2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N) | 978-1-6654-3811-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICAC3N53548.2021.9725546

Face Mask Detection using Keras, Opencv and Tensorflow by Implementing Mobilenetv2

[1]Sashank Machiraju, [2]Siddhaling Urolagin, [3]Ram Krishn Mishra, [4]Vishnu Sharma
[1,2,3]Department of Computer Science,
Birla Institute of Technology and Science, Pilani,
Dubai International Academic City,
Dubai, United Arab Emirates.
[2]APPCAIR, BITS-Pilani, Goa, India
[4]Galgotias College of Engineering and Technology
[1]f20180017@dubai.bits-pilani.ac.in
[2]siddhaling@dubai.bits-pilani.ac.in
[3]rkmishra@dubai.bits-pilani.ac.in
[4]vishnusharma97@gmail.com

***Abstract*—The pandemic of Corona Virus Disease (COVID-19) is triggering a public health emergency. Wearing a face mask is one of the most important ways to combat the infection. Face mask identification is discussed in the article, and it can be used by authorities to enable COVID-19 avoidance, assessment, prevention, and action plans. In this project, we developed a deep learning model to detect the presence of a mask on a person. We have trained it using Keras and OpenCV. The image classification method: MobilenetV2 is used to speed up the training time and we use approx. 2000 images to train the model.**

Keywords—Face recognition, OpenCV, MobilenetV2, Keras

I. INTRODUCTION

We, as a society, have faced a lot of travesties in the year 2020, witnessing how devastating nature can be. The covid-19 has been the most life-altering event in most of our lifetimes. From hygiene standards to hospitals and public places in general, sanitation has become the number one priority. Of these hygiene standards wearing a face mask has been the most important one. Each and every person wears a mask once they are out and authorities all over the world are designated to ensure that people are wearing masks while they are in public places. To monitor and rectify people not following the hygiene standards, a face mask detector system can be implemented. To implement this software, we have to go through two major steps. The first one is detecting a face while the second one is to detect a mask on that face. Face detection is one of the most widely used methods in modern day machine learning. The Covid-19 pandemic has made us aware of how handy Machine Learning and Data Science can be in handling tough situations.

This is a computer vision and deep learning-based facial recognition mask model. By increasing the detection of persons wearing and not wearing face masks, the suggested model may be coupled with surveillance cameras to prevent COVID-19 transmissions. Using OpenCV, tensor flow, and Keras, the model combines deep learning and traditional machine learning approaches. Deep transfer learning is utilised to extract features in this case, along with three traditional machine learning techniques. The Keras network architecture was used to create a model. The first element of this endeavour is model training, and the second is webcam testing with OpenCV.

In section II, papers related to this topic were reviewed and valuable input was taken. Section III describes the methodology used to implement the model. Section IV sets up the experiment by defining all the variables and outlining the environment and section V displays all the results obtained.

II. LITERATURE SURVEY

In recent years, a lot of academicians and observers have focused largely on grayscale face images (Ojala, Pietikainen, & Maenpaa, 2002). Although some were purely based on pattern recognition models, they have prior face model information, whilst others used AdaBoost (Kim, Park, Woo, Jeong, & Min, 2011), an excellent classifier for training purposes. So, essentially, it's not working in gloomy and poor light. As a result, researchers began exploring for a new alternative model capable of identifying faces and masks on the face. Annotations are provided for current faces rather than previous ones in these databases. Large datasets are considerably more required for better training and testing data, as well as for more easy execution of real-world applications. This necessitates a number of deep learning algorithms that can read faces and masks directly from the user's input.A breakthrough face detection technology then was developed named as Viola Jones detector that was an optimized technique of using Haar [1], digital image features used in object recognition. However, it failed because it did not perform well on faces in dark areas and non-frontal faces. Since then, researchers are eager to develop new algorithms based on deep learning to improve the models. A dataset was created by Prajna Bhandary using a PyImageSearch reader. This dataset consists of 1,376 images belonging to all races and is balanced. There are 690 images with masks and 686 without masks. Firstly, it took normal images of faces and then created a customized computer vision Python script to add face masks to them. Thereby, it created a realworld applicable artificial dataset. This method used the facial landmarks which allow them to detect the different parts of the faces such as eyes, eyebrows, nose, mouth, jawline etc. To use the facial landmarks, it takes a picture of a person who is not wearing a mask, and, then, it detects the portion of that person's face. After knowing the location of the face in the image, it extracted the face Region of Interest (ROI). After localizing facial landmarks, a picture of a mask is placed into 3 the face. In this project, embedded devices are used for deployment that could reduce the cost of manufacturing.

