DETECTION OF FACE MASK USING SOFT COMPUTING TECHNIQUE

A.Dhok[1], A. Bohrupi[1], B. Nagale[1], M. Mohod[1], N. Junare[1], S.Thakare[1], S. Patil[1], S.Gadge[1], V. Solankke[1], Y.Lokhande[1], S.K. Nanda[2]

[1]Department of Computer Science and Engineering, P.R. Pote Patil College of Engineering and Management, Amravati, MS, India

[2] Department of Electronics and Telecommunication Engineering, P.R. Pote Patil College of Engineering and Management, Amravati, MS, India

ABSTRACT

COVID-19 pandemic has rapidly affected our day-to-day life-disrupting world trade and movements. Wearing masks has been mandatory since the pandemic was discovered. As a social distance, mask detection is equally important to prevent any further surge in COVID cases. It also proves beneficial in detecting masks in public places like hospitals, parks, etc. Therefore, face mask detection has become a crucial task to help global society. This paper presents a simplified approach to achieve this purpose using some basic Machine Learning packages like TensorFlow, Keras, OpenCV and Scikit-Learn, and Deep Learning. The proposed method detects the face in real-time correctly and then identifies if it has a mask on it or not. Using this, we can alert pedestrians to wear masks whenever they step out of their houses.

***Keywords**: Face mask detection, deep learning, OpenCV, TensorFlow, Keras*

1. Introduction

The coronavirus COVID-19 pandemic is causing a global health crisis. Governments all over the world are struggling to stand against this type of virus. The protection from infection caused by COVID-19 is a necessary countermeasure, according to the World Health Organization (WHO).As we know, it is a pandemic that has claimed millions of lives in the year 2020. Wearing a face mask has been identified as a successful method of preventing the spread of COVID among people. It is strongly recommended to wear a mask in public places. Most people follow the guidelines and wear masks. Some people do not wear it while others wear it incorrectly which doesn't cover their nose/mouth as it should.

COVID-19 mainly spreads through droplets produced as a result of coughing or sneezing by an infected person. This transfers the virus to any person who is in direct close contact (within one-meter distance) with the person suffering from coronavirus. Because of this, the virus spreads rapidly among the masses. With the nationwide lockdowns being lifted, it has become even harder to track and control the virus. Face masks are an effective method to control the spread of the virus. It has been found that wearing face masks is 96 % effective to stop the spread of the virus. Governments, all over the world, have imposed strict rules that everyone should wear masks while they go out. But still, some people may not wear masks and it is hard to check whether everyone is wearing masks or not. In such cases, computer vision will be of great help.

There are no efficient face mask detection applications to detect whether the person is wearing a face mask or not. This increases the demand for an efficient system for detecting face masks on people for transportation means, densely populated areas, residential districts, large-scale manufacturers, and other enterprises to ensure safety. This project uses machine learning classification using OpenCV and Tensorflow to detect face masks on people.

Public use of face masks has been common in India and other nations in the world since the beginning of the new coronavirus disease outbreak. We now know from recent studies that a significant portion of individuals with coronavirus lack symptoms ("asymptomatic") and that even those who eventually develop symptoms ("pre-symptomatic") can transmit the virus to others before showing symptoms, according to the advisory published by the Health Centre. "This means that the virus can spread between people interacting nearby — for example, speaking, coughing, or sneezing — even if those people are not exhibiting symptoms". The recent information also gives a trace of a new strain of coronavirus, the mutant coronavirus, in which the virus has changed its structure and become mutant. The new strain is not even able to detect using the RT-PCR test we use now. So it is inevitable for the people

Our project aims to train a model on images of people wearing masks and develop an interface to identify the faces of people wearing the mask correctly, wearing it incorrectly, or not wearing a mask. Face detection and Face Recognition are often used interchangeably but these are quite different. Face detection is just part of Face Recognition. Face recognition is a method of identifying or verifying the identity of an individual using their face. Various algorithms can do face recognition but their accuracy might vary. Here I am going to describe how we do face recognition using deep learning. Further, the different hyperparameters are tried for the model. The model has used OpenCV to fulfill the purpose of using the video stream for capturing the frames in the video stream

2. Related work

Generally, most of the publications focus on face construction and identity recognition when wearing face masks. In this research, our focus is on recognizing the people who are not wearing face masks to help in decreasing the transmission and spreading of the COVID-19. Researchers and scientists have proved that wearing face masks help in minimizing the spreading rate of COVID-19. The various researcher work on the project of facemask detection to help to control the spread of the coronavirus, some of their work is as follow:

