



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 8, August 2021

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.569

9940 572 462

6381 907 438

ijrset@gmail.com

www.ijrset.com



Real Time Face Mask Detection Using Machine Learning Techniques

Likith Reddy J¹, Raghavendra Rao B G²

P.G. Student, Department of Computer Applications, Sir M Visvesvaraya Institute of Technology, Krishnadevaraya Nagar, Hunasamaranahalli, Bengaluru, Karnataka, India¹

Assistant Professor, Department of Computer Applications, Sir M Visvesvaraya Institute of Technology, Krishnadevaraya Nagar, Hunasamaranahalli, Bengaluru, Karnataka, India²

ABSTRACT: The Face Mask detection is a computer vision technology which is used to monitor the people, whether the person is wearing the mask or not which is very essential for the safety of himself and other individuals during this covid-19 pandemic. We use machine learning concepts to get the desired output, for this we require a dataset of faces with masks and without masks, and we perform data pre-processing before working on the data so that we could train the machine learning model with this dataset to find the difference between the face with mask and without mask. Once the model is trained it can able to notice the differences in the face, then we will scan the faces of the people using in-built camera and the machine learning model classify each face as with mask and without mask.

KEYWORDS: Facedetection, Mask Detection, Object Detection

I. INTRODUCTION

The Face Mask Detection is basically an update to the face detection, vehicle Detection system, etc., which are currently used by Traffic Police, Crime Investigation Department, Police, etc. for security purpose.

This system could be used in public places to monitor the people who are not wearing the mask. This monitoring is very important as we are currently dealing with covid-19 pandemic. We are using a large dataset to train the machine learning model to detect faces with masks and without masks. The result will be displayed with the keywords 'Mask and No Mask', along with the predicted accuracy of the model. The face will be displayed within a red colored box if the person is not wearing the mask, and it will show within green colored box for wearing the mask.

Paper is organized as follows. Section II describes the related work that has done in this machine learning field for Object and Face detection. The implementation techniques for this project requires a large dataset to train the machine learning model which classify the faces of the people is given in Section III. Section IV presents experimental results showing results of people tested. Finally, Section V presents conclusion.

II. RELATED WORK

This section is a basic overview of the major techniques used in the face recognition system that apply mostly to the front face of the human being. The methods include neural networks, hidden Markov model, face matching done geometrically and template matching. Eigenface is one of the most widely used methods in face recognition and detection which are broadly called as the principle components in mathematical terms. The eigenvectors are ordered to represent different amounts of the variations in the faces. Neural networks are highly used in the face recognition and detection systems. An ANN (artificial neural network) was used in face recognition which contained a single layer Which shows adaptiveness in crucial face recognition systems. The face verification is done using a double layer of WISARD in neural networks. Graph matching is other option for face recognition. The object as well as the face recognition can be formulated using graph matching performed by optimization of a matching function. Hidden Markov Models is the way by which stochastic modelling of no stationary vector time series based on HMM model applied to the human face recognition wherein the faces gets divided into parts such as the eyes, nose, ears, etc The face recognition and correct matching is 87% correct as it always gives out the best and right choice of face detection through stored dataset. Or else the relevant model reveals the identity of the face. The geometrical feature matching is

the technique which is based on the geometrical shapes of the face. The geometrical face configuration has sufficient dataset for face detection and recognition system. This is one of the commonly used method of the face recognition and detection. This system apparently gives satisfactory results

III. METHODOLOGY

Data Preprocessing is a very vital step in Machine Learning. Most of the real-world data that we get is messy, so we need to clean this data before feeding it into our Machine Learning Model. This process is called Data Preprocessing or Data Cleaning. Here we have two categories in a dataset which are 'with mask' and 'without mask', what we are going to do is, we first set the size of the image as (224, 224) by using 'load_img' function from Keras library and then we will convert the images to arrays by using 'img_to_array' function provided by Keras.

After converting the images to arrays, we create two lists called 'data' and 'labels'. We append all the image arrays into 'data' list and the corresponding image category into the 'labels' list. Now all the data list has numerical values of images but in labels list we have alphabetical values, in order to convert these into numerical values we use 'LabelBinarizer' method which is from SKlearn. Then we convert both the lists into NumPy arrays cause Deep Learning models work only with arrays.

```
In [17]: # perform one-hot encoding on the labels
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

data = np.array(data, dtype="float32")
labels = np.array(labels)
data
```

(a) Using LabelBinarizer to convert label values into numerical.

```
In [11]: labels
Out[11]: array([[0., 1.],
                [1., 0.]],

                [[0., 1.],
                [1., 0.]],

                ...,

                [[1., 0.],
                [0., 1.]],

                [[1., 0.],
                [0., 1.]],

                [[1., 0.],
                [0., 1.]], dtype=float32)
```

