**MASK DETECTION**

**R M NARESH KUMAR**

[**1934025aiml@cit.edu.in**](mailto:1934025aiml@cit.edu.in)

**S SRIRAM Coimbatore Institute of Technology**

[**1934048aiml@cit.edu.in**](mailto:1934048aiml@cit.edu.in)Peelamedu,

**S DEIVARANI** Coimbatore 641-014,

[**deivarani@cit.edu.in**](mailto:deivarani@cit.edu.in)Tamilnadu, India

**ABSTRACT**

The goal of this program is to detect the person who is wearing mask and who is not using Deep Learning. The recognition is done through the camera. The program is trained with a large set of images under 2 categories [With Mask, Without Mask]. The model used here is a light weight one, which can also run in compact devices like mobile, tablets etc. The performance of this model is also relatively higher than other model around 85-90%

1. **INTRODUCTION**

In today’s pandemic situation, it’s always recommended to wear a mask where ever we go, but there are some people who are being very lethargic and careless, who don’t wear a mask. In order to find them we have come up with detection model which detects the person who is wearing mask and who is not.

Many precautionary measures have been taken to reduce the spread of this disease where wearing a mask is one of them. In this paper, we propose a system that restrict the growth of COVID-19 by finding out people who are not wearing any facial mask in a small organisation or school/college institution where all the public places are monitored with CCTV cameras.

It is hands free software that means just turn on the software, the other things will be taken care by the software itself. A deep learning architecture is trained on a dataset that consists of images of people with and without masks collected from kaggle. This program is in initial stage, we have also planned to update this software which enables the software to note the person’s name who are not wearing the mask and export that data to a xl sheet.

**2. RELATIVE WORK**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **ARTICLE** | **PUBLISHMENT** | **WORK** |
| 1 | A real time DNN-based face mask detection system using single shot multi-box detector and MobileNetV2 | Sustain Cities Soc. 2021 Mar; 66: 102692.Published online 2020 Dec 31. | SSDMNV2 approach uses Single Shot Multi-box Detector as a face detector and MobilenetV2 architecture as a framework for the classifier. |
| 2 | AI on the Edge : Face Mask Detection - Sergio Virahonda | Published on Code Project  Date : 25 Jan 2021 | Combining Keras, MobileNet V2, and OpenCV to implement the model. |
| 3 | Face Mask Recognition using Machine Learning | IJIRT Paper ID: 150498  Published on : May 2021. | Using TensorFlow, Keras, Pytorch.  3 stages : Images Pre-processing – Detection – Classifier |

**3. METHODS**

**3.1 Dataset Description**

The dataset consists of around 1900+ images including both with and without mask. These are saved as two folders.

**3.2 Data Pre-processing**

Since we use a deep learning model (MobileNet), there isn’t a need to convert the images into gray, instead the images are processed as colour image itself.

**3.3 Exploratory Data Analysis**



MobileNet uses depth wise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. A convolutional layer is a matrix applied to images that perform a mathematical operation on the individual pixels to produce new pixels that are then passed as the input for the next layer and so on until the network's end is reached. The last layer is a single integer that turns the image output into a numerical class prediction that corresponds to an object we are trying to predict. This results in lightweight deep neural networks.

**4. MODEL BUILDING**

The model is built in 2 stages:

1. Detection (Face detection in real time).

2. Recognition (Recognising Mask on that face).

To reduce the stress on the resources, the following parameters are set.

**EPOCH** - It is set to 20. {Total Cycle}

**BATCH SIZE -** Is set to 32. {No of inputs per batch}

**LEARNING RATE -** Is set to 0.0001{for activation Function}

In our proposed model, the Depth Wise Separable Convolution is been used, in which the depth and spatial dimension of a filtered pixel matrix [from the detected portion of the image i.e., face] can be separated.

A depth wise separable convolution is made from two operations.

1. Depth wise convolution.
2. Point wise convolution.

**Depth wise convolution**

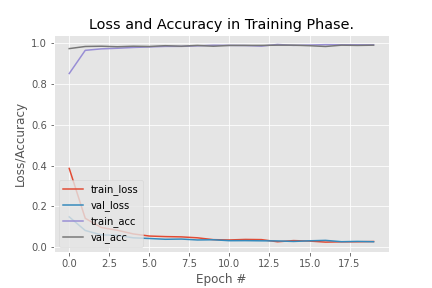
* Depth wise convolution is the channel-wise DK×DK spatial convolution. Suppose in the figure above, and we have five channels; then, we will have 5 DK×DK spatial convolutions.
* It is a map of a single convolution on each input channel separately. Therefore its number of output channels is the same as the number of the input channels. Its computational cost is, **Df² \* M \* Dk².**