Author	**Year**	**Methodology**	**Dataset**	**Accuracy**	**Conclusion**
Teboulbi et al.	2021	Real-Time Implementation of AI-Based Face Mask Detection and Social Distancing Measuring System for COVID-19 Prevention	RMFD Dataset SMFD Dataset	99%	Evaluated the numerical results, the best models are tested on an embedded vision system consisting of a Raspberry Pi board and webcam where efficient real-time deep learning-based techniques are implemented with a social distancing task to automate the process of detecting masked faces and maintaining distance between people.
Talahua et al.	5 June 2021	Facial Recognition System for People with and without Face Mask in Times of the COVID-19 Pandemic.	-	99.52%	This prototype system allows for the facial recognition of people with and without a mask, and could be used as a low computational consumption proposal for personnel access control.
Susanto et al.	8 Oct. 2020	The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam	-	98.45%	The algorithm can detect and distinguish between a non-wearing and a wearing mask precisely with any condition of the surrounding environment.
Shah et al.	2020	Detection of Face Mask using Convolutional Neural Network	Kaggle, RMFD dataset	99%	demonstrated a facemask detector using Convolutional Neural Network and move learning techniques in neural organizations

loeyet al.	1 Jan 2021	A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic	RMFD Dataset SMFD Dataset	99.49%	The proposed model super passed the associated works in terms of testing accuracy.

3. Proposed Methodology

I. Dataset

The dataset used for the current project consists of a total of 4,595 images. The dataset contains 2,706 images with people wearing a mask and the rest of the 1,889 images with the people who do not wear a mask. Images in the dataset are collected from the Kaggle and the various operations are performed on the images to increase the number of the images in the dataset while training. The abstract view with the mask dataset used for training the model is shown in **Figure 1**

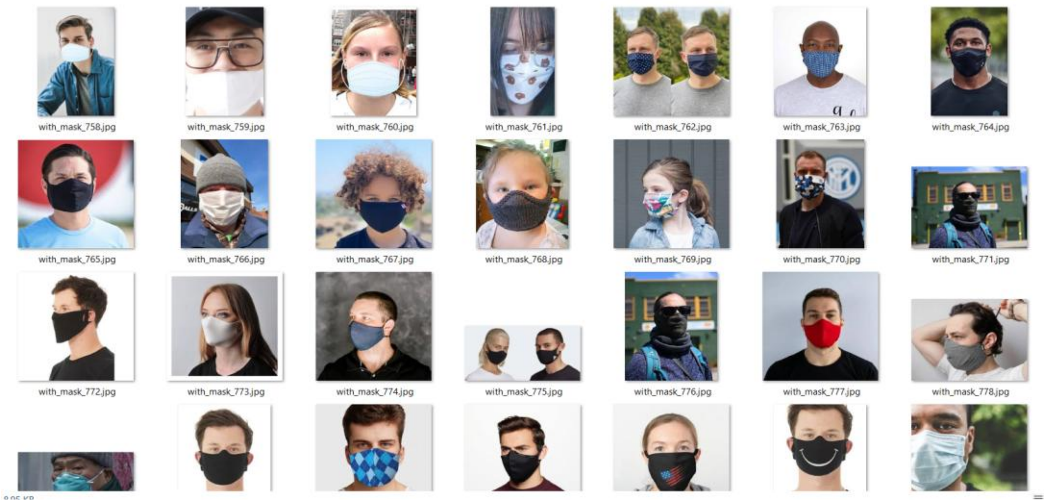


Figure 1: The abstract view with mask dataset.

Figure 1 mostly contains a front face pose with a single face in the frame and with the same type of mask having different colors and shapes. **Figure 2** contains the sample images of a person without a mask.

