As per UGC guidelines an electronic bar code is provided to seure your paper

International Journal for Modern Trends in Science and Technology, 8(05):392-398, 2022
Copyright © 2022 International Journal for Modern Trends in Science and Technology
ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0805058
Available online at: http://www.ijmtst.com/vol8issue05.html

Detection of Face Mask using Deep Learning

P V Kumar[1] | Dhyavade Manoj Pradeep[2] | Rapaka Shashi Mohan[2] | Garlapati Sai Manikanta[2]

[1]Professor, Department of Information Technology, Anurag Group of Institutions.
[21]Department of Information Technology, Anurag Group of Institutions.
Corresponding Author Email ID:pvkumarit@cvsr.ac.in, 18h61a1275@cvsr.ac.in, 18h61a12B0@cvsr.ac.in, 18h61a1280@cvsr.ac.in

To Cite this Article
P V Kumar, Dhyavade Manoj Pradeep, Rapaka Shashi Mohan and Garlapati Sai Manikanta. Detection of Face Mask using Deep Learning. International Journal for Modern Trends in Science and Technology 2022, 8(05), pp. 392-398. https://doi.org/10.46501/IJMTST0805058

Article Info
Received: 16 April 2022; Accepted: 15 May 2022; Published: 19 May 2022.

ABSTRACT

Since we are amidst the pandemic and it is not very long since the first wave left us devastated and second wave is right here, it becomes our prime objective to contribute in this scenario as per our capability. There are certain precautions that one must take to avoid unwanted infections. Face mask and social distancing are two of the main precautions. Since most of the people are new with the concept of face masks, they are being irresponsible about wearing mask. As our country starts going through various stages of reopening after the COVID-19 pandemic, World Health Organization (WHO) has declared the use of a face mask as a mandatory biosafety measure.

1. INTRODUCTION

Face mask detection refers to detect whether a person is wearing a mask or not and also whether the person is wearing mask in a proper way or not. In fact, the problem is reverse engineering of face detection where the face is detected using different machine learning algorithms for the purpose of security, authentication and surveillance. Face detection is a key area in the field of Computer Vision and Pattern Recognition. A significant body of research has contributed sophisticated to algorithms for face detection in past. The primary research on face detection was done in 2001 using the design of handcraft feature and application of traditional machine learning algorithms to train effective classifiers for detection and recognition. The problems encountered with this approach include high complexity in feature design and low detection accuracy. In recent years, face detection methods based on deep convolutional neural networks (CNN) have been widely developed to improve detection performance.

Although numerous researchers have committed efforts in designing efficient algorithms for face detection and recognition but there exists an essential difference between ‘detection of the face under mask’ and ‘detection of mask over face’. As per available literature, very little body of research is attempted to detect mask over face. Thus, our work aims to develop a technique that can accurately detect masks over the face in public areas (such as airports. Railway stations, crowded markets, bus stops, etc.) to curtail the spread of Coronavirus and thereby contributing to public healthcare. Further, it is not easy to detect faces with/without a mask in public as the dataset available for detecting masks on human faces is relatively small leading to the hard training of the model. So, the concept of transfer learning is used here to transfer the learned kernels from networks trained for a similar face detection task on an extensive dataset. The dataset covers various face images including faces with masks, faces without masks, faces with and without masks in one image and confusing images without masks. With an extensive dataset containing 3800 images, our technique achieves outstanding accuracy of 98.2%.

1.1 Overview

An end-to-end video-based Face Mask Recognition model was designed using Tensorflow, OpenCV that detects faces in an image and recognizes whether the person is wearing a Face-Mask or not. We used the pre-trained Caffe Model provided in the dnn module in OpenCV for Face Detection and MobileNet V2 CNN model with some modifications for training. We achieved a 99.60% accuracy on the training set and a 99.20% accuracy on the test set.

The prime objective of training this particular model was the current impact of COVID- 19 on public health and the importance of the usage of Face-Masks in times of come.

