International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A SOLUTION TO COVID-19 PANDEMIC: FACE MASK DETECTION FROM IMAGE-VIDEO USING DEEP LEARNING CNN

Mr. Joshi Mayuresh D[1], Prof. Dhakane V. N.[2]

PG Student[1], S.N.D.COE and RC, Yeola
Asst. Professor[2], S.N.D.COE and RC, Yeola
Email ID: mayureshj40@gmail. vikasdhakane@gmail.com

ABSTRACT

This project intends to develop a Face Mask Detection system using OpenCV, Keras/Tensor Flow and Deep Learning. The system can be easily integrated to various embedded devices with limited computational capacity as it uses MobileNetV2 architecture. It will detect face masks in images as well as in real- time videos.

In recent times, where Covid-19 has impacted a domino effect on manufacturing, travel, tourism, hospitality, crippling the global economy. In addition to it, is the growing curve of human deaths across the globe due to the pandemic, this project which relies on computer vision and deep learning, intends to make an impact and solve the real-world problem of safety measures at some significant level.

This project can be used at airports, offices, hospitals and many more public places to ensure that the safety standards are maintained and people are abiding by the rules and regulations to wear protective masks at public places. If the detection system classifies as 'No Mask', reminders can be given as well as actions can be taken against such individuals.

***Keywords**: Corona virus, Covid-19, Machine Learning, Face Mask Detection, Convolutional Neural Network, Tensor Flow, Deep Learning.*

1. INTRODUCTION

In the present scenario due to Covid-19, there is 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 that the safety guidelines are strictly followed. Also, the absence of large datasets of 'with mask' images has made this task more cumbersome and challenging.

The model proposed here is designed and modeled using python libraries namely Tensor flow, Keras and OpenCV. The model we used is the MobileNetV2 of Convolutional neural network. The method of using MobileNetV2 is called using Transfer Learning.

Transfer learning is using some pre trained Model to train your present model and get the prediction which saves time and makes using training the different models easy. We tune the model with the hyper parameters: learning rate, number of epochs and batch size. The model is trained with a dataset of images with two class, with mask and without mask. The dataset has 993 images of with mask class and 1918 images of without mask class.

2. REVIEW OF LITERATURE

Arjya Das[1], Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV, this article is free to access and download, along with rights for full text and data mining, re-use and analysis Detection Using Tensor Flow, Keras and OpenCV, 2020. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like

Ms. R. Suganthalakshmi, A. Hafeeza, P. Abinaya, A.Ganga Devi[2] Covid-19 Facemask Detection with Deep Learning and Computer Vision International Journal of Engineering Research & Technology (IJERT),2021. Here we introduce a facemask detection model that is based on computer vision and deep learning. The proposed model can be integrated with Surveillance Cameras to impede the COVID-19 transmission by allowing the detection of people who are wearing masks not wearing face masks.

Mohammad Marufur Rahman [3], an Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network, IEEE 2020. 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 smart city network where all the public places are monitored with Closed-Circuit Television (CCTV) cameras. While a person without a mask is detected, the corresponding authority is informed through the city netw Riya Chirag Kumar Shah, Detection of Face Mask using Convolutional Neural Network, Riya Chirag Kumar Shah[4], Detection of Face Mask using Convolutional Neural Network, In this tutorial, you will learn how to perform fine-tuning with Keras and Deep Learning, We have used convolutional neural network for the same. The model is trained on a real-world dataset and tested with live video streaming with a good accuracy. Further the accuracy of the model with different hyper parameters and multiple people at different distance and location of the frame is done.

3. SYSTEM ARCHITECTURE

The major requirement for implementing thisproject using python programming language along with Deep learning Machine learning, Computer vision and also with python libraries. The architecture consists of Mobile Net as backbone, it can be used for high and low computation scenarios. We are usingCNN Algorithm in our proposed system.

We have four modules

1. **Datasets Collecting**: We collect no of data sets with face mask and without masks. We can get high accuracy Depends on collecting the number of images.