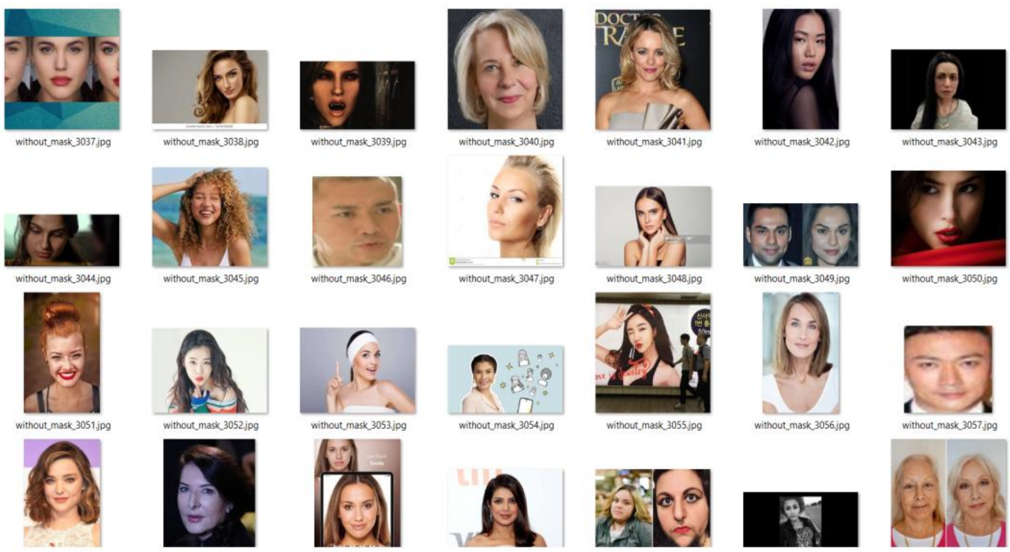


Figure 2: Sample images of people without masks.

II. Incorporated packages

A. TensorFlow

TensorFlowis an open-source end-to-end platform for creating Machine Learning applications. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on the training and inference of deep neural networks. It allows developers to create machine learning applications using various tools, libraries, and community resources.

B. Keras

Keras gives fundamental reflections and building units for the creation and transportation of ML arrangements with high iteration velocity. It takes full advantage of the scalability and cross-platform capabilities of TensorFlow. The various modules in Keras are used to pre-process the input and to perform various operations in the current method. Along with the conversion of the class vector to the binary class matrix in data processing, it helps to compile the overall model.

C. Sklearn

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelingincluding classification, regression, clustering, and dimensionality reduction. Here we use scikit-learn to train our model using train_test_split.

D. OpenCV

OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. The proposed method makes use of these features of OpenCV in resizing and color conversion of data images. We used OpenCV tocapture the live streaming and detect the face mask.

III. Proposed method

The proposed method divides the project into two sections. The first one is a data training model and the remaining one is a deployment of the model. Data training sections include the loading of the face mask detection dataset from disk, training a model on this dataset, and then serializing the face mask detector to disk. In the deployment section, 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 a mask or without a mask.

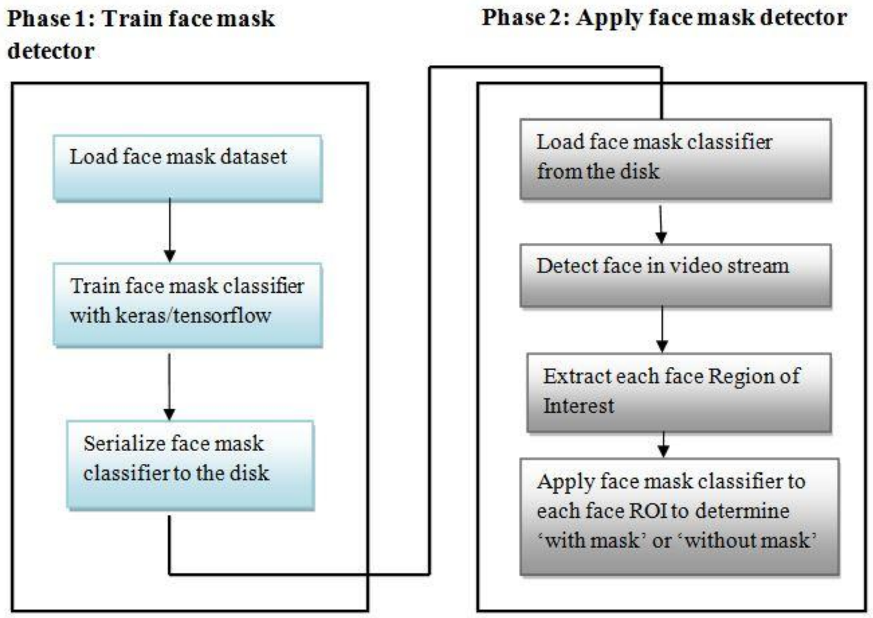


Figure: 3 Block Diagram of Proposed Methodology

A. Pre-processing of data

Data pre-processing is the process of transforming raw data into an understandable format. It can be in any form like tables, images, videos, graphs, etc. This organized information fits in with an information model or composition and captures relationships between different entities. It is also an important step as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms. Here we pre-process our image format data using NumPy and OpenCV.