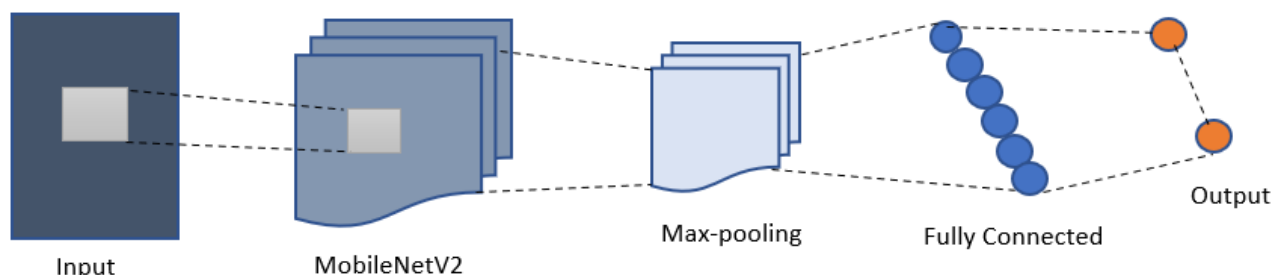
(b) Converted Labels in array format

The `train_test_split` is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, `Sklearn train_test_split` will make random partitions for the two subsets. However, you can also specify a random state for the operation. We have manually specified the random state as 42 and split the dataset into 20% for Testing and 80% for Training the model.

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset. The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.

We are going to follow convolutional neural network but with a small change, instead of convolution we are using MobileNetV2. The flow will be like we will send the processed input array into the MobileNetV2 and then do Max pooling then Flatten it and later create a Fully-connected layer and get the output.

The Initial Learning rate = $1e-4$
EPOCHS = 20



(C) Convolution neural network using MobileNetV2

MobileNets are a family of neural network architectures released by Google to be used on machines with limited computing power, like mobile devices. They strive to provide state of the art accuracy, while requiring as little memory and computing power as possible. This makes them a very fast family of networks to use for image processing. MobileNets achieve this performance by reducing dramatically the number of learnable parameters, which also makes them faster and easier to train compared to more traditional networks.

Maximum pooling, or max pooling, is a pooling operation that calculates the maximum, or largest value in each patch of feature map. The results are down sampled or pooled feature maps that highlight the most present feature in the patch, not the average presence of the feature as in the case of average pooling. Once the pooled featured map is obtained, the next step is to flatten it. Flattening involves transforming the entire pooled feature map matrix into a single column (vector) which is then fed to the neural network for processing.

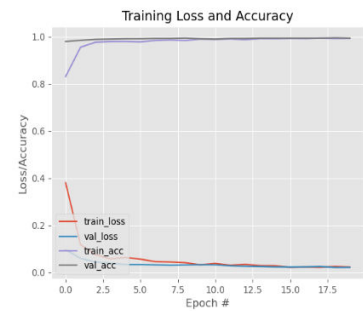
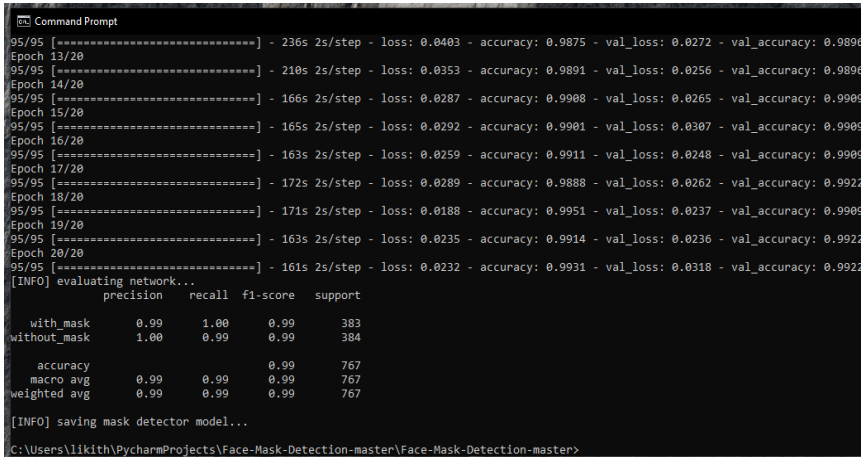
This step is to use the vector we obtained above as the input for the neural network by using the Dense function in Keras. The first parameter is output_dim which is the number of nodes in the hidden layer. You can determine the most appropriate number through experimentation. The higher the number of dimensions the more computing resources you will need to fit the model. A common practice is to pick the number of nodes in powers of two. The second parameter is the activation function. We usually use the ReLu activation function in the hidden layer.

The next layer we have to add is the output layer. In this case, we'll use the sigmoid activation function since we expect a binary outcome. If we expected more than two outcomes, we would use the softmax function.

IV. EXPERIMENTAL RESULTS

Figures shows the results of training model. Accuracy plot of the model and face mask detection in real time.

