Face Mask Detection System Using Deep Learning

[1]Mrs.Supriya Kurlekar, [2]Mr. Aniket A. Omanna., [3]Mr. Onkar A. Deshpande, [4]Mr. Dinesh B. Patil.

[1]SITCOE(Yadrav)India

[2]SITCOE(Yadrav)India

[3]SITCOE(Yadrav)India

[4]SITCOE(Yadrav)India

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 16 April 2021

Abstract: In the present scenario due to Covid-19, there are no efficient face mask detection applications which are now in high demand for transportation means, densely populated areas, residential districts, large-scale manufacturers and other enterprises to ensure safety. This system can therefore be used in real-time applications which require face-mask detection for safety purposes due to the outbreak of Covid-19. This project can be integrated with embedded systems for application in airports, railway stations, offices, schools, and public places to ensure that public safety guidelines are followed.

Keywords: Face Mask Detection, Deep Learning, Tensorflow, Keras, Pi Camera, Raspberry Pi 4, Python Programming

1. INTRODUCTION

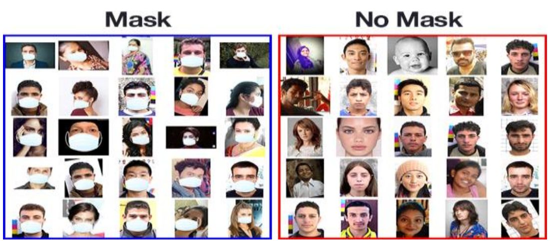
The world is struggling with Covid-19 pandemic and are so many essential equipments needed to combat against Corona virus. One of such most essential is Face Mask and mask was not mandatory for everyone but as the day surpasses scientist and Doctors have recommended everyone to wear the mask. Therefore to detect whether a person is wearing Face Mask or not, there are detection technique. Face Mask Detection Platform utilizes Artificial Network to identify if a person does/doesn't wear a mask. The application can be associated with any current or new IP cameras to identify individuals with/without a mask.

2. WORKING PRINCIPLE

Our Face Mask Detection system built with OpenCV, Keras/TensorFlow using Deep Learning and Computer Vision concepts. We can detect face masks in Static Images as well as in Real-time Video streams.

A data set is a collection of data. In Deep Learning projects, we need a training data set. It is the actual data set used to train the model for performing various actions. A collection of instances is a dataset and when working with machine learning methods we typically need two datasets for different purposes.

- **Training Dataset**: A dataset that we feed into our Deep learning algorithm to train our model.
- **Testing Dataset**: A dataset that we use to validate the accuracy of our model but is not used to train the model. It may be called the validation dataset.



Our dataset consists of 3835 images belonging to two classes:

- with_mask: 1916 images
- without_mask: 1919 images

How was our face mask dataset created?

To create this dataset, we follow these two steps:

- Taking normal images of faces

- Then creating a custom computer vision Python script to add face masks to them, thereby creating an artificial (but still real-world applicable) dataset.

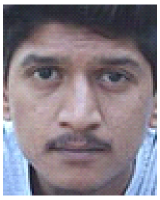
Facial landmarks allow us to automatically infer the location of facial structures, including

- Eyes
- Eyebrows
- Nose
- Mouth
- Jaw line

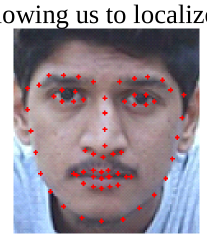
To use facial landmarks to build a dataset of faces wearing face masks, we need to first start with an image of a person not wearing a face mask:

From there, we apply face detection to compute the bounding box location of the face in the image:

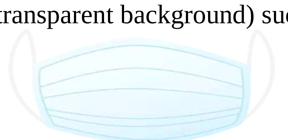
Once we know where in the image the face is, we can extract the face Region of Interest (ROI):



And from there, we apply facial landmarks, allowing us to localize the eyes, nose, mouth, etc.:



Next, we need an image of a mask (with a transparent background) such as the one below:



This mask will be automatically applied to the face by using the facial landmarks (namely the points along the chin and nose) to compute where the mask will be placed.