- **Data visualization**

Data visualization is the process of transforming abstract data into meaningful representations using knowledge communication and insight discovery through encodings. It is helpful to study a particular pattern in the dataset.The dataset in theform of several images is classified into two categories –'with mask' and 'without a mask'.

The statement 'for an image in os.listdir(path)' in the loop categorizes the list of directories in the specified data path. So the categories will look like categories=['with_mask','without_mask']. Here the categories are a list variable that has value ['with_mask','without_mask']. Thenos.path. join() method joins one or more path components intelligently. This method concatenates various path components with exactly one directory separator ('/') following each non-empty part except the last path component.After that, the image is loaded into the model using the load image function from TensorFlow,Keras, pre-processing, and imagelibrary.

- **Image reshaping**

Before processing aninput image it must be in the proper format. Some images captured by a camera and fed to our AI algorithm vary in size, therefore, we should establish a base size for all images fed into our AI algorithms. Principally, our machine learning models train faster on smaller images. An input image that is twice as large requires our network to learn from four times as many pixels — and that time adds up. So each image in input data is resized using taget_size=(224,224), where 224 is the height and width of the image.

Converting an image to an array is an important task to train a machine learning model based on the features of an image. We mainly use the NumPy library in Python to work with arrays so we can also use it to convert images to an array. Other than NumPy, we can also use the Keras library in Python for the same task. In this proposed method we used both the img_to_array method from TensorFlow, Keras, pre-processing, image as well as the NumPy library.Then, by using the pre-process_input method, to adequate your image to the format the model requires.

B. Training model

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.Usingthe scikit-learns convenience method, segment our data into 80% training and the remaining 20% for testing. During training, we applied on-the-fly mutations to our images in an effort to improve generalization. This is known as data augmentation, where the random rotation, zoom, shear, shift, and flip parameters are used.

then prepared MobileNetV2 for fine-tuning, Fine-tuning setup is a three-step process:

Load Mobile Net with pre-trained ImageNet weights, leaving off the head of the network.

Construct a new FC head, and append it to the base in place of the old head.

Freeze the base layers of the network. The weights of these base layers will not be updated during the process of backpropagation, whereas the head layer weights will be tuned.

After fine-tuning, we compiled our model with the Adam optimizer, a learning rate decay schedule, and binary cross-entropy. Face mask training is launched via model. Fit() and Aug. Flow. model. Predict()make predictions on the test set, grabbing the highest probability class label indices. Then, we print a classification report in the terminal for the inspection.model.save() serializes our face mask classification Model to disk.

C. Deployment

Once the data training is complete, we created a new file called detect_face_mask.py.

Driver script requires three TensorFlow/Keras imports toload our Mask Net model and pre-process the input image. OpenCV is required for display and image manipulations. OpenCV opens the camera and starts thelive video streaming.

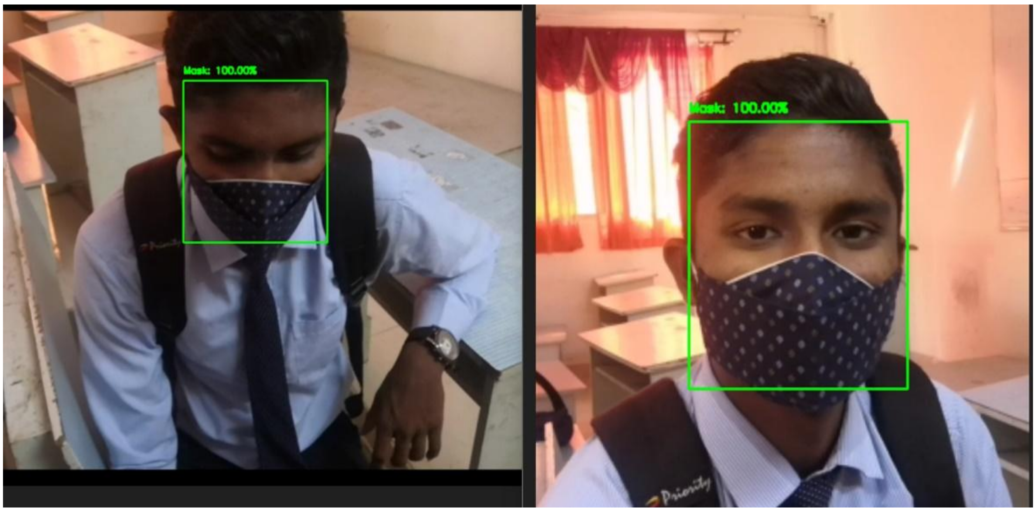
Then compute the bounding box value for a particular face and ensure that the box falls within the boundaries of the image. Extract the face ROI via NumPy slicing, Pre-process the ROI the same way it did during training, and after that Perform mask detection to predictwith maskor without_mask. Once all detections have been processed, display the output on live streaming video.