ISBN: 978-1-6654-3811-7/21/$31.00 ©2021 IEEE

Most of the time, the projects are focused on detecting persons who are concealed wearing masks. The other projects are focused on studying the effects of wearing a mask on the spread of covid-19. The authors of [1] then divided the people wearing a face mask into three categories. The first two categories are incorrect and no face mask-wearing. The third category is face mask-wearing. They were ready to categorise three types of people who used face masks. Face mask wear, improper face mask wear, and no face mask wear are the three categories. To recognise the person, Saber et al [2] used principal component analysis on masked and unmasked face detection. PCA has been used in [3]. The author devised a method for removing spectacles off the frontal faces of humans. The YOLOv3 method was utilized by the authors in [4] for face detection. YOLOv3 uses Darknet-53 as its backbone. Nizam et al. [5] proposed a completely new GAN-based network that would automatically remove the mask covering the face region and reconstruct the image by filling in the missing pixels. The authors of [6] proposed a technique for identifying whether a required medical mask is present or not in the operating room. The major goal is to decrease the rate of false positive face detections as much as achievable while still identifying absent masks, so that only medical workers who do not use face masks receive warnings. Shaik et al [7] used deep learning real-time facial emotion classification and identification. VGG-16 was used to classify seven different countenances. Under the present Covid-19 lock-in time, this technique is effective.

III. Methodology

We use many big-name software in this project, namely: Keras, OpenCV, TensorFlow.OpenCV (Open-Source Computer Vision Library) is a free and open-source runtime environment for computer vision and machine learning. OpenCV was designed to provide a shared infrastructure for image processing applications and to help commercial applications integrate machine perception more easily. Since OpenCV is a BSD-licensed software, it is simple for companies to use and change the code. TensorFlow is an open-source platform that enables developers to create and deploy machine learning models.

It has various levels of concepts to choose from. The framework is written in Python and is easily verifiable through various tools.

Fig. 1. Training the mask detector

Fig. 2. Application of the mask detector

We're working with a collection of pictures that come in a variety of colours, heights, and orientations. As a result, we'll have to convert all of the photos to grayscale to guarantee that colour doesn't play a role in mask recognition. Then, before adding the photos to the neural network, we must resize them all. This graphic is made up of two convolutional layers (Two Convo2D 100@3x3). After that, the convolutional architecture must be fine-tuned. There is now a blueprint. Add(Dropout(0.5)) to get rid of overfitting. When there are two types of data, we shall utilize binary cross entropy (with mask and without mask). Keep in mind that for higher accuracy, we may utilize more Convo layers or external trainers like MobileNetV2. The graphs may be plotted to help us make better validation judgments. We must first load the model that we developed. Following that, we made the selected camera the default. The two probabilities must then be labeled (0 for with mask and 1 for without mask). The bounding rectangle's color will then be changed using RGB values. We'll read frame by frame from the camera, convert them to grayscale, then detect faces in an infinite loop. And, since the training network expects 4D data, it will be run through a for loop for each face, detecting the region of interest, resizing and reshaping it to 4D. To achieve the result, we will use the best model widely accessible. If we use MobileNetV2 to fine-tune, we can boost accuracy. This model can also be used in embedded devices. We can assist assure our own and others' safety if we use it appropriately. During training, we compare the default bounding boxes with varying sizes and aspect ratios to the ground truth boxes and then use the Intersection over Union (IoU) technique to choose the best matching box for each pixel. IoU assesses how well our anticipated box matches the ground reality.

