Intelligent System For Face Mask Detection

Rushi Mehta[1], Dheeraj Jain[2], Heenal Jain[3], Rekha Sharma[4]

[1,2,3] Student of Dept. of Computer Engineering, Thakur College of Engineering & Technology, Maharashtra, India.
[4]Associate Professor, Dept. of Computer Engineering, Thakur College of Engineering & Technology, Maharashtra, India.

Abstract - *Creating a model with high accuracy to identify whether a person has worn a mask or not. Achieving this with less number of images in the dataset is a hard task. To address these two problems, this paper will first introduce a dataset, with almost 480 masked and unmasked faces. We used the latest tools such as Mobile NetV2, Tensorflow, Keras - To create a convolution neural network (CNN). Original images in the datasets are augmented and additional images are created for training. Using OpenCV live video feed or image can be provided to the model and prediction can be made. OpenCV also helps in detection of faces, from the whole image/video only the part with the face is used by the model for prediction. Experimental results on the dataset shows that the proposed approach remarkably outperforms achieving prediction accuracy of 99%. This paper presents an overview of a system that detects faces and predicts masks worn or not.*

KeyWords*: *CNN, OpenCV, MobileNet, Deep Learning, Covid-19, Keras, Sklearn, Tensorflow.

1. INTRODUCTION

The Coronavirus disease bears the symptoms of fever, dry cough, tiredness, aches and pains, sore throat and shortness of breath. People are not following the instructions given by the government to wear masks and stop the spread of coronavirus. So our motive behind this project is to install cameras or use the existing ones to detect if people are wearing masks or not at airports and other public places, and they will only be allowed to enter if they are wearing masks properly.

According to data collected by the World Health Organization, the global pandemic of COVID-19 has severely impacted the world and has now infected more than eight million people worldwide. Wearing face masks and following safe social distancing are two of the enhanced safety protocols that need to be followed in public places in order to prevent the spread of the virus. With new variants such as Delta and now omicron, we need a system that will help in impeling to protocols that need to be followed while in public places. To create a safe environment that contributes to public safety, we propose an efficient computer vision-based approach focused on the real-time automated monitoring of people to detect face masks in public places by implementing the model.

1.1 Literature Survey:

The absence of large datasets of masked faces, and the absence of facial cues from the masked regions. To address these two issues, this paper first introduces a dataset, denoted as MAFA, with 30, 811 Internet images and 35,806 masked faces. the absence of large datasets of masked faces, and the absence of facial cues from the masked regions. To address these two issues, this paper first introduces a dataset, denoted as MAFA, with 30, 811 Internet images and 35,806 masked faces. The problem of face detection in the wild has been explored in many existing researches, and the corresponding face detectors have been tested on datasets of normal faces. On these datasets, some face detectors have achieved extremely high performances and it seems to be somehow difficult to further improve. However, the 'real wild' scenarios are much more challenging than expected for containing faces captured at unexpected resolution, illumination and occlusion. In particular, the detection of masked faces is an important task that needs to be addressed so as to facilitate applications such as video surveillance [1].

Their face mask identifier is least complex in structure and gives quick results and hence can be used in CCTV footage to detect whether a person is wearing a mask perfectly so that he does not pose any danger to others. Mass screening is possible and hence can be used in crowded places like railway stations, bus stops, markets, streets, mall entrances, schools, colleges, etc. By monitoring the placement of the face mask on the face, we can make sure that an individual wears it the right way and helps to curb the scope of the virus. A face mask recognition project that focuses on capturing real-time images indicating whether a person has put on a face mask or not. The dataset was used for training purposes to detect the main facial features (eyes, mouth, and nose) and for applying the decision- making algorithm. Putting on glasses showed no negative effect. Rigid masks gave better results whereas incorrect detections can occur due to illumination, and to objects that are noticeable out of the face [2].

In this paper, a hybrid model using deep and classical machine learning for face mask detection was presented. The proposed model consisted of two parts. The first part was for the feature extraction using Resnet50. Resnet50 is one of the popular models in deep transfer learning. While the second part was for the detection process of face masks using classical machine learning algorithms. The Support Vector Machine (SVM), decision trees, and ensemble algorithms were selected as traditional machine learning for investigation. The major drawback is not tray most of classical machine learning methods to get lowest consume time and highest accuracy. One of the possible future tasks is to use deeper transfer learning models for feature extraction and use the neutrosophic domain as it shows promising potential in the classification and detection problems [3].