The mask is then resized and rotated, placing it on the face:

We can then repeat this process for all of our input images, thereby creating our artificial face mask dataset:

3. BLOCK DIAGRAM

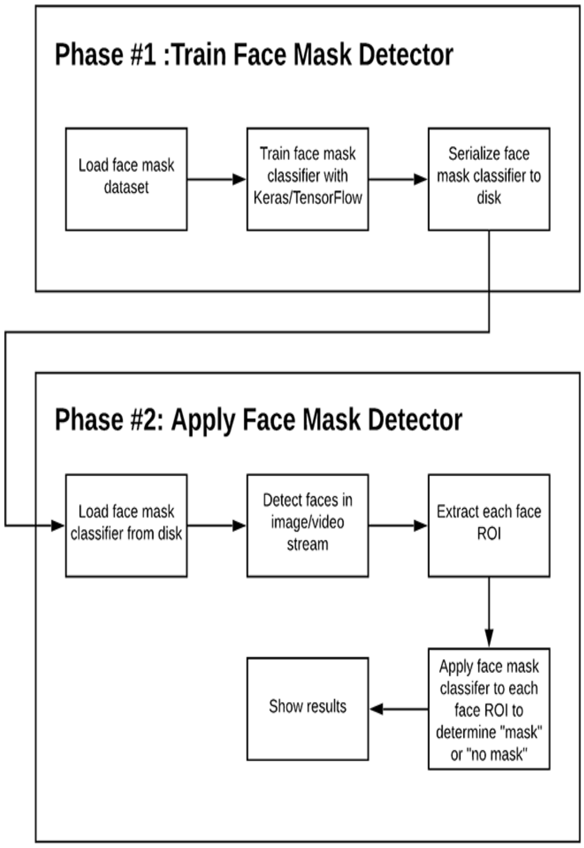


Figure.1 Block Diagram

In order to train a custom face mask detector, we need to break our project into two distinct phases, which is,

- **Training**: Here we focus on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk

- **Deployment**: Once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with mask of without mask.

4. HARDWARE IMPLEMENTATION

RaspberryPi

Raspberry Pi is a low cost, credit card sized computer that connects to monitor and uses standard keyboard and mouse. The hardware components of the Raspberry Pi include power supply, storage, input, monitor and network.

- CPU: Broadcom BCM2836 900MHz
- quad-core ARM Cortex-A7 processor
- RAM: 1 GB SDRAM

- USB Ports: 4 USB 2.0 ports
- Network: 10/100 Mbit/s Ethernet
- Power Ratings: 600 mA (3.0 W)
- Power Source: 5V Micro USB
- Size: 85.60 mm × 56.5 mm
- Weight: 45 g (same as Raspberry Pi B+)
- 802.11n Wireless LAN
- 40 GPIO pins
- Full HDMI port
- Combined 3.5mm audio jack and composite video
- Camera interface (CSI)
- Display Interface (DSI)
- Micro SD card slot

Piamera

The Raspberry Pi camera module can be used to take high-definition video, as well as stills photographs. The camera module is very popular in home security applications, and in wildlife camera traps.

- 5MP sensor
- Wider image, capable of 2592x1944 stills, 1080p30 video
- 1080p video supported
- CSI
- Size: 25 x 20 x 9 mm

5. SOFTWARE IMPLEMENTATION

5.1 FRAMEWORKS

OpenCV

- OpenCV means Open Source Computer Vision Library
- It is a library of programming functions mainly aimed at real-time computer vision.
- This library originally developed by Intel.
- OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. All of the new developments and algorithms appear in the C++ interface.
- There are bindings in Python, Java and MATLAB.