1.2 Problem Statement:

The development of the face detection model helps to detect the face of individuals and conclude whether they are wearing masks or not at that particular moment when they are captured in the image.

2. LITERATURE SURVEY:

The proposed model can be incorporated with observation cameras to block the COVID- 19 transmission by permitting the discovery of individuals who are wearing veils not wearing face covers. The model is integration between deep learning and classical machine learning techniques with OpenCV, TensorFlow, and Keras. We have used deep transfer learning for feature extractions and combined it with three classical machine learning algorithms. We conducted an examination between them to locate the most appropriate calculation that accomplished the most noteworthy exactness and burned-through minimal time during the time spent preparing and identification.

Machine Learning:

ML is the study of computer algorithms that improve automatically through experience. It is seen as a subset of AI. AI calculations construct a numerical model dependent on example information, known as &preparing information&, to settle on expectations or choices without being expressly modified to do as such. AI calculations are utilized in a wide assortment of utilizations, for example, email sifting and PC vision, where it is troublesome or infeasible to create traditional calculations to perform the needed tasks.

Computer Vision:

PC vision is an interdisciplinary logical field that manages how PCs can acquire undeniable level comprehension from computerized pictures or recordings. From the viewpoint of designing, it tries to comprehend and computerize assignments that the human visual framework can do, Computer vision errands incorporate strategies for securing, handling, examining, and understanding advanced pictures, and extraction of high dimensional information from this present reality to create mathematical or emblematic data.

Deep learning:

Profound learning techniques target taking in component pecking orders with highlights from more elevated levels of the progression framed by the structure of lower-level highlights. Consequently learning highlights at various degrees of reflection permits a framework to learn complex capacities planning the contribution to the yield straightforwardly from information, without relying totally upon human-created highlights. Profound learning calculations look to misuse the obscure design in the information dissemination to find great portrayals, regularly at numerous levels, with more elevated level learned highlights characterized as far as lower-level highlights.

Opencv:

Opencv (Open Source Computer Vision Library) is an opensource computer vision and machine learning software library. OpenCV was designed to give a typical foundation to PC vision applications and to quicken the utilization of machine discernment in business items. The library has in excess of 2500 streamlined calculations, which incorporates an extensive arrangement of both works of art and cutting edge PC vision and AI calculations. These calculations can be utilized to distinguish and perceive faces, recognize objects, group human activities in recordings, track camera developments, track moving articles, separate 3D models of items, produce 3D point mists from sound system cameras, line pictures together to create a high-goal picture of a whole scene, find comparable pictures from a picture data set, eliminate red eyes from pictures taken utilizing streak, follow eye developments, perceive view and build up markers to overlay it with enlarged reality, and so on.

TensorFlow:

It is a free and open-source programming library for dataflow and differentiable programming across a degree of assignments. It is a representative mathematical library and is likewise utilized for AI applications, for example, neural organizations. It is utilized for both examination and creation at Google, TensorFlow is Google Brain&#39;s second-age framework. Keras is an API intended for people, not machines. Keras follows best practices for decreasing intellectual burden: it offers steady and straightforward APIs, it limits the number of client activities needed for basic use cases, and it gives clear and significant error messages.

Keras:

While deep neural networks are all the rage, it has been a barrier to their use for developers new to machine learning. There have been several proposals for improved and simplified high-level APIs for building neural network models, all of which tend to look similar from a distance but show differences in closer examination. Keras (is one of the high level neural networks APIs) a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

3. PROBLEM STATEMENT

Face detection problem has been approached using Multi-Task Cascaded Convolutional Neural Network (MTCNN). Then facial features extraction is performed using the Google Face Net embedding model. This system is capable to train the dataset of both persons wearing masks and without wearing masks

After training the model the system can predicting whether the person is wearing the mask or not wearing mask. Mtcnn (multi-task cascaded convolutional neural networks) is a calculation comprising of 3 phases, which distinguishes the bounding boxes of appearances in a picture alongside their 5 point face landmarks. Each stage gradually improves the detection results by passing its inputs through a cnn, which returns candidate bounding boxes with their scores, followed by non max suppression. In stage 1 the information picture is downsized on numerous occasions to fabricate a picture pyramid and each scaled form of the picture is gone through its cnn. In stages 2 and 3 we extract image patches for each bounding box and resize them (24x24 in stage 2 and 48x48 in stage 3) and forward them through the cnn of that stage. Other than bounding boxes and scores, stage 3 moreover figures 5 face milestones focuses for each bounding box.

