GowriSankar Penubothu, Smriti Banwari, Suvidha Yadav

[[1]](#footnote-1)

Face Mask Detection

(2022)

*Abstract*— **People are experiencing a health crisis due to the COVID-19 pandemic. One effective way to combat the virus is to wear a face mask. This document provides information on face mask detection that authorities can use to mitigate, evaluate, prevent, and plan actions against COVID19. The face mask recognition in this study was developed with a machine learning algorithm using Image**   **classification model based on MobileNetV2 architecture and OpenCv face detector is used. So, you can use these steps to determine where a person is and whether or not he is wearing a face mask. The FaceNet model is used as a feature extractor and feedforward multilayer perceptron for face recognition. Model building phase includes data collection, preprocessing, data segmentation , model testing, and model implementation. The generated model can detect a masked person other than the wearing a mask with 99% accuracy.**

# INTRODUCTION

**S**ince the spread of the COVID-19 (corona) virus, there were many precautions and efforts were made to control the spread of the Corona virus. Corona virus is transmitted through the respiratory system and direct interaction occurs by close contact with oral secretions via a physical transfer of the microorganism. Explicit interaction happens while a person carrying the virus is sneezing or coughing that disperses droplets of the virus to surfaces. Many countries made it mandatory to wear face masks and many laws have been passed stating that who violates the mandatory rules will be under certain punishments. In some places laws like charging some penalty make to do physical punishment in public place. And is necessary that every responsible citizen should wear mask to keep him safe from virus and the society.

Since government and WHO’s a key preventive measure is to wear facemask when going outside as well as social distancing. Therefore, it has become essential to create automated programs to find out whether anyone is wearing a mask or not so measures can be taken accordingly. But authorities face many difficulties in the process of supervising the corona situation. Authorities need an optimized solution to vividly control the whole situation of wearing masks, following the covid protocols accurately.

## About Problem

One of the solutions is to use a face mask detection software which differentiates people who are with and without mask. This model allows face mask detection that can be used by the organizations or governments to easily monitor, control the spread of the corona virus by having this model implemented. In other ways industries which make face masks can know the habit of the people wearing the masks and can make certain models so that people get used to wearing face masks.

## Motivation

As in this time of pandemic many lives are passing away for not just carrying the face masks on. So, when we came to know about a model something like this immediately without second mind decided to work on this model which may or may not help the society and helps us to get experience on the many machine learning models like CNN, MobileNetV2 etc.

## Background Knowledge

As the students perusing AIML had certain knowledge on the basic classification models how they work and had an experience on the image pre-processing in python and we will do the research and learn the CNN and MobileNetV2 that we are planning to work on.

# Related Work

There are several models proposed about face mask detection. Face mask detection is a subset of object recognition that uses image processing algorithms. Digital image processing may be divided into two broad categories: classical image processing and deep learning- based image analysis. As opposed to classical image analysis, which uses complex formulas to recognize and interpret pictures, deep learning-based approaches utilize models that mimic the workings of the human brain. Deep Learning models have been used in most of the past research. After correctly recognizing the face in the picture or video, the **CNN**-based approach by Kaur et at [1] evaluates if the face has been disguised.

As just a monitoring job performance, this could also make the distinction of a movable face as well as a mask in a video. The above technique has a high degree of precision. Bhuiyan et al. [2] designed the **YOLO-v3** algorithm for detecting face mask in public venues. They had their own generated dataset of photographs with individuals marked as "mask and no -mask" to train the YOLO-v3 model. Mata [3] exploited data augmentation to strengthen the model's performance. To retrieve the facial area as a **ROI**, a CNN model which can differentiate amongst ROIs with and without a face mask must be designed.

Toppo et a1. [4] uses **MobileNetV2** to create a technique for detecting face masks which thus comprises three separate face detector models to check and assess the model's reliability and functionality. The results of the trained model will help for low energy consumption execution, enabling the mask detection method's participation quicker than prior technique.

Balaji et a1. [5] used a VGG-16 CNN model implemented in **Keras**/**TensorFlow** and **Open**-**CV** to locate people who aren't really wearing face masks in government buildings to monitor people who were not wearing face masks. Fan et a1. [6] suggested two new alternative measures to solve for the model's small weight. For sensitive face mask sections, there seems to be a one-of-a-kind persistent situational awareness module. To obtain enhanced mask differentiation features, a two-stage synthetic Gaussian heat map regression analysis was performed. Thus, according to ablation research, such measures can boost feature extraction and, as a response, numerical identification success. The recommended model outperformed existing models for AIZOO and Moxa3K.