2. **Datasets Extracting:** We can extract the features using mobile net v2 of mask and no mask sets

3. **Models Training:** We will train the model using open CV, Keras (python library).

4. **Facemask Detection:** We can detect Preprocessing image and also detect via live video. If people wear mask, it will permit them, if not then it will give the buzzer to wear mask to prevent them from virus transmission.

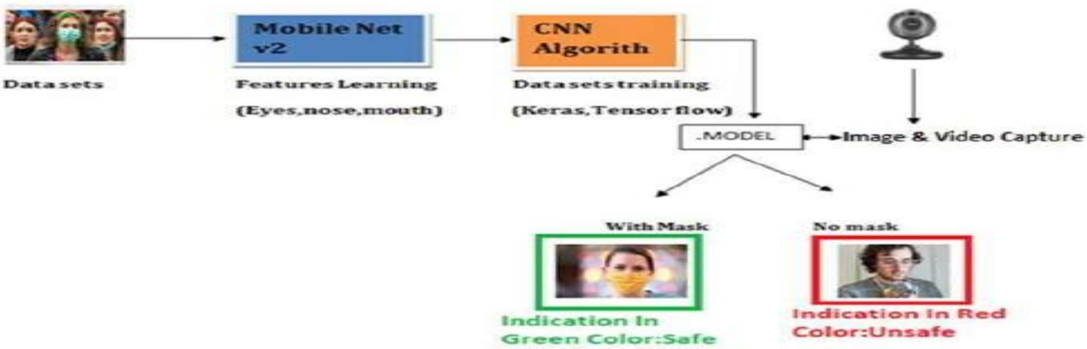


Figure 1: System Overview

DATA COLLECTION

The data will be collected from the below mentionedsources:

- Real-World Masked Face Dataset (RMFD)
- Kaggle Datasets
- Bing Search API

3.1 Fundamental steps to perform for face maskdetection

Phases and individual steps for building a COVID-19 face mask detector with computer vision and deep learning using Python, OpenCV, and TensorFlow/Keras.

i) **Training:**

a) Loading the face mask detection dataset from disk b
b) .Training a model using Keras/TensorFlow on this loaded Dataset
c) Serializing the face mask detector back to disk

ii) Deployment:

a) Loading the face mask detector

b) Performing face detection

c) Classifying each face as “Mask” or “No Mask”

The aim is to collect more than 1800+ images in both “with mask” and “without mask” classes. A Python script will be used for Bing Search API for finding images with multiple queries like “covid mask”, “face mask”, “N95 mask”.