**Point wise convolution**

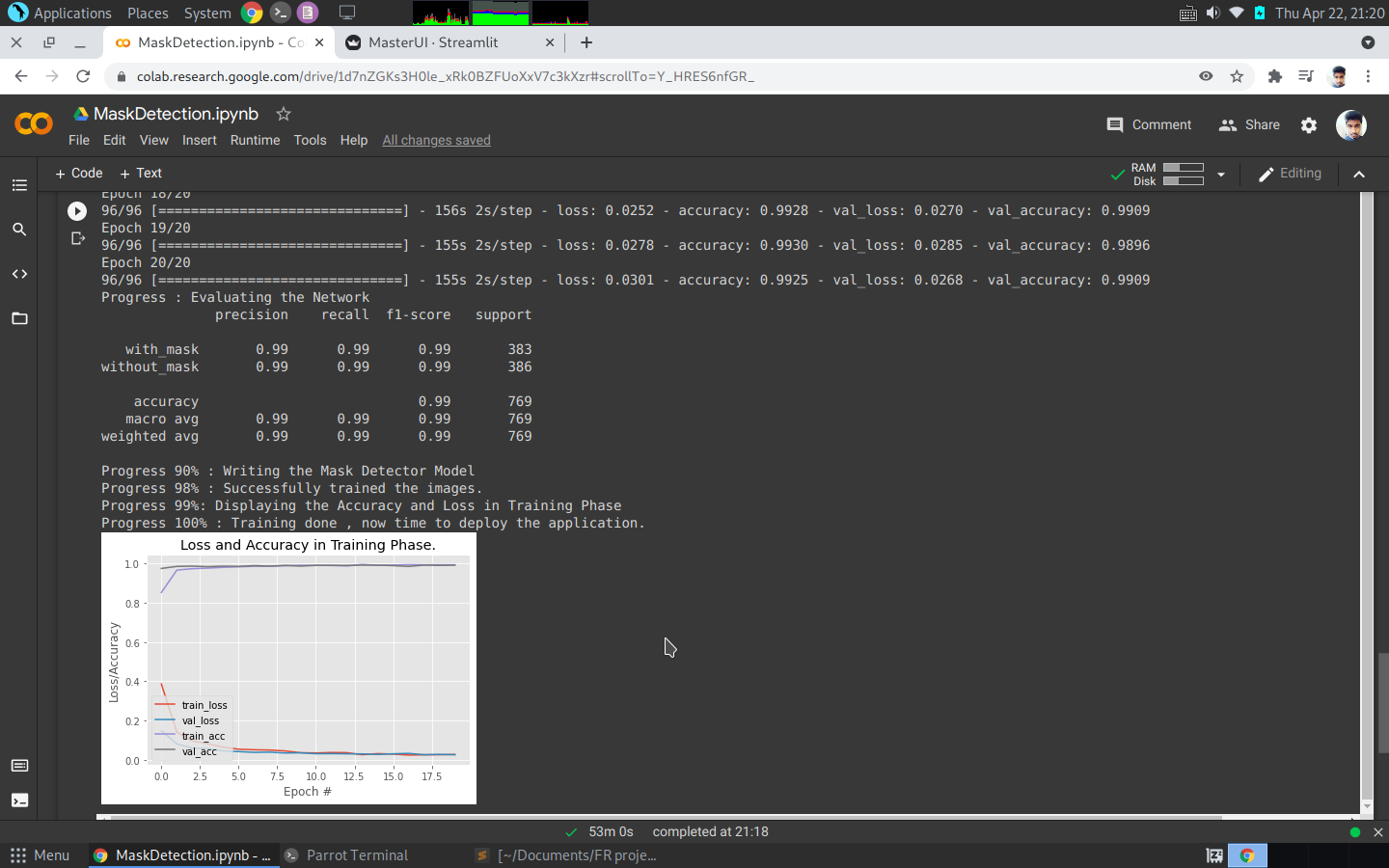
* Point wise convolution is the 1×1 convolution to change the dimension.
* Convolution with a kernel size of 1x1 that simply combines the features created by the depth wise convolution. Its computational cost is, **M \* N \* Df².**

**5. PERFORMANCE AND RESULTS**

Loss/accuracy graph

****

Accuracy



**6. CONCLUSION**

The article proposed an efficient real-time deep learning model to detect the masked faces, where each masked face is identified in real-time with the help of bounding boxes. This Mask detection model gives around 85-90% accuracy, since MobileNet Model is a light weight deep learning model and faster as compared R-CNN, that’s the reason we chose to work on the and this software can also be implemented in mobile phones and other portable devices too.

**7. REFERENCES**

* MobileNet
  + <https://medium.com/analytics-vidhya/image-classification-with-mobilenet-cc6fbb2cd470>
* Article References (S.no)

1. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7775036/>
2. <https://www.codeproject.com/Articles/5293068/AI-on-the-Edge-Face-Mask-Detection>
3. <http://ijirt.org/Article?manuscript=150498>

* Project Link
  + <https://github.com/Nary-Vip/nary_ram_project.git>