The architecture consists of Mobile Net as the backbone; it can be used for high and low computation scenarios. In order to extract more robust features, we utilize transfer learning to adopt weights from a similar task face detection, which is trained on a very large dataset. The accuracy of the model is achieved and the optimization of the model is a continuous process and we are building a highly accurate solution by tuning the hyper parameters. This specific model could be used as a use case for edge analytics. Disadvantage is that it requires a large amount of data — if you only have thousands of examples, deep learning is unlikely to outperform other approaches. Is extremely computationally expensive to train. The most complex models take weeks to train using hundreds of machines equipped with expensive GPUs[4].

1.2 Convolutional Neural Network:

CNN plays a prominent role in computer vision related pattern recognition tasks, because of its highly characterized feature extraction capability and less computation cost.[5] CNN uses convolution kernels to coil with the original pictures or featured maps to extract high-level features. However, how to create a better convolutional neural network architecture still remains as a major question. Inception network proposes to allow the network to learn the best combination of kernels. As object detectors are usually deployed on the mobile or embedded devices, where the computational resources are very less, Mobile Network (MobileNet) is proposed. It uses depth wise convolution to extract features and channel wise convolutions to adjust channel number, so the computational cost of MobileNet is much lesser than networks using standard convolutions.

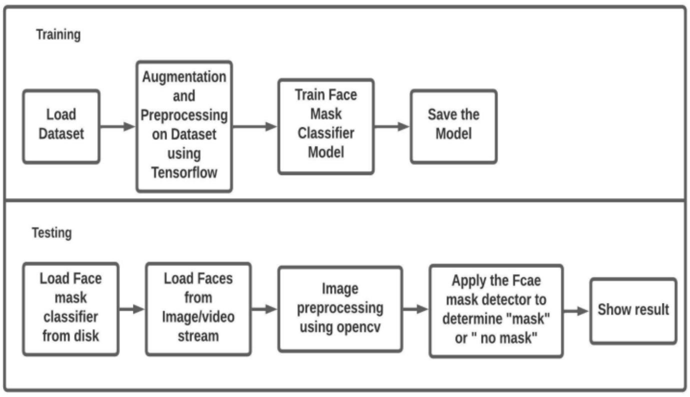
1.3 OpenCV:

OpenCV contains various tools to solve computer vision problems. It contains low level image processing functions and high-level algorithms for face detection, feature matching and tracking. OpenCV stands for Open-Source Computer Vision. To put it simply, it is a library used for image processing. In fact, it is a huge open-source library used for computer vision applications, in areas powered by Artificial Intelligence or Machine Learning algorithms, and for completing tasks that need image processing.[6] As a result, it assumes significance today in real-time operations in today's systems. Using OpenCV, one can process images and videos to identify objects, faces, or even the handwriting of a human.

1.4 Tensor Flow: TensorFlow allows developers to create data flow graph structures that describe how data moves through a graph or a series of processing nodes.[7] Each node in the graph represents a mathematical operation, and each connection or edge between nodes is a multidimensional data array, or tensor. TensorFlow provides all of this for the programmer by way of the Python language. Python is easy to learn and work with, and provides convenient ways to express how high-level abstractions can be coupled together. Nodes and tensors in TensorFlow are Python objects, and TensorFlow applications are themselves Python applications.

1.5 Keras: Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation. Keras is relatively easy to learn and work with because it provides a python frontend with a high level of abstraction while having the option of multiple back-ends for computation purposes [8]. This makes Keras slower than other deep learning frameworks, but extremely beginner friendly. Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine. It also allows use of distributed training of deep-learning models on clusters of Graphics processing units (GPU) and tensor processing units (TPU).

2. Approach



2.1 Deep Learning Framework:

CNN (Convolutional Neural Network) has many versions of pre-trained and well-architected networks for example AlexNet, ResNet, Inception, LeNet, MobileNet and so on. In our case we have chosen the MobileNetV2 due to its lightweight and very efficient mobile-oriented model.

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 bottleneck layers. The intermediate expansion layer uses lightweight depth wise convolutions to filter features as a source of non-linearity. As a whole, the architecture of MobileNetV2 contains the initial fully convolution layer with 32 filters, followed by 19 residual bottleneck layers.

2.2 Dataset: As shown in Fig -1 our dataset consists of 2 folders. First is with a mask and second is no mask, so we can say it was labeled data. Both categories consisted of 480 images each. Images in each category consisted of people with different age groups and gender.