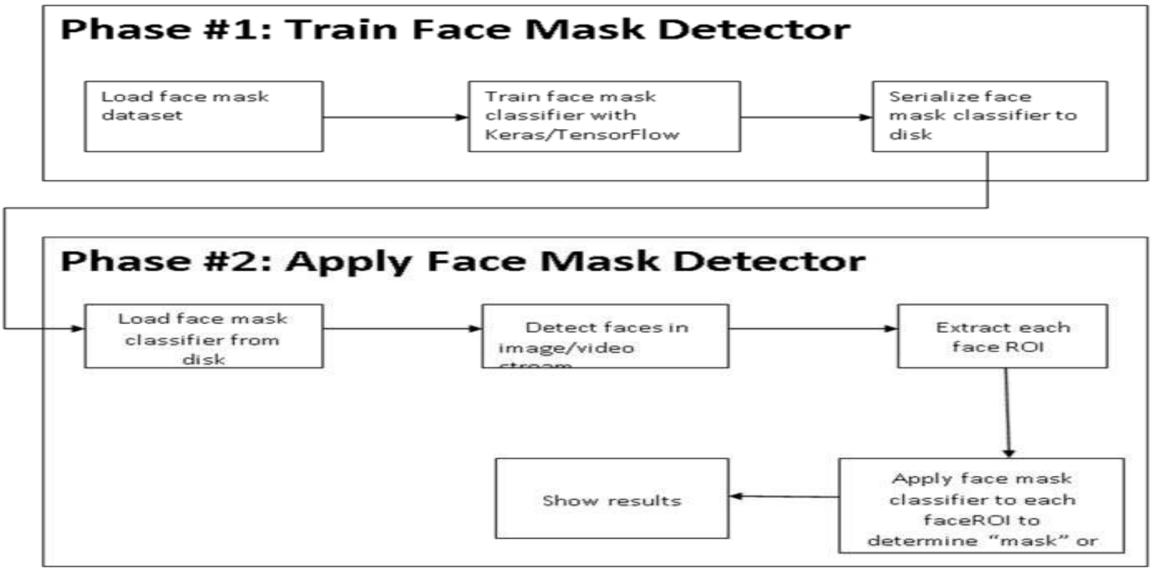


Figure 2: Fundamental steps to perform for face mask detection

Training of Model

a) **Building the model using** CNN architecture: CNN has become ascendant in miscellaneous computer vision tasks. The current method makes use of Sequential CNN. The First Convolution layer is followed by Rectified Linear Unit (ReLU) and Max Pooling layers. The Convolution layer learns from 200 filters. Kernel size is set to 3 x 3 which specifies the height and width of the 2D convolution window. Asthe model should be aware of the shape of the input expected, the first layer in the model needs to be provided with information about input shape.

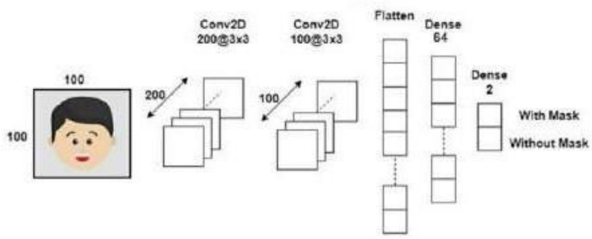


Figure 3 Convolutional Neural Network architecture

b) Splitting the data and training the CNN model:

After setting the blueprint to analyze the data, the model needs to be trained using a specific dataset and then to be tested against a different dataset. A proper model and optimized train test split help to produce accurate results while making a prediction. The test size is set to 0.1 i.e. 90% data of the dataset undergoes training and the rest 10% goes for testing purposes. The validation loss is monitored using Model Checkpoint. Next, the images in the training set and the test set are fitted to the Sequential model. Here, 20% of the training data is used as validation data. Themodel is trained for 20 epochs (iterations) which maintains a trade-off between accuracy and chances ofover-fitting.

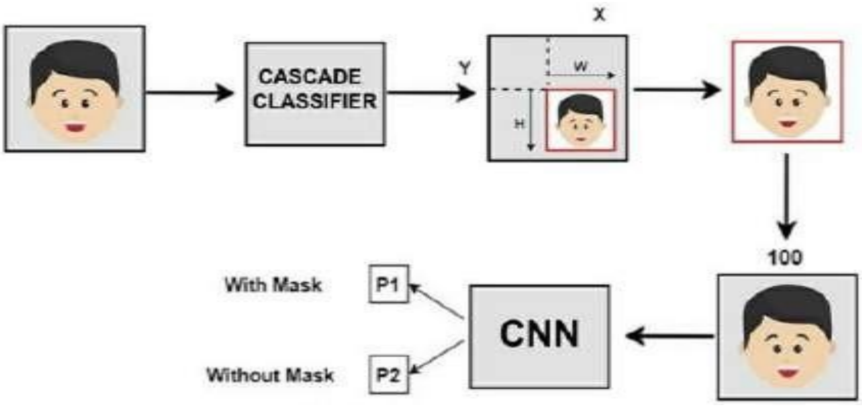


Figure 4 Overview of the Model

4. METHODOLOGY & ALGORITHM

A. OpenCV's "deep neural networks" (dnn) module:

OpenCV (3.3 or later) comprises of the highly efficient dnn module supported by a number of deep learning frameworks such as Caffe, TensorFlow, and Torch/PyTorch. This module has a more accurate Caffe-based face detector. In this project, we will be training our deep learning model using Caffe, hencewe need the following files:

