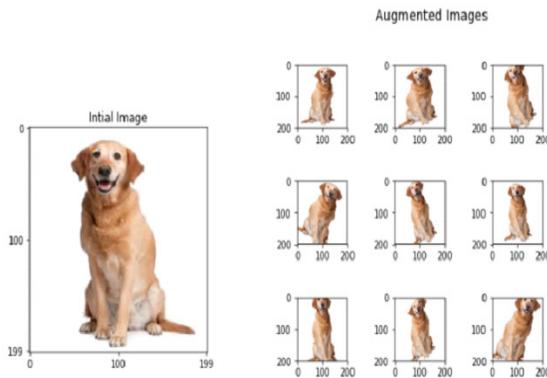


Figure 5 – Image Agumentation

B) MobileNetV2

In MobileNet[7] architecture the residual connection is from Bottleneck layer to another Bottleneck layer (Inverted Residual block) . So the residual blocks remains thin which helps in reduction of no of parameters and matrix multiplication and also increases depth.

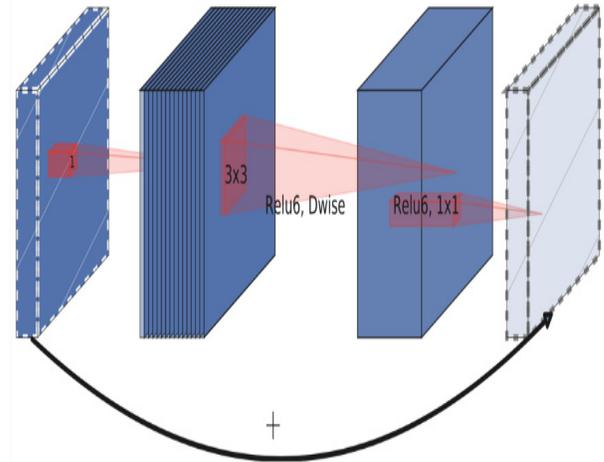


Figure 6 – MobileNetV2 (bottleneck with residual)

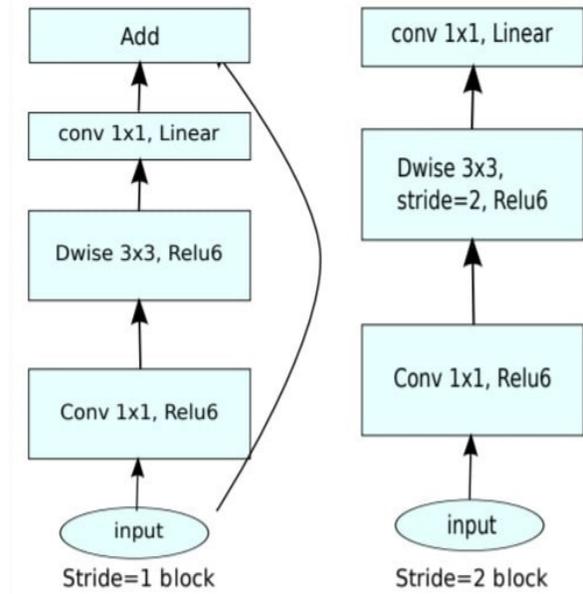


Figure 7- MobileNetV2 Architecture

V.CONCLUSION

To reduce the spread of COVID-19. We have developed a model which can be used in a face mask detection system. The model has an SSD architecture. Due to its high efficiency, it is easier to install the model on embedded systems.. The model was tested on images and videos and accuracy was nearly 99% as we used the MobileNet architecture, which has excellent precision and recall. The dataset we used consists of 1900 images which have an equal number of mask and without mask images. This Face mask detection system can be used at airports, institutions, companies and various public places where surveillance is necessary. It will definitely help to detect people violating rules by not wearing a facemask and in reducing the spread of Covid-19.

[7] M. Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2018

REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multiboxdetector," in European conference on computer vision. Springer, 2016, pp. 21–37

[2] Leonardo De Marchi ,Laura Mitchell, *Hands-On Neural Networks: Learn how to Build and Train Your First Neural Network Model Using Python*,2019,pp.65-80.

[3] Introduction to modules, layers, and models, Tensorflow Core.
https://www.tensorflow.org/guide/intro_to_modules

[4] Ivan Idris, *NumPy Beginner's Guide*, Second edition,2013.

[5] Deep Learning with OpenCV DNN Module: A Definitive Guide

<https://learnopencv.com/deep-learning-with-opencvs-dnn-module-a-definitive-guide/>

[6] Jason Brownlee, How to Configure Image Data Augmentation in Keras, Last Updated on July 5, 2019, Last Visited May 5,2021

<https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>