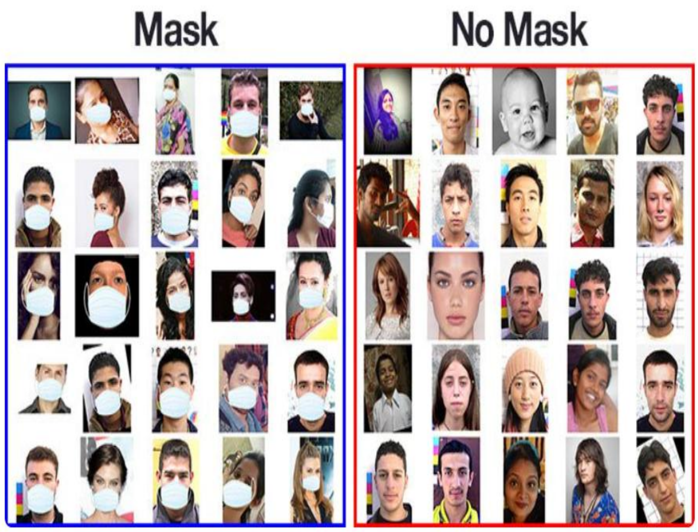


Fig -1: Dataset

2.3 Data Pre-processing: As our dataset consisted of only 480 images the accuracy and robustness of the model was very less. So using keras's ImageDataGenerator function we augmented each image on different parameters such as rotation, zoom, width, height, shear and horizontal flip. Final resulting dataset consisted of around 900 images with variations for our model to train upon.

```
# construct the training image generator for data augmentation
aug = ImageDataGenerator(
        rotation_range=20,
        zoom_range=0.15,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.15,
        horizontal_flip=True,
        fill_mode="nearest")
```

Fig -2: Augmenting images

2.4 Model Creation: First the base model was created using MobileNetV2 where the input shape of images was put as (224, 224, 3). Then the head models created consisted of 5 layers as shown in Fig -3. First was the AveragePolling2D which was used to down sample the input by taking average values. Flatten was used to convert our two-dimensional data into one. Next layer used is Dense which is a network layer. It is deeply connected with its predecessor layers and feeds all outputs from the previous layer to all its neurons and each neuron further provides one output to the next layer. Activation functions often used in dense layers are ReLU and SoftMax. Conventionally, ReLU is used as an activation function in NNs, with SoftMax function as their classification function. Dropout layer is used to prevent the model from overfitting [9].

```
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
```

Fig -3: Model Creation

2.5 Accuracy Overview:

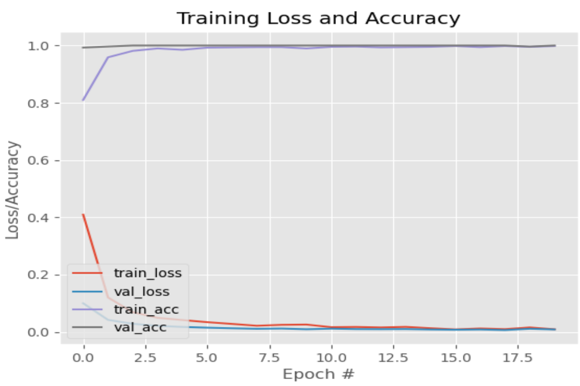


Chart 1- Accuracy Overview

As seen in Chart 1 around 15 Epochs we hit a plateau, training accuracy remained the same at 99% so we stopped at 20 Epochs.

2.6 Results: We split our data into 80% for training and 20% for testing. Chart1 depicts the accuracy of the model on testing data after training is completed. On real time data also, we produced the same accuracy. The result for the Fig.4 with mask was about 99.95% and the result for Fig.5 without mask was almost the same as that of with mask around 99.91%.