- The. prototxt file(s) which will define the model architecture (i.e., the layers)
- The. caffemodel file which will contain theweights for the actual layers
- **OpenCV's face detector** based on the Single Shot Multibox Detector (SSD) Framework combined with MobileNetV2 architecture: In order to obtain the bounding box (x, y) coordinates for an object (mask in this case) in an image we need to apply object detection.
- **Kera's Image Data Generator: Generate batches of tensor image data with real-time data augmentation. The data will be looped over (in batches).**
- **Keras Imag eData Genera tor class worksby:**
- Accepting a batch of images used fortraining.
- Taking this batch and applying a series of random transformations to each image in the batch (including random rotation, resizing, shearing, etc.).
- Replacing the original batch with the new, randomly transformed batch.
- Training the CNN on this randomly transformed batch (i.e., the original data itself is not used for training).

4.1 Algorithmic Strategies

A. Feature Extraction using ConvNets

Traditional machine learning approach uses feature extraction for images using Global feature descriptors such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HoG), Color Histograms etc. or Local descriptors such as SIFT, SURF, ORB etc. These are hand-crafted features that require domain level expertise.

But here comes Convolutional Neural Networks (CNN). Instead of using hand-crafted features, Deep Neural Nets automatically learns these features fromimages in a hierarchical fashion.

Lower layers learn low-level features such as Corners, Edges whereas middle layers learn color, shape etc. and higher layers learn high-level features representing the object in the image.

B. Transfer Learning Algorithm

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is the idea of overcoming the isolated learning paradigm and utilizing the knowledge acquired for one task to solve related ones.Traditional learning is isolated and occurs purely based on specific tasks, datasets and training separate isolated models on them. No knowledge is retained which can be transferred from one model to another. In transfer learning, you can leverage knowledge (features, weights etc.) from previously trained models for training newer models and even tackle problems like having less data for the newer task! Learning is not an easy process, not for humans and not for machines either. It is a heavy-duty, resource-consuming and time-consuming process and hence it was important to devise a method that would prevent a model from forgetting the learning curve that it attained from a specific dataset and alsolets it learn more from new and different datasets.

C. Transfer Learning with a pre-trainedConvNet

We can have two ways to customize a pre-trainedmodel:

- Feature Extraction: Use the representations learned by a previous network to extract meaningful features from new samples. We simply add a new classifier, which will be trained from scratch, on top of the pre- trained model so that we can repurpose the feature maps learned previously for the dataset.

- We do not need to (re)train the entire model. The base convolutional network already contains features that are generically useful for classifying pictures. However, the final, classification part of the pre-trained model is specific to the original classification task, and subsequently specific to the set of classes on which the model was trained.

4.2 Incorporated packages

a) TensorFlow

TensorFlow, an interface for expressing machine learning algorithms, is utilized for implementing ML systems into fabrication over a bunch of areas of computer science, including sentiment analysis, voice recognition, geographic information extraction, computer vision, text summarization, information retrieval, computational drug discovery and flaw detection to pursue research [18]. In the proposed model, the whole Sequential CNN architecture (consists of several layers) uses TensorFlow at backend. It is also used to reshape the data (image) in the data processing.

b) Kera's

Keras gives fundamental reflections and building units for creation and transportation of ML arrangements with high iteration velocity. It takes full advantage of the scalability and cross-platform capabilities of TensorFlow. The core data structures of Keras are layers and models [19]. All the layers used in the CNN model are implemented using Keras. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

c) OpenCV

OpenCV (Open Source Computer Vision Library), an open source computer vision and ML software library, is utilized to differentiate and recognize faces, recognize objects, group movementsin recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparativepictures from an image database, perceive landscapeand set up markers to overlay it with increased reality and so forth [20]. The proposed method makes use of these features of OpenCV in resizing and color conversion of data images.