4.PROPOSED SYSTEM

With time, deep cnn have become important tools for many computer vision related tasks like classification of images. The face mask recognition system in this study is developed using a machine learning algorithm through the image classification method: mobilenetv2. Mobilenetv2 is a method based on cnn that developed by google with improved performance and enhancement to be more efficient and faster than cnn. Data scientists have derived an inception network to achieve better accuracy for image classification and segmentation. In general, convolution neural networks (cnn) with large filters tend to have high computational cost. One prominent solution to reduce this cost is inception modules. Inception reduces the cost by finding optimal local sparse structures. The idea of the inception block is to design a layer by layer construction with the analysis of layer correlation statistics. The clusters of highly correlated layers are used to form groups of units. Each unit from an earlier layer corresponding to some region of the input image is referred to as a filter bank. This process ends with the concatenation of huge filter banks from a single region.

Mobilenetv2

Mobilenetv2 builds upon the ideas from mobilenetv1, using depth wise separable convolution as efficient building blocks. In any case, v2 acquaints two new highlights with the design:

1. Linear bottlenecks between the layers, and
2. Shortcut connections between the bottlenecks.
3. Typical mobilenetv2 architecture has as many layers,

The weights of each layer in the model are predefined based on the imagenet dataset. The loads show the cushioning, steps, portion size, input channels, and yield channels. Mobilenetv2 was picked as a calculation to construct a model that could be sent on a cell phone. A tweaked completely associated layer which contains four consecutive layers on top of the mobilenetv2 model was created. The Layers are

1. Average pooling layer with 77 weights
2. Linear layer with relu activation function
3. Dropout layer
4. Straight layer with softmax initiation works with the after effect of 2 qualities. The last layer of softmax work gives the aftereffect of two probabilities: everyone addresses the characterization of &cover& or&not veil&.

5. IMPLEMENTATION

This study conducted its experiments on two original datasets. The first dataset was taken from the kaggle dataset and the real-world masked face dataset (rmfd); used for the training, validation, and testing phase so the model can be implemented to the dataset. The model can be produced by following some steps which are:

(1) Data collection
(2) Pre-processing
(3) split the data
(4) Building the model
(5) Testing the model
(6) Implement the model.

5.1 Data Collection:

The development of the Face Mask Recognition model begins with collecting the data. The dataset trains data on people who use masks and who do not. The model will differentiate between people wearing masks and not. For building the model, this study uses 1.916 data with mask and 1.930 data without a mask. At this step, the image is cropped until the only visible object is the face of the object. The next step is to label the data. The data which has been collected is labeled into two groups; with and without a mask. After the data has been labeled, it is grouped into those two groups.

5.2 Data preprocessing:

The dataset used [kaggle] contains a total of 8226 images out of which 3972 images are people's faces without a mask and 4254, with a mask each. 75% of the images from the dataset are used for training and 25% of the images are used for testing the model. The images in our dataset are preprocessed as follows, before being fed into the mobilenetv2.

•Resize the input images and center-crop the image with the pixel value of 224 x 224 x 3 via augmentation.

•Apply color filtering (RGB) over the channels (our model MobileNetV2 supports 2- dimensional three-channel images).