Conventional deep learning approaches for lighter weight facial image classification alone will not produce a decent distinguishing feature set, as indicated by the research mentioned above, and they also complicate the model by increasing the number of factors and processing requirements needed.

In this study, a Depth wise Separable Convolution Neural Network-based MobileNet is built for the identification of face masks by identifying facial images to face the in adequacies of preceding research on this topic.

By switching traditional convolution in the neural network with such a context separable convolution, our technique is used to improve the effort delivered. A tabular summary of past projects is included in the table below.

Table1. A tabular summary of past projects.

|  |  |  |  |
| --- | --- | --- | --- |
| REFERENCES | TECHNIQUES | MODEL  DESCRIPTION | LIMITATIONS |
| [1]Kaur et al. | CNN-based  approach | CNN is most  commonly applied to analyze visual  imagery. | Light weight DWS-based CNN can provide more  efficient results. |
| Bhuiyan et al. | YOLO-v3 | YOLOv3 is | YOLOv4 |

|  |  |  |  |
| --- | --- | --- | --- |
| [2] | model | tremendously  fast and precise. Furthermore, you would simply tradeoff for both speed and accuracy by just modifying the model's sim,  without a need for retraining! | needs to be  compared using the proposed model. |
| Mata [3] | CNN model | Multi-layered  perceptron is standardized version of **CNNs.**  Multilayer perception are primarily fully interconnected networks, ensuring that every neuron throughout one layer is attached to all neurons in the subsequent  layer. | More  effective techniques are required for **improved** results. |
| Toppo et al. [4] | Mobile NetV2 | MobileNetV2 is  a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottlenecks  layers. | Revised parameter settings can improve the system performance. |
| Balaji et al. [5] | VGG-16 CNN | VGG Net is the name of a pre- trained  convolutional neural network (CNN).VGG  Net has been trained on ImageNet ILSVRC data  Set which includes image  Of 1000 classes split into three sets of 1.3 million training images, 100,0£D testing image and 50,0€D validation  images. | DWS  solution can  provide better results. |
| Fan et al. [6] | Residual  contextual awareness module | In information  and communication technologies, context awareness refers to the ability to analyze the condition of individuals, which could include and are therefore not exclusive to  users of the application | Due to the  constraints of the datasets, more processing is necessary to generate visualizations. |

# Proposed Methodology:

This project particularly builds on the Keras, TensorFlow, OpenCV, MobileNetv2 that are python and deep learning models. those Keras and TensorFlow helps to image data preprocessing, testing, and training. MobileNetV2 is a quite efficient structure that can be carried out to embedded devices. it also includes some of the algos with are sub parts of this libraries like Adam’s algorithms and so forth.

MobileNetV2 is an architecture of bottleneck depth separable convolution building of basic blocks with residuals. It has two types of blocks. Both blocks have three layers. The first one is 1x1 convolutions with “ReLU6”. The second layer contains the depth "convolution" and the third layer contains the 1x1 "convolution" with no nonlinearities. The first layer is the one-step residual block. The second layer is also used for compression as the residual block of step 2. The object detection methodology is to perform classification to determine the input class and regression to adjust the bounding box. Apart from the last fully connected layer, most backbone networks for discovery are networks for classification problems. The backbone network acts as a simple feature extractor for object detection problems by taking images as input and generating a feature map for each input image.

Pre-defined trained methods are usually used to extract functional maps with high-quality classification problems. This part of the model is called the base model. The base model is a MobileNetV2 network with "clean image" weights. ImageNet is particularly useful for image classification as it is an image database trained on hundreds of thousands of images. The estimated "bounding box" is compared to the "ground truth box" during training and the trained parameters are modified as needed during backpropagation. The kernel is used to produce a result showing the corresponding score for each pixel in each feature space, the presence of an element, and the corresponding bounding box size. The MobileNet project consists of two parts: the base model and the classifier. This model reuses the base model, cuts the hair and uses two fully connected layers.