4. Results and Discussion

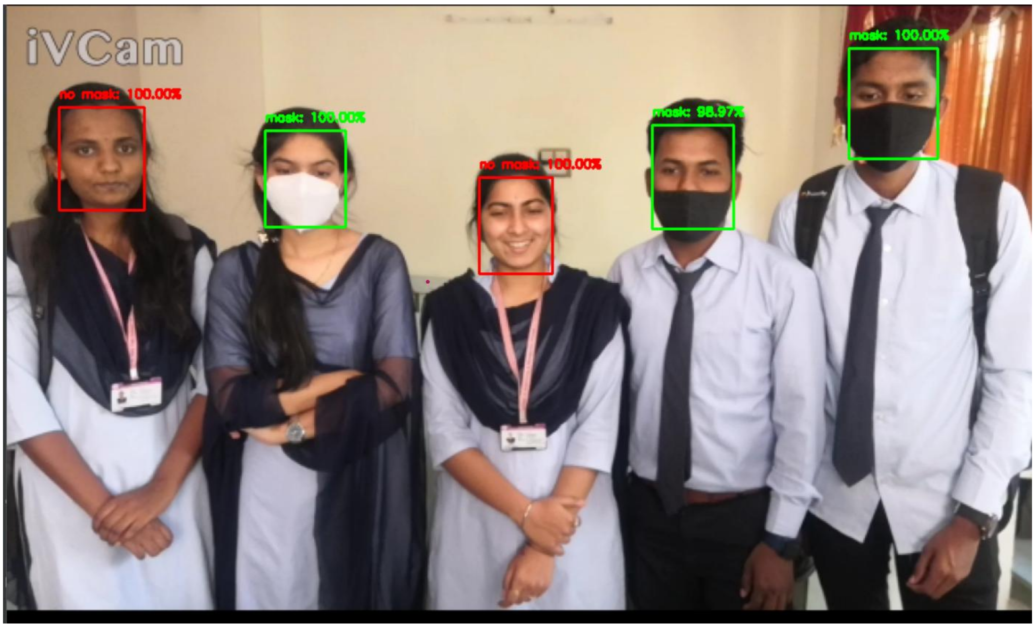
The model is trained and validated with a training accuracy of 98%, as the number of images in the dataset is large enough to train the model with high accuracy. Fig 3, shows the graphical representation of epoch against the loss/accuracy while training the model. The number of epochsi.e 20 and batch size 32 helps to achieve the maximum accuracy.

After training the model, the achieved training accuracy is 0.9827 and loss is 0.0470. The validity loss and accuracy are 0.0368 and 0.9860. The validity loss is less than the training loss so, it shows some situations of overfitting. 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 an annotated mask or face covered by hand.**Figure: 4** shows graph plotted against the epoch and loss/accuracy whereas **Figure: 5**shows**s**creenshots of identification of mask and no mask in real-time videos.

Figure: 4 Graph plotted against the epoch and loss/accuracy.