$$IoU(B_1, B_2) = \frac{B_1 \cap B_2}{B_1 \cup B_2}$$

Equation 1: Intersection over Union method

MobileNetV2 is an inverted residual system that is designed to be mobile-friendly. Its architecture consists of a convolutional layer and 19 residual bottleneck layers.A pre-existing face detection program is used to assist the model in detecting faces to apply the model.

Loss function

The net loss can be broken down into two main subsets: localization and confidence loss. Localization loss is the difference between the predicted and the actual bounding box. For a given centre (cx, cy), we try to reduce the loss by altering the dimensions of the bounding box. The equations for the above loss functions respectively are:

$$L_{loc}(x,l,g) = \sum_{i \epsilon Pos}^{N} \sum_{m \epsilon \{cx,cy,w,h\}} x_{ij}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w$$

$$\hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^w = \log \frac{g_j^w}{d_i^w} \qquad \hat{g}_j^w = \log \frac{g_j^w}{d_i^w}$$

Equation 2: Localization loss equations

$$L_{loc}(x,c) = -\sum_{i\epsilon Pos}^{N} x_{ij}^{p} \log \hat{c}_i^p - \sum_{i\epsilon Neg} \log \hat{c}_i^0 \quad where$$

$$\hat{c}_i^p = \frac{\exp\left(c_i^p\right)}{\sum_p \exp\left(c_i^p\right)}$$

Equation 3: Confidence loss equation

Notations:

- g: ground truth bounding box
- l: predicted
- d: default bounding box
- x^{p}_{ij} : matching i^{th} predicted box with j^{th} default with "p" category
- cx, cy: total distance from the centre of the box on the x and y axis
- w, h: width and height of the image
- c: confidence

IV. Experimental setup & Database

The dataset for the program is obtained from Kaggle.com. It contains 100 photos of people' faces with mask and 1000 photos of people's faces without mask. This dataset is used to train the model to appropriately detect masks on faces. Supervised algorithms are used in the model to train. The model is built by importing Keras and TensorFlow libraries in Python language. Sklearn is used to implement machine learning methods to the model. One-hot encoding is implemented so that data can be displayed more expressively by the model. Many ML algorithms struggle to work with categorical data directly. The data musty be converted to numbers for the model to interact with it better, which is the role of on-hot encoding here.

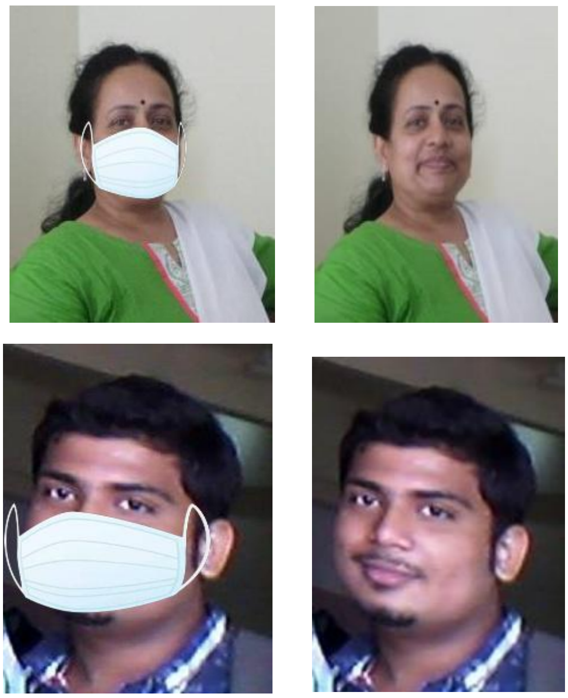

Fig. 3. Sample images from the dataset (with_mask/withoutmask)

V. Experimental Results

First, we check the model's accuracy. This can be done by typing the following command in the console.