## Training deep neural networks is expensive and time consuming because it requires a lot of computing power and other resources. Transfer learning based on deep learning has evolved to speed up network trains. Transfer learning is essentially the process of learning and predicting a new data set using a pre-trained model trained on a previous data set. The Imagenet dataset is used as a source of pretrained weights in most computer vision applications. This plan is used as a basis. The network employs pretrained a set of weights derived from a similar task–image classification– and has been trained on a certain collection of data like mask face detection in the ablation study of transfer learning. One explanation is that using pretrained weights from a closely related job improves feature extraction capabilities. Finally, a tall thick layer with two neurons and a softmax activation function is added to classify whether a person is wearing a mask. on figs. 4 schematically shows the proposed methodology.

# RESULT AND DISCUSSION

## **Data Collection:**

The collection of data is the first step in developing this model. The dataset collects information on persons who use masks and who don't. The collected data is divided into two groups: with and without mask.

Dataset consists of images with and without masks, this data set is taken from various sources like

Kaggle and google images. And some images with mask which can be used for training are artificially advanced by means of creating a python script to add masks to the images by means of making use of facial landmarks technique, created via PyImage Search reader Prajna Bhandary.

This dataset consists of **3833 images** belonging to two classes:

* + With mask: 1915 images
  + Without mask: 1918 images

A picture containing clothing, person, wearing, head covering

Description automatically generated A picture containing close

Description automatically generated

With mask without mask

Diagram

Description automatically generated

## Flow Diagram

**Figure1:**Phases and individual steps for building a COVID-19 face mask detector with computer vision and deep learning using Python, OpenCV, and TensorFlow/Keras

## **Preprocessing:**

# To find the types of attacks that can occur during data transfer, you need to train your model to predict the types of vulnerabilities. To train a model, you need records that you can use as a data set. For this reason, it is necessary to clean and preprocess the dataset before training the model. Many preprocessing steps include loading the dataset into the IDE, importing the library, reading the dataset, standardizing the dataset, encoding, removing nulls, splitting the dataset into training, evaluating the dataset, and extending its capabilities. Load datasets into the backend IDE using a variety of methods. It is different for each IDE. Libraries like NumPy, Pandas, and Sklearn are required to read, parse, and train models. Read the data set using the "Pandas.read\_csv(dataset name)" command, a method in the Pandas library. The StandardScalar command is used to standardize data sets.

# The preprocessing stage is the stage before training and testing the data. Preprocessing has 4 steps: resizing the image, converting the image to an array, preprocessing the input using MobileNetV2, and finally performing hot encoding on the label.

# Image scaling is an important preprocessing step in computer vision due to the effectiveness of the training model. The smaller the image size, the better the model's performance. Resizing the image in this study changes the image to 224×224 pixels.

# The next step is to convert all images in the dataset to an array. The image is converted to an array so that it can be called in the loop function. The image is then used for input preprocessing in MobileNetV2.

# And the last step in this step is to hot code the label. This is because many machine learning algorithms cannot use data labels directly. All input and output variables, including this algorithm, must be numeric. Labeled data is converted to numeric labels so that algorithms can understand and process the data.

# Data augmentation

Because the size of the training dataset is limited, data augmentation is used to expand the size of the training dataset by artificially manipulating the images in the dataset. "Shift, Contrast, Flip, Rotate, Scale and Blur" are all used to enhance the training image. You can reduce the size of the current model by rescaling the input image to 224\*224 and converting it to a single channel.

## **Split the data:**

## After preprocessing, the data is divided into two batches: training data, which accounts for 80% of the total, and testing data, which accounts for the remaining 20%. Images with and without masks are included in each batch.

## **Building the model:**

Construction of the training image generator for augmentation, the basicmode1 with Mobi1eNetV2, adding model parameters, compiling the model, training the model, and saving the model for future prediction are the six processes in generating the model.

## **Testing the model:**

There are phases in testing the model to ensure that it can predict well. Making predictions on the testing set is the initial stage.