Caffe

- Caffe is a deep learning framework made with expression, speed, and modularity in mind. It is developed by Berkeley AI Research (BAIR) and by community contributors.
- We use caffe deep learning framework for face detection.

Why caffe?

- Expressive architecture encourages application and innovation. Models and optimization are defined by configuration without hard-coding.
- Speed makes Caffe perfect for research experiments and industry deployment. Caffe can process over 60M images per day with a single NVIDIA K40 GPU*.

TensorFlow

- TensorFlow is an end-to-end open source platform for machine learning.
- It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.
- **Easy model building**
  Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.
- **Robust ML production anywhere**
  Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.
- **Powerful experimentation for research**

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

Keras

- Keras is the most used deep learning framework. Because Keras makes it easier to run new experiments.
- Take advantage of the full deployment capabilities of the TensorFlow platform.
- Keras is used by CERN, NASA, NIH, and many more scientific organizations around the world
- Keras has the low-level flexibility to implement arbitrary research ideas while offering optional high-level convenience features to speed up experimentation cycles.

6. FLOW OF PROCESS

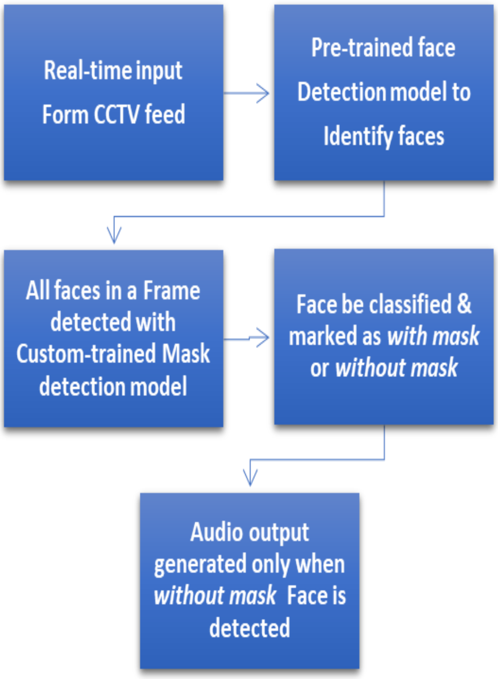
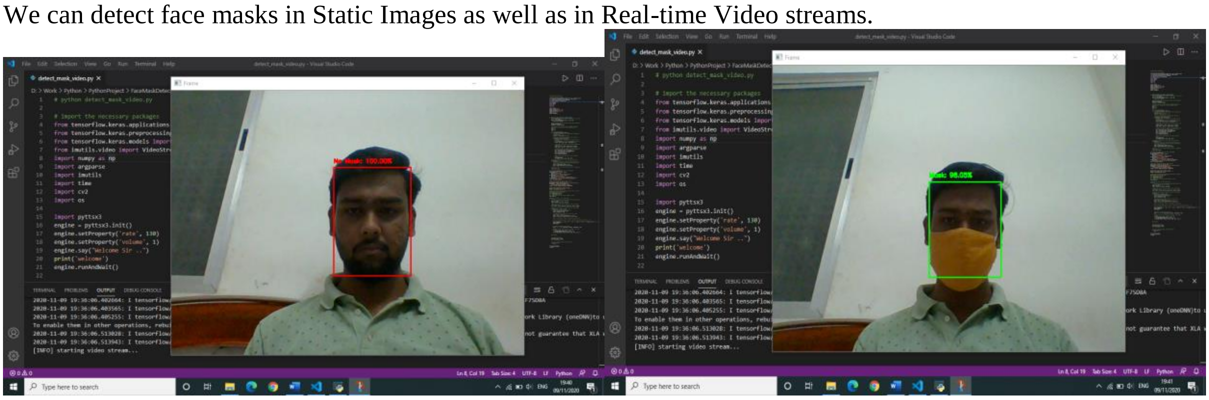


Figure.2 Flow of Process

7. RESULTS

We can detect face masks in Static Images as well as in Real-time Video streams.