Mask: 100.00%
Mask: 100.00%


Mask: 100.00%


iVCam
no mask: 100.00%
mask: 100.00%
no mask: 100.00%
mask: 98.97%
mask: 100.00%

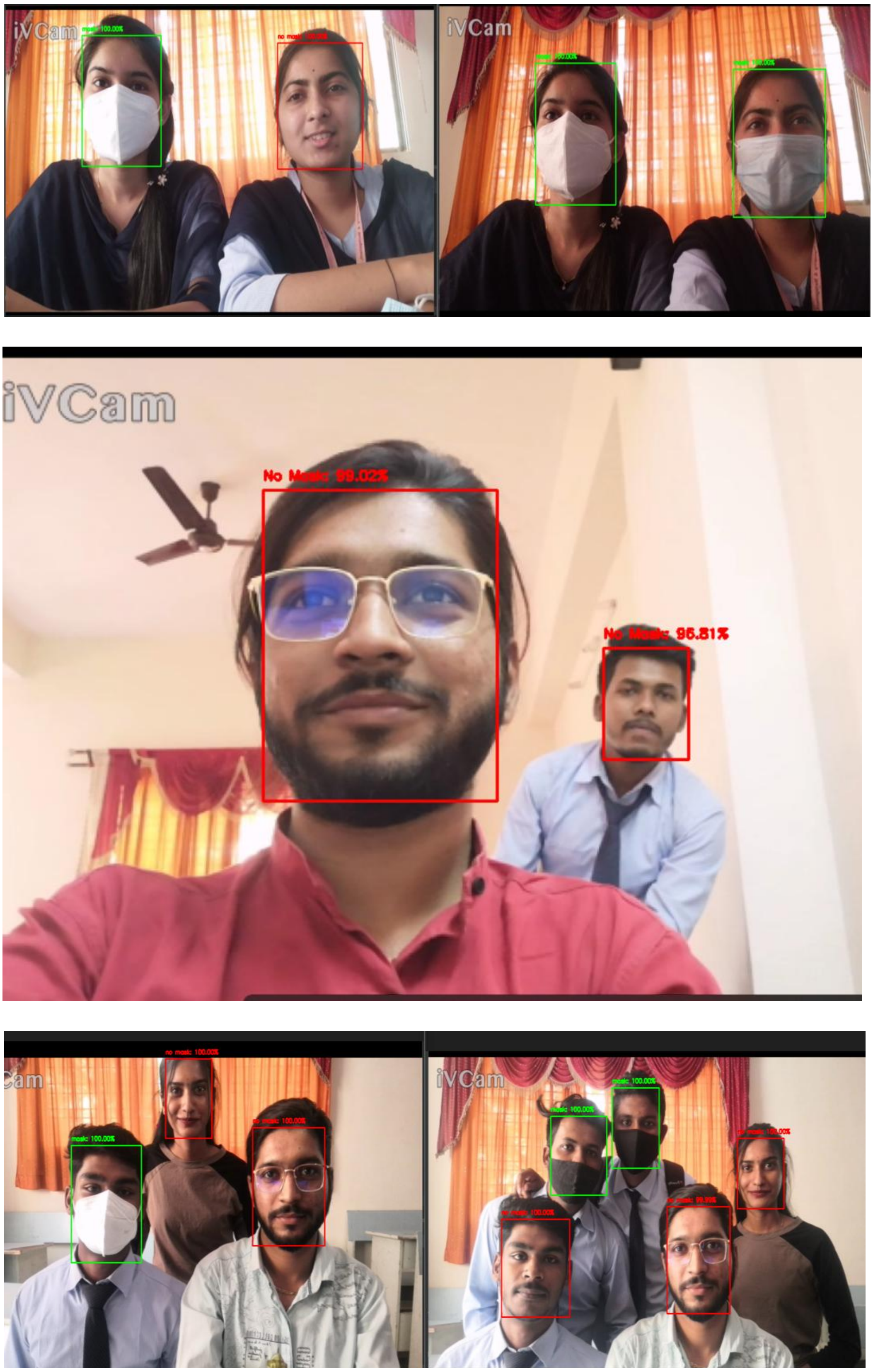


Figure 5: Screenshots of identification of mask and no mask in real-time videos

Performance metrics details and performance metrics comparison are depicted in **Figure 6** and **Figure 7** below.