5. DATASET

Two datasets have been used for experimenting the current method. Dataset 1, consists of 1376 images in which 690 images with people wearing face masks and the rest 686 images with people who do not wear face masks. Fig. 1 mostly contains front face pose with single face in the frame and with same type of mask having white color only. Dataset 2 from Kaggle consists of 853 images and its countenances are clarified either with a mask or without a mask. In fig. 2 some face collections are head turn, tilt and slant with multiple faces in the frame and different types of masks having different colours as well.

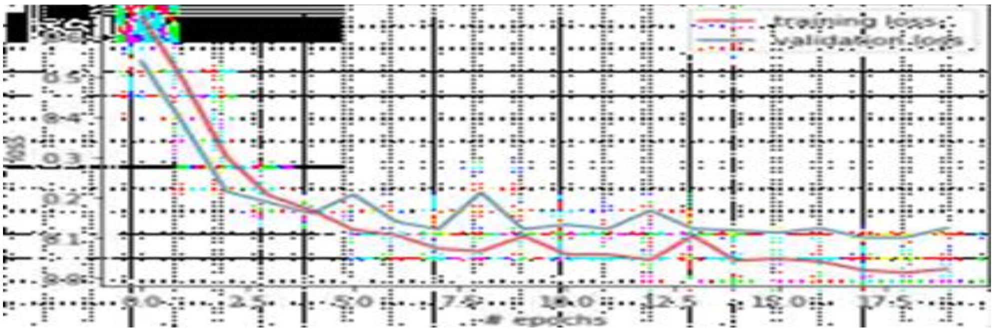
Dataset 2 from Kaggle consists of 853 imagesand its countenances are clarified either with a mask or without a mask. In fig. 2 some face collections are head turn, tilt and slant with multiple faces in the frame and different types of masks having different colours as well.

Samples from Dataset 2 including faces without masks and withmasks
Samples from Dataset 2 including faces without masks and withmasks

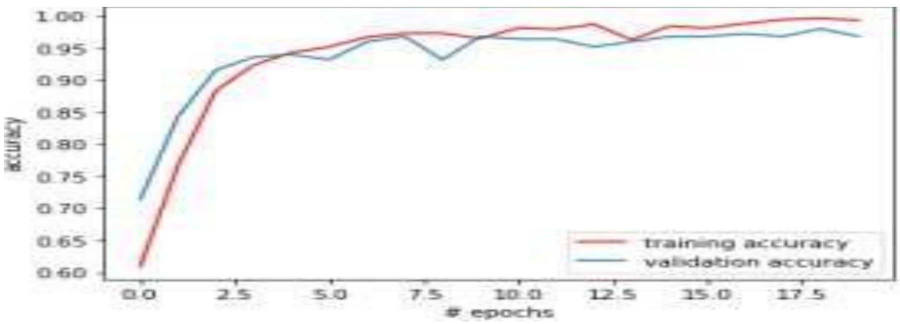
Samples from Dataset 2 including faces without masks and withmasks

6. RESULT AND ANALYSIS

The model is trained, validated and tested upon two datasets. Corresponding to dataset 1, the method attains accuracy up to 95.77% (shown in fig. 7). Fig. 6 depicts how this optimized accuracy mitigates the cost of error. Dataset 2 is more versatile than dataset 1 as it has multiple faces in the frame and different types of masks having different colors as well. Therefore, the model attains an accuracy of 94.58% on dataset 2 as shown in Fig.19,Fig. 8 depicts the contrast between training and validation loss corresponding to dataset 2. One of the main reasons behind achieving this accuracy lies in *Max Pooling*. It provides rudimentary translation in-variance to the internal representation along with the reduction in the number of parameters the model has to learn. This sample- based discretization process down-samples the inputrepresentation consisting of image, by reducing its dimensionality. Number of neurons has the optimized value of 64 which is not too high. A much higher number of neurons and filters can lead to worse performance. The optimized filter values and pool size help to filter out the main portion (face) of the image to detect the existence of mask correctly without causing over-fitting.



epochs vs loss corresponding to dataset 1



The system can efficiently detect partially occluded faces either with a mask or hair or hand. It considers the occlusion degree of four regions – nose, mouth, chin and eye to differentiate between annotated mask or face covered by hand. Therefore, a mask covering the face fully including nose and chin will only be treated as "with mask" by the model.

