



COVID-19 FACE MASK DETECTOR

Rajat Arora

Department of Information Technology, Guru Gobind Singh Indraprastha University,
Sector 16C, Dwarka, Delhi, 110078

Abstract— COVID-19 or popularly known as Corona Virus has changed and disrupted our lives to such an extent that we humans are forced to lock ourselves inside the home and avoid any physical connection with someone else. During this phase we got to know the importance of Face Mask and Hand sanitizers that we usually neglect in our daily lives. We now avoid a person who is not wearing a face mask as it can cause you Covid-19. But sometimes it is very difficult for a human to detect out of many people, who is wearing mask or not. Thus I built a Computer Vision model that detects whether a person is wearing a mask or not using OpenCV and TensorFlow/Keras.

Keywords— Face Mask Detector, Convolutional Neural Network, Deep Learning, Python, OpenCV, TensorFlow, Keras.

I. INTRODUCTION

With the increase in number of Covid-19 infected cases WHO has recommended people to wear face masks to protect themselves and others.

The virus can easily spread from person to person through speaking, coughing, sneezing – even in those who are not having any symptoms. However, Governments and doctors are recommending to wear cloth face coverings or face masks where social-distancing may not be feasible **This model** easily detects with 99% accuracy whether a person is wearing mask or not and outlining his face with green box if he/she is wearing a mask or red box if he/she is not wearing face mask. This model is entirely built using Python, TensorFlow/Keras and OpenCV. This model can be used to automate entry inside a building or premises where wearing mask is compulsory.

A. Application Fundamentals:

This Computer-Vision model is developed entirely on Python. But we are using OpenCV, TensorFlow/Keras for building Convolutional Neural Networks and also using existing Neural Networks. TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. I am using dataset that is developed using OpenCV of images of people

with mask and without mask. This dataset consists of **1,376 images** belonging to two classes: Mask (690 images) and without-mask (686 images). For Face detection we will be fine-tuning the MobileNetV2 model a highly efficient architecture that can be applied to embedded devices with limited computational capacity

II. LITERATURE SURVEY

This project uses MobileNetV2 convolutional neural network that classifies images and was developed by Cornell University. MobileNetV2, improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes. The MobileNetV2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations in the input an MobileNetV2 uses lightweight depth-wise convolutions to filter features in the intermediate expansion layer.

MobileNet-v2 is a convolutional neural network that is 53 layers deep. You can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

We can fine tune it by loading the model and train it on our own dataset.

III. METHODOLOGY

This project implements concepts like Deep Learning, Computer Vision and Convolutional Neural Networks. Model first detects human's face using pre-trained MobileNetV2 network and then runs second convolutional network to classify whether that person is wearing mask or not.

1. Deep Learning:

Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.

2. Computer Vision

Computer vision is an interdisciplinary field related to, e.g., artificial intelligence, machine learning, robotics, signal processing and geometry. The purpose of computer vision is to program a computer to "understand" a scene or features in an image. Computer vision shares many topics and methods with image processing and machine vision.

3. Convolutional Neural Networks:

In deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. They have applications in image and video recognition, recommender systems, image classification, medical image analysis, natural language processing, and financial time series.

IV. TECHNOLOGY

This project is fully developed using Python and it's various libraries (imutils, numpy), and TensorFlow/Keras and OpenCV.

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source BSD license. OpenCV is built using C++. OpenCV is mostly used for computer-vision task and provides us some really helpful features.

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. TensorFlow is developed by Google Brain Team.



Fig. 1. TensorFlow

TensorFlow provides us various methods and pre-built deep learning networks and functions which can be directly used by engineers to develop Machine Learning projects without even caring about what is happening behind it.

V. FUTURE SCOPE

There is always a scope of improvement in every application or industry practice that is carried out in the world however the need is to keep the practice as simple as possible so that even a non-technical user who doesn't have the technical knowledge regarding the field can adopt our practice.

In the future one can easily integrate hardware functionality with this model. For instance, in metro entrance one can use this model to only allow entrance to those who are wearing a mask and thus blocking those who are not wearing the mask.

Also we can retrain our model on more realistic dataset so that it does not under-fits or over-fits. We can also optimize it using various optimization technique.

VI. CONCLUSION

There are few motivation to this project, with increase in cases of Covid-19 across PAN India it was essential to wear face mask but there are times when people avoid to wear this. Since it is now matter of health and safety one can either hire a person to check who all are wearing this mask or use Deep Learning or Artificial Intelligence to solve this problem. This model can easily detect whether a person is wearing a mask or not in real time too, hence no risk to hire someone and no risk of spreading. With the hardware integration this model can be deployed to numerous sites where it is necessary to wear mask and simultaneously there is no one to check whether someone is wearing mask or not. With addition of more images efficiency of model can be increased significantly.

VII. ACKNOWLEDGEMENT

I would like to take the opportunity to acknowledge the support and help of all who have assisted me in the research. Without their contribution and advice, I would have never been able to progress with the work in the research.. The information and feedback provided by friends were extremely helpful and useful for implementing this model.

VIII. REFERENCES

- [1] *TensorFlow OfficialWebsite:* <https://www.tensorflow.org/>
- [2] *Python Website* <https://www.python.org/>.
- [3] *Keras-MobileNetV2Function*
<https://keras.io/api/applications/mobilenet/>
- [4] *Introduced by Sandler et al. in MobileNetV2: Inverted Residuals-and-LinearBottlenecks*
www.paperswithcode.com/method/mobilenetv2/
- [5] *Google's Blog on MobileNetV2 Posted by Mark Sandler and Andrew Howard, Google Research*



<https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html>

[6] *Coursera Machine Learning Course* By Andrew Ng
<https://www.coursera.org/learn/machine-learning>

[7] *Deep Learning Specialization Course* by DeepLearning.ai

<https://www.coursera.org/specializations/deep-learning?>

[8] *Convolutional Neural Networks* by Andrew Ng
<https://www.coursera.org/learn/convolutional-neural-networks?specialization=deep-learning>.

[9] *Convolutional Neural Networks in TensorFlow*
<https://www.coursera.org/learn/convolutional-neural-networks-tensorflow>

[10] *Anaconda Platform* <https://www.anaconda.com/>

[11] Krishna Devulapalli(2015) *Neural networks for classification and regression. Biom Biostat Int J.* 2015;2(6):183-185. DOI: 10.15406/bbij.2015.02.00046

<https://medcraveonline.com/BBIJ/neural-networks-for-classification-and-regression.html#:~:text=Neural%20Networks%20are%20well%20known%20techniques%20for%20classification,the%20results%20obtained%20from%20these%20models%20were%20studied.>

[12] *Article on CNN Implementation* by Sumit Saha (2018)
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

[13] *OpenCV Official Website* <https://opencv.org/>

[14] *OpenCV-tutorial*
https://opencv.org/courses/course_computer_vision_one.html