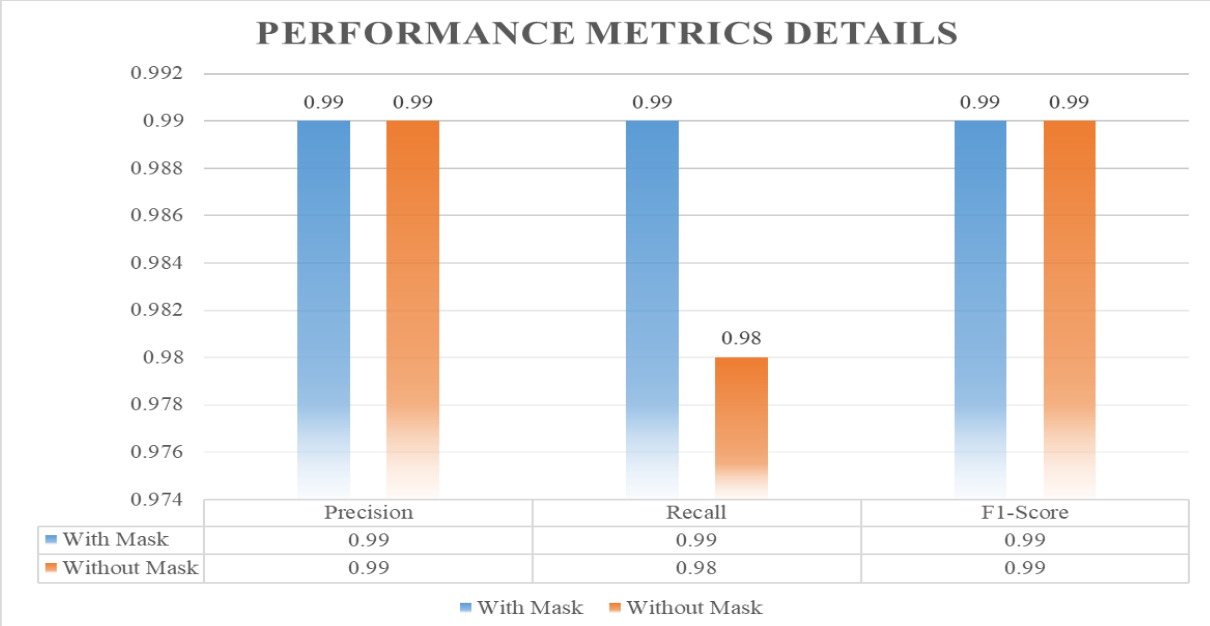


Figure: 6 Performance Metrics Details

Figure: 7 Performance Metrics Comparison

5. Conclusion and future scope

To mitigate the spread of the COVID-19 pandemic, measures must be taken. A facemask detector is modeled. The model is trained in an authentic database. OpenCV, tensor flow, Keras, and CNN are used to determine whether the people are wearing a face mask or not. The model is tested with real-time photos and video. The accuracy of the model is achieved at 99%.

The model optimization is an ongoing process and accurate solutions for tuning hyperparameters are being built. This particular model can be used as utility edge case analytics. This face mask detector can be deployed in many areas like hospitals, parks, shopping malls, airports, and other heavy traffic places to monitor the public and to avoid the spread of the disease by checking who is following basic rules and who is not.

References

1. M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic," Measurement, vol. 167, Article ID 108288, 2021.View at: Publisher Site | Google Scholar
2. B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," Sensors, vol. 20, no. 18, p. 5236, 2020.View at: Publisher Site | Google Scholar
3. X. Zhang, H. Saleh, E. M. Younis, R. Sahal, and A. A. Ali, "Predicting coronavirus pandemic in real-time using machine learning and big data streaming system," Complexity, vol. 2020, Article ID 6688912, 10 pages, 2020.View at: Google Scholar
4. M. Razavi, H. Alikhani, V. Janfaza, B. Sadeghi, and E. Alikhani, "An automatic system to monitor the physical distance and face mask wearing of construction workers in COVID-19 pandemic," 2021, https://arxiv.org/abs/2101.01373.View at: Google Scholar D. Sridhar and D. Gurdasani, "Herd immunity by infection is not an option," Science, vol. 371, no. 6526, pp. 230-231, 2021.View at: Publisher Site | Google Scholar
5. J. Wong and N. Wong, "The economics and accounting for COVID-19 wage subsidy and other government grants," Pacific Accounting Review, vol. 33, no. 2, pp. 199–211, 2021.View at: Google Scholar
6. M. Piraveenan, S. Sawleshwarkar, M. Walsh et al., "Optimal governance and implementation of vaccination programs to contain the COVID-19 pandemic," Royal Society Open Science, vol. 8, no. 6, Article ID 210429, 2021.View at: Publisher Site | Google Scholar
7. A. Echtioui, W. Zouch, M. Ghorbel, C. Mhiri, and H. Hamam, "Detection methods of COVID-19," SLAS TECHNOLOGY: Translating Life Sciences Innovation, vol. 25, no. 6, pp. 566–572, 2020.