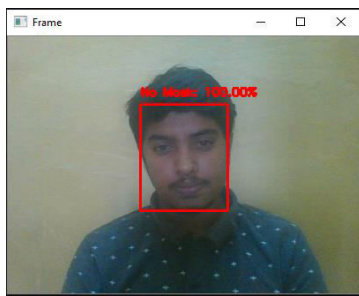
Fig.1. (a) shows the training of the model. (b) is the accuracy plot of the model. Fig.2. (a) detection of No Mask in real time, (b) Detection of Mask in real time,(c) Can distinguish hand and mask, (d) and (e) can detect multiple faces at a time in real time.



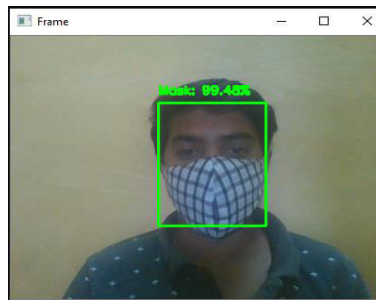
(a)

(b)

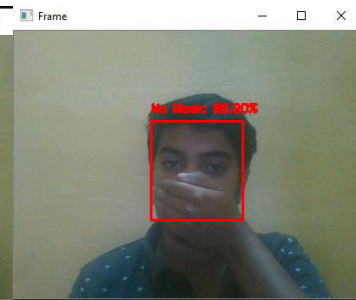
Fig. 1. (a) Training of the model, (b) Accuracy plot of the model



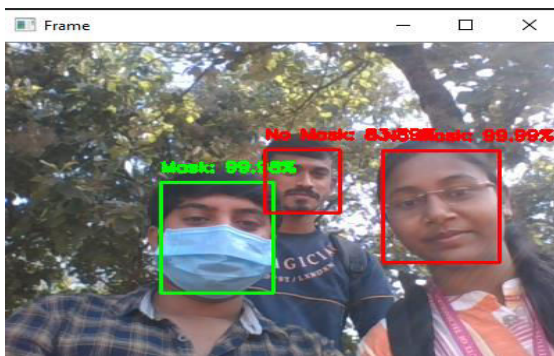
(a)



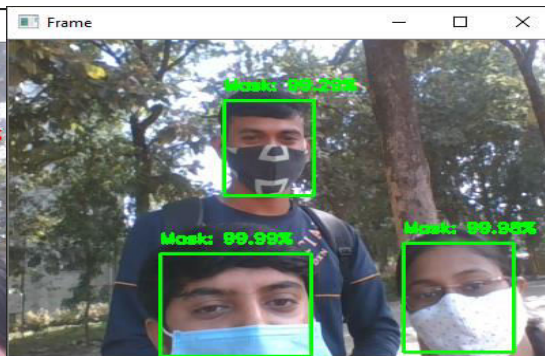
(b)



(c)



(d)



(e)

Fig. 2. (a)detecting No Mask, (b) Detecting Mask, (c) predicted No Mask for covering with Hand, (d) & (e) Multiple Face Detection.



V. CONCLUSION

As the technology are blooming with emerging trends the availability so we have novel face mask detector which can possibly contribute to public healthcare. 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. We used OpenCV, tensor flow, keras and Python to detect whether people were wearing face masks or not.

The models were tested with images and real-time video streams. 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. By the development of face mask detection, we can detect if the person is wearing a face mask and allow their entry would be of great help to the society.

REFERENCES

- [1] Face Detection in Real Time Based on HOG. N. J. Wang, S. C. Chang and P. J. Chou. Taipei, Taiwan: IEEE, DOI:10.1109/ISPACS.2012.6473506, 2012. International Symposium on Intelligent Signal Processing and Communications Systems. pp. 333-337. ISBN: 978-1-4673- 5081-5.
- [2] Face Recognition Homepage, <http://www.face-rec.org/algorithms/> [3] Seeing with OpenCV, Article, http://www.cognitics.com/opencv/servo_2007_series/part_1/index.html [4] Python with deep learning, <https://www.tutorialspoint.com> [5] Face Detection and Tracking using OpenCV. S.V.Viraktamath, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni. 3, s.l.: SIJ, July-August 2013, The Standard International Journals (The SIJ) , Vol. 1, pp. 45-50. ISSN: 2321 – 2403
- [3] Takeo Kanade. Computer recognition of human faces, volume 47. Birkh" auser Basel, 1977.
- [4] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.
- [5] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, Jan 1991.
- [6] Dong chen He and Li Wang. Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4):509–512, Jul 1990.
- [7] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In 2009 IEEE 12th International Conference on Computer Vision, pages 32–39, Sept 2009.
- [8] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, Jul 1997.
- [9] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 1, pages I–511–I–518 vol.1, 2001.
- [10] Meng Xiao He and Helen Yang. Microarray dimension reduction, 2009.



**Impact Factor:
7.569**



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details