7. BENEFITS

- Manual Monitoring is very difficult for officers to check whether the peoples are wearing mask or not. So, in our technique, we are using web cam to detect people's faces and to prevent from virus transmission.
- It has fast and high accuracy
- This system can be implemented in ATMs, Banks etc.
- We can keep peoples safe from our technique.
- It provides buzzer sound to wear mask.

8. CONCLUSION & FUTURE SCOPE

An accurate and efficient face mask detection system has been developed which achieves comparable metrics with the existing state-of-the-art system. This project uses recent techniques in the field of computer vision and deep learning. Custom dataset was made from scratch using Bing Search API, Kaggle datasets and RMFD dataset, and the evaluation of the model on test dataset was found consistent. The system correctly detected the presence of face masks on human faces that it detected in static images as well as real-time video streams.

To create our face mask detector, we trained a two-class model with images of people wearing masks and not wearing masks. We then fine-tuned our model using MobileNetV2 on our mask/no mask dataset and obtained an image classifier that was 93% accurate.

We then took this face mask classifier and applied it to both images and real-time video streams by:

1. Detecting faces in the images/video
2. Extracting each individual face ROI
3. Applying our face mask classifier

Our face mask detector is accurate, and since we used the MobileNetV2 architecture, it's also computationally efficient and thus making it easier to deploy the model to embedded systems (Raspberry Pi, Google Coral, etc.).

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.

REFERENCES

[1] Arjya Das, Covid-19 Face Mas IEEE 2021. This article is free to access and download, along with rights for full text and data mining, re-use andanalysis Detection Using Tensor Flow, Keras and OpenCV, 2020.

[2] Ms. R. Suganthalakshmi, A. Hafeeza, P. Abinaya, A.Ganga Devi Covid-19 Facemask Detection with Deep Learning and Computer Vision International Journal of Engineering Research & Technology\ (IJERT), 2021.

[3] Mohammad Marufur Rahman, an Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network, IEEE 2020.

[4] https://keras.io/api/applications/mobilenet/#mobilenetv 2-function

[5] https://arxiv.org/abs/1704.04861

[6] https://www.pyimagesearch.com/2019/06/03/fine- tuning-with-keras-and-deep-learning/

[7] https://github.com/X zhangyang/Real-World-Masked-Face-Dataset

[8] https://github.com/AIZOOTech/FaceMaskDetection

[9] https://www.pyimagesearch.com/2018/04/09/how- to-quckly-build-a-deep-learning-imagedataset/

[10] https://gogul.dev/software/flower-recognition- deep-learning

[11] https://www.tensorflow.org/tutorials/images/transf er_learning

[12] https://towardsdatascience.com/detecting-faces-with-python-and-opencv-facedetectionneural-Network-f72890ae531c

[13] https://www.indulgexpress.com/msociety/2020/may/04/Hyderabad-companylaunches-aibased-surveillance-tech-to-detect-face-mask-and-social- distancing-viol-24656.html

[14] https://www.engadget.com/uber-face-mask- detection-technology-184137514.html

[15] https://www.leewayhertz.com/face-mask- detection-system/

[16] https://www.thehansindia.com/telangana/hyderaba d-cameras-to-detect-face-masks-testtemp soon

[17] https://www.kaggle.com/chandrikadeb/face-mask- dataset

[18] https://www.pyimagesearch.com/2019/07/08/keras -imagedatagenerator-and-dataaugmentation/