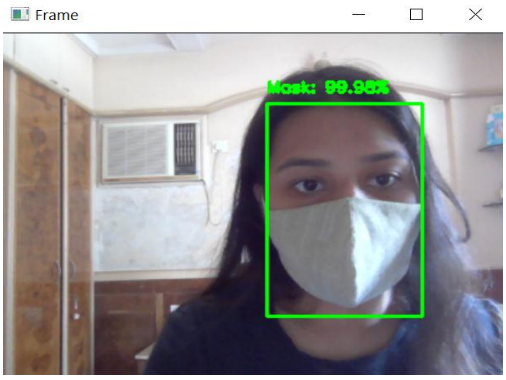


Fig -4: Output with Mask

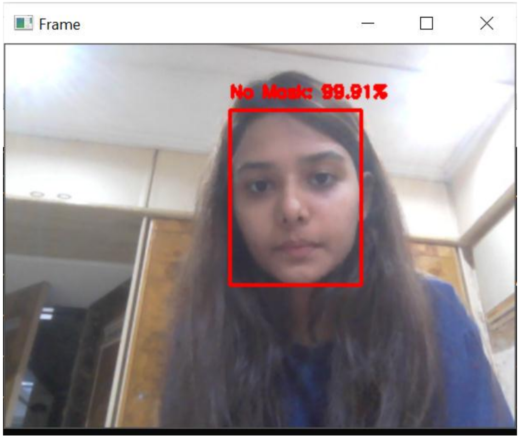


Fig -5: Output without mask

3. Application:

Different public areas where we can implement this Face Mask Detection method are as follow:

3.1 Airport: The Face Mask Detection System can be used at airports to detect passengers violating rules stuck to wearing masks. Face data of passengers can be captured in the system at the entrance. However, their image shall be transferred to the field authorities so that they can take quick action, If a rubberneck is plant not wearing a mask.

3.2 Hospital: Using Face Mask Detection System, Hospitals can cover if quarantined people needed to wear a mask are doing so or not. The same holds good for covering staff on duty too.

3.3 Offices & Workplaces: This system can be used at office demesne to ascertain if workers are maintaining safety norms at work. It monitors workers without masks and sends them a memorial to wear a mask.

3.4 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.

4. Conclusion:

The proposed system can help maintain a secure environment and ensure individual's protection by automatically monitoring public places, offices, etc which helps and assists in security checks and also helps police by minimizing their physical surveillance and public areas where surveillance is required by means of camera feeds in real-time. The solution has the potential to properly monitor and reduce violations in real-time, so the proposed system would improve public safety through saving time and helping to reduce the spread of coronavirus. This solution can be used in places like temples, shopping complexes, metro stations, airports, etc.

Technologies used in this system are TensorFlow, OpenCV, Keras, MobilenetV2. We got the desired accuracy from this system and it holds true for live image and video streams as well. The method used to create a model that can learn to classify images based on labeled data and the method to overcome limited size of dataset for training can be used for different categories of images as well.

REFERENCES

[1] Shiming Ge, Jia Li Qiting Ye, Zhao Luo "Detecting Masked Faces in the Wild with LLE-CNNs" 2017 IEEE conference on computer vision and pattern recognition (CVPR).

[2] Madhura inamdar, Ninad Mehendale "Real-Time Face Mask Identification Using Face MaskNet Deep Learning Network" 2021 10th IEEE international conference on communication systems and network technologies (CSNT).

[3] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, Nour Eldeen M. Khalifa "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic" 2020

[4] Vinitha, Velantina "COVID-19 FACEMASK DETECTION WITH DEEP LEARNING AND COMPUTER VISION" 2020

[5] Sakshi Indolia, Anil Kumar Goswami, S.P.Mishra, Pooja Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach" 2018 international conference on computational intelligence and data science.

[6] Naveenkumar M., Vadivel Ayyasamy "OpenCV for Computer Vision Application" 2015 proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15)

[7] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, "A system for large-scale machine learning" 2016 proceedings of the 12th USENIX Symposium on operating systems design and implementation (OSDI'16).

[8] Ketkar N. (2017) Introduction to Keras. In: Deep Learning with Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2766-4_7

[9] Abien Fred M. Agarap, "Deep Learning using Rectified Linear Units (ReLU)" 2019

BIOGRAPHIES

Rushi Mehta is pursuing B.E. in Computer Engineering. He has done various projects in the area like Artificial Intelligence. His area of interest includes Machine Learning, Augmented Reality.

Dheeraj Jain is pursuing B.E. in Computer Engineering. He has done various projects in the area like Genetic Algorithms. His area of interest includes Artificial Intelligence.

Heenal Jain is pursuing B.E. in Computer Engineering. She has done various projects in the area like Web-Development. Her area of interest includes UI/UX.

Dr. Rekha Sharma is working as Associate Professor and Activity Head (R&D) in Thakur College of Engineering and Technology. She has more than 20 years of experience in Teaching and Industry. She has more than 40 publications in national and International Journals and conferences. Her area of Interest includes education technology, software localization and natural language processing.