**Table 2. Iteration of checking the loss and accuracy**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epoch | Loss | Accuracy | Val\_loss | Val\_acc |
| 1/10 | 0.1335 | 0.9529 | 0.0325 | 0.9909 |
| 2/10 | 0.0489 | 0.9868 | 0.0418 | 0.9870 |
| 3/10 | 0.0382 | 0.9885 | 0.0308 | 0.9896 |
| 4/10 | 0.0385 | 0.9858 | 0.0299 | 0.9909 |
| 5/10 | 0.0313 | 0.9901 | 0.0287 | 0.9922 |
| 6/10 | 0.0293 | 0.9911 | 0.0337 | 0.9922 |
| 7/10 | 0.0302 | 0.9918 | 0.0298 | 0.9918 |
| 8/10 | 0.0315 | 0.9895 | 0.0263 | 0.9935 |
| 9/10 | 0.0117 | 0.9960 | 0.0364 | 0.9896 |
| 10/10 | 0.0215 | 0.9934 | 0.0323 | 0.9896 |

## **Table 1** shows that the accuracy at increases at the beginning of the second epoch and the loss at decreases thereafter. The table can then be displayed on graph shown in **Figure 2.**

***Table 3***. **Model Evaluation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1-score** | **Support** |
| With mask | 0.98 | 0.99 | 0.99 | 383 |
| Without mask | 0.99 | 0.98 | 0.99 | 384 |
| Accuracy |  |  | 0.99 | 767 |
| Micro avg | 0.99 | 0.99 | 0.99 | 767 |
| Weighted avg | 0.99 | 0.99 | 0.99 | 767 |

## Chart, line chart Description automatically generated

**Figure 2:** COVID-19 face mask detector training accuracy/loss curves demonstrate high accuracy and little signs of overfitting on the data.

## Implementing the Model:

When the model is tested with the video streams, if a face is detected, the programmed moves onto the next step. Reprocessing will be done out on recognized frames

containing faces, including shrinking the image size, converting to the array, and pre-processing input using MobileNetV2. Predicting input data from the saved model is the next step. Predict the input image that has been processed using a model that has already been created. In addition, the video frame will be labelled with whether the person is wearing a mask, as well as the predicted percentage.

A picture containing text, indoor, screen, orange

Description automatically generated

**Figure 4**. Result of Predicting input data

A person wearing a mask

Description automatically generated with medium confidence

**Figure 4**. Result of Predicting input data

# Conclusion

The MobileNet-based Depth wise Separable Convolution Neural Network (DS-CNN) is introduced for mask detecting in facial pictures in this research. On introducing specific datasets, we relate our outcomes to the basic convolution layers. Per the results, the designed methodology excelled existing standard convolutions in testing. The prediction model is also linked to previous studies on a justified base classifier. The strategy demands additional operations to just provide visualizations, and it seems unable to differentiate between acceptable and unacceptable mask usage due to dataset restrictions. Our final goal would be to either generate face mask recognition datasets with numerous masks wearing states or to use zero shot understanding to make this project notice inaccurate mask wearing conditions.

To create our face masks detector, we trained a two-class model of people wearing mask and people now not wearing mask. subsequently after training the dataset with Keras/TenseFlow and deep learning. And undergoing the fine-tuning with MoblieNetV2, when the images is loaded using OpenCv and using the live camera output model offers the prediction with 99% accuracy.

## Future work

It is planned to implement temperature detection within the same model.

References

[1] S. Susanto, F. A. Putra, R. Analiaand I.K.L.N. Suciningtyas,

"Thc Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam," 2020 3rd International Confeimœ on Applied Engineering (ICAE), 2020, pp. 1-5, doi: 10.1109/ICAE50557.2020.9350556.

[2] A. Das, M. Was if Ansari and R. Basak, "Covid -19 Face Mask Detection Using TensorFlow, Keias and OpenCV," 2020 IEEE 17th India Council International Conference (INDICOJ, 2020, pp. 1-5, doi: 10.1109/INDICON49873.2020.9342585.

[3] M. M. Rahman, M. M. H. Manik, M. M. 1 slain, S. Mahmud and J. -H. Kim, "An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network," 2020 IEEE Internationd IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020, pp. 1 -5, doi: 10. l109/IEMTRONICS51293.2020.9216386.

[4] Kiran. Kallii au d Kiran, Ho kk a Vamsi and Sia, Devai‹ipal1i Mask Detection Using M ach in e Learning ( Septernber 1 7, 2021).

Available at SSRN: <https://ssm.com/abstracm3925736> o<http://dx.dotorg/10.2139/ssrn.3925736>

[5] H. Adusumalli, D. Kalyani, R. K. Sri, M. Pratapteja and P. V. R D.P. Rao, "Face Mask Detection Using

OpenCV," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (I CI CV), 2021, pp. 1304-1309, doi: 10.1 l09dCICV50876.2021.9388375.

1. [↑](#footnote-ref-1)