Training the model takes a lot of time. For our use-case, it took about 2 hours to process both with_mask and without_mask images.

TABLE I. Epoch, Loss, Accuracy, Val_Loss and Val_acc values as obtained from training the model

Epoch	Loss	Accuracy	Val_loss	Val_acc
1/20	0.6195	0.8921	0.4811	0.9912
2/20	0.3457	1.0651	0.3660	1.041
3/20	0.2907	1.0955	0.3709	1.0379
4/20	0.267	1.0856	0.2300	1.1034
5/20	0.2126	1.1212	0.2873	1.0706
6/20	0.1981	1.1258	0.2064	1.1096
7/20	0.186	1.1303	0.3234	1.0706
8/20	0.1555	1.1449	0.3317	1.0706
9/20	0.1812	1.1347	0.3871	1.0535
10/20	0.1635	1.1396	0.3127	1.0769
11/20	0.1416	1.1499	0.2568	1.0878
12/20	0.1445	1.1515	0.4256	1.0519
13/20	0.1278	1.1558	0.2150	1.1034
14/20	0.1427	1.1472	0.4577	1.0472
15/20	0.1543	1.1429	0.3725	1.0597
16/20	0.1297	1.1546	0.3282	1.0738
17/20	0.1289	1.1484	0.2522	1.0972
18/20	0.1301	1.1491	0.3094	1.0769
19/20	0.1282	1.1512	0.2614	1.0940
20/20	0.1098	1.1622	0.3002	1.0862

TABLE I indicates that starting with the second epoch, accuracy increases and error reduces. The table may then be viewed in the graph in Fig. 2. (given below).

TABLE II. PRECISION, RECALL, F-SCORE AND SUPPORT OF DIFFERENT USE-CASES

	Precision	Recall	F- Score	Support
With mask	0.97	0.84	0.90	386
Without mask	0.83	0.9	0.93	390
Accuracy			0.91	775
Macro avg	0.93	0.90	0.91	775
Weighted avg	0.93	0.91	0.91	775

When the accuracy line remains constant, it assures that no more iteration is required to increase the model's accuracy. In TABLE II, we observe the accuracy of the model to be fairly good. The model predicts photos with masks about 98% of the time and mask less photos 85% of the time. The high F-score (0.90-0.91) indicates a near perfect precision and recall. Training loss and accuracy graph is given below.

Fig. 4. Training loss and Accuracy graph of the model

The expression Epoch is used in machine learning to describe the number of runs the machine learning algorithm has made over the entire training dataset. Batches are often used to group data sets (especially when the amount of data is very large). We have mentioned 20 Epochs in the program code. From this graph, we can infer that the model's accuracy increases as the epochs increase.

The training loss gradually dips and the curve gradually flattens which means that the model built is good. When the program is run in the console, all the dependencies are loaded and the program initializes the camera of the device.

The same output is given below:

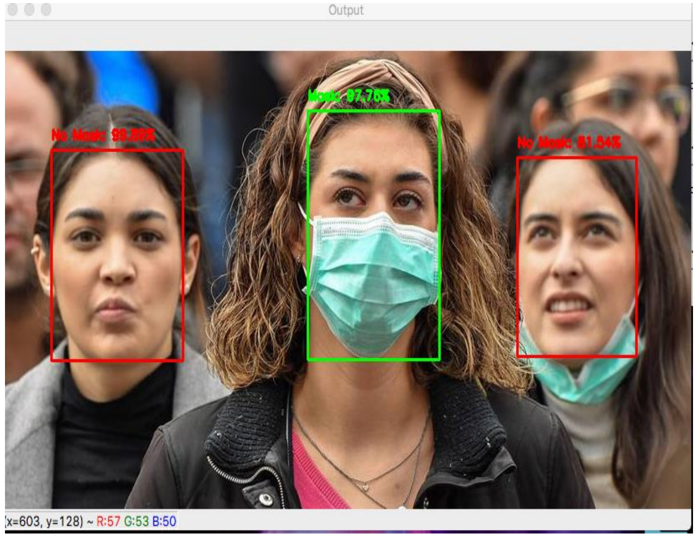

Fig. 5. Sample input and output

VI. CONCLUSION

Finally, this research introduces a face mask recognition model based on machine learning. The model will properly estimate the number of individuals using face masks in certain cities after the training, validation, and testing phases. The training of the model was done using a database of over 1000 images. The training was stopped once the value of Val_acc stopped increasing. Failing to do so would have resulted in a model which would have been overfit.

This research will be a simple leap for authorities to use more unstructured data tools for more data-based mitigation, monitoring, detection, and action plans against COVID-19 in the name of the statistical agency that needs to move rapidly to implement and take advantage of machine learning and new modern data resources.

REFERENCES

[1] B. QIN and D. Li, Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19, May 2020, doi: 10.21203/rs.3.rs28668/v1.

[2] M.S. Ejaz, M.R. Islam, M. Sifatullah, A. Sarker Implementation of principal component analysis on masked and non-masked face recognition 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (2019),
pp. 15, 10.1109/ICASERT.2019.8934543 International Journal of Computer Engineering and Applications, Volume XIV, Issue X, OCT 2020, www.ijcea.com ISSN 2321-346922 AMRIT KUMAR BHADANI ANURAG SINHA

[3] Jeong-Seon Park, You Hwa Oh, Sang ChulAhn, and Seong-Whan Lee, Glasses removal from facial image using recursive error compensation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 805–811, doi: 10.1109/TPAMI.2005.103.

[4] C. Li, R. Wang, J. Li, L. Fei, Face detection based on YOLOv3, in:: Recent Trends in Intelligent Computing, Communication and Devices, Singapore, 2020, pp. 277–284, doi:10.1007/978-981-13-9406-5_34.

[5] N. Ud Din, K. Javed, S. Bae, J. Yi A novel GAN-based network for unmasking of masked face IEEE Access, 8 (2020), pp. 4427644287, 10.1109/ACCESS.2020.2977386

[6] A. Nieto-Rodríguez, M. Mucientes, V.M. BreaSystem for medical mask detection in the operating room through facial attributes Pattern Recogn. Image Anal. Cham (2015), pp. 138-145, 10.1007/978-3-319-19390-8_16

[7] S. A. Hussain, A.S.A.A. Balushi, A real time face emotion classification and recognition using deep learning model, J. Phys.: Conf. Ser. 1432 (2020) 012087, doi: 10.1088/1742-6596/1432/1/012087.

[8] Batagelj, B.; Peer, P.; Štruc, V.; Dobrišek, S. How to Correctly Detect Face-Masks for COVID-19 from Visual Information?. Appl. Sci. 2021, 11, 2070. https:// doi.org/10.3390/app11052070

[9] S. Sakshi, A. K. Gupta, S. Singh Yadav and U. Kumar, "Face Mask Detection System using CNN," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 212-216, doi: 10.1109/ICACITE51222.2021.9404731.

[10] Das, Arjya & Ansari, Mohammad & Basak, Rohini. (2020). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. 10.13140/RG.2.2.21473.94565.

[11] Balaji, S. and Balamurugan, B. and Kumar, T. Ananth and Rajmohan, R. and Kumar, P. Praveen, A Brief Survey on AI Based Face Mask Detection System for Public Places (March 28, 2021). Irish Interdisciplinary Journal of Science & Research (IIJSR) 2021, Available at SSRN: https://ssrn.com/abstract=3814341

[12] B. Heiselet, T. Serre, M. Pontil and T. Poggio, "Component-based face detection," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, 2001, pp. I-I, doi: 10.1109/CVPR.2001.990537.