8. APPLICATIONS

The system can be used in the following places to identify people with or without masks:

Airports

The Face Mask Detection System can be used at airports to detect travelers without masks. Face data of travelers can be captured in the system at the entrance. If a traveler is found to be without a face mask, their picture is sent to the airport authorities so that they could take quick action.

Hospitals

Using Face Mask Detection System, Hospitals can monitor if their staff is wearing masks during their shift or not. If any health worker is found without a mask, they will receive a notification with a reminder to wear a mask. Also, if quarantine people who are required to wear a mask, the system can keep an eye and detect if the mask is present or not and send notification automatically or report to the authorities.

Offices

The Face Mask Detection System can be used at office premises to detect if employees are maintaining safety standards at work. It monitors employees without masks and sends them a reminder to wear a mask. The reports can be downloaded or sent an email at the end of the day to capture people who are not complying with the regulations or the requirements.

Government

To limit the spread of coronavirus, the police could deploy the face mask detector on its fleet of surveillance cameras to enforce the compulsory wearing of face masks in public places. Analyzing the current scenario, government and private organizations want to make sure that everyone working or visiting a public or private place is wearing masks throughout the day. Our face mask detection system can quickly identify the person with a mask, using cameras and analytics.

REFERENCES

1. Z.Wang, G.Wang, B.Huang, Z.Xiong, Q.Hong, H.Wu, P. Yi, K. Jiang, N.Wang, Y.Pei, H.Chen, Y.Miao, Z.Huang, And J.Liang. 2020. Masked Face Recognition Dataset And Application
2. Agrawal, Prateek, S. R. P. Sinha, And S. U. B. O. D. H. Wairya. "Quantum Dot Cellular Automata Based Parity Generator And Detector: A Review." *International Journal Of Electronics And Communication Engineering* 5.3 (2016).
3. J.Redmon, S.Divvala, R.Girshick, A.Farhadi. You Only Look Once: Unified, Real-Time Object Detection
4. Arefin, Syed Samsul, I. B. R. A. H. I. M. Azad, And H. U. M. A. Y. U. N. Kabir. "Modified Sack-Tcp And Some Application Level Techniques To Support Real-Time Application." *International Journal Of Electrical And Computer Engineering (Ijece)* 6 (2013): 105-114.
5. J.Huang, V.Rathod, C.Sun, M.Zhu, A.Korattikara, A.Fathi, I.Fischer Z.Wojna, Y.Song, S.Guadarrama, K.Murphy. 2017. Speed/Accuracy Trade-Offs For Modern Convolutional Object Detectors.
6. Ugla, A., Dhuha J. Kamil, And Hassan J Khaudair. "Interpretable Recommender System With Heterogeneous Information: A Geometric Deep Learning Perspective." *International Journal Of Mechanical And Production Engineering Research And Development (Ijmperd)* 10.3 (2020): 2411-2430
7. Kumar, B. Satish, And Y. Kalyan Chakravarthy. "Prediction Of Optimal Torques From Gait Analysis Applying The Machine Learning Concepts." *International Journal Of Mechanical And Production Engineering Research And Development (Ijmperd) 9. 4, Aug 2019, 685* 698.
8. El Hoummaidi, L. A. L. A., And Abdelkader Larabi. "Dubai Smart Map, A New Tool Enabling Geo-Digital Government, Driven By Innovative Geospatial Thinking For A Smart And Sustainable Dubai." International Journal Of Civil, Structural, Environmental And Infrastructure Engineering Research And Development (Ijcseierd)9.6, Dec 2019, 17–36
9. Danthala, S. W. E. T. H. A., Et Al. "Robotic Manipulator Control By Using Machine Learning Algorithms: A Review." *Int J Mech Prod Eng Res Dev* 8.5 (2018): 305-310.