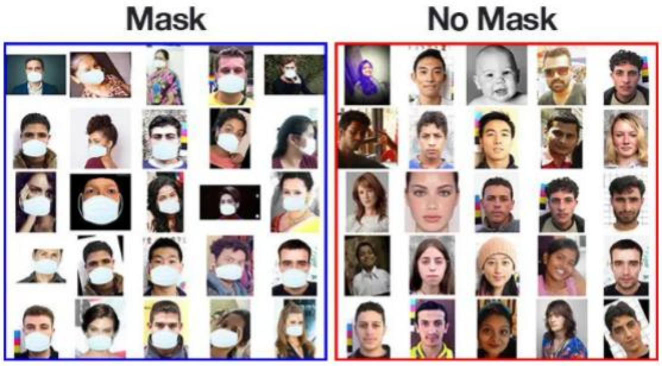
• Scale/Normalize images using ImageDataGenerator of the Keras library.

•Downsample the images to a fixed resolution of 256*256 by extracting random 224*224 patches from 256*256 images.

•Finally, converting them into tensors, similar to NumPy arrays. Our goal is to take a new image that falls into a category we have trained and run it through a command that will tell us the category in which the image fits--"mask" or "no mask" category. And the last step in this phase is performing hot encoding on labels because many machine learning algorithms cannot operate on data labeling directly. They require all input variables and output variables to be numeric, including this algorithm. The labeled data will be transformed into a numerical label, so the algorithm can understand and process the data.

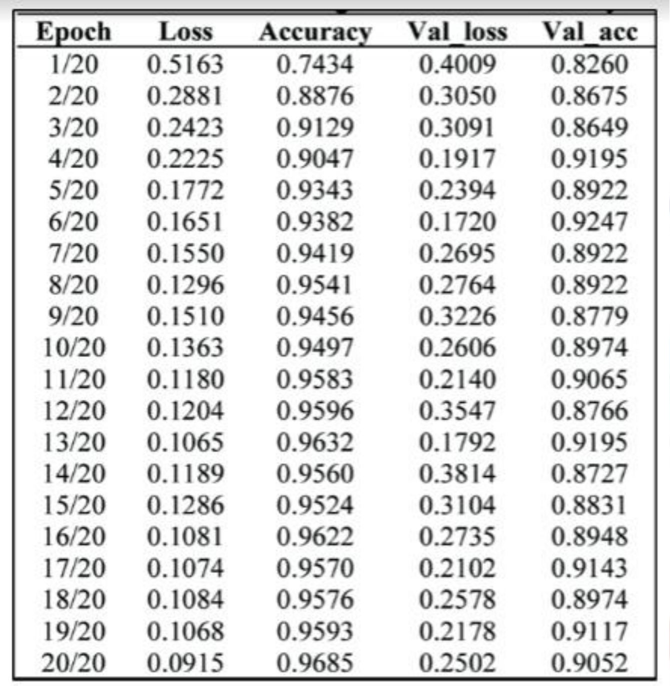
5.3 Split the Data:

After the preprocessing phase, the data is split into two batches, which are training data namely 75 percent, and the rest is testing data. Each batch is containing both with-mask and without-mask images.



5.4 Building the Model

The next phase is building the model. There are six steps in building the model which are constructing the training image generator for augmentation, the base model with MobileNetV2, adding model parameters, compiling the model, training the model, and the last is saving the model for the future prediction process. To make sure the model can predict well, there are steps in testing the model. The first step is making predictions on the testing set. The result for 20 iterations in checking the loss and accuracy when training the model is shown in Table Below.



Epoch	Loss	Accuracy	Val_loss	Val_acc
1/20	0.5163	0.7434	0.4009	0.8260
2/20	0.2881	0.8876	0.3050	0.8675
3/20	0.2423	0.9129	0.3091	0.8649
4/20	0.2225	0.9047	0.1917	0.9195
5/20	0.1772	0.9343	0.2394	0.8922
6/20	0.1651	0.9382	0.1720	0.9247
7/20	0.1550	0.9419	0.2695	0.8922
8/20	0.1296	0.9541	0.2764	0.8922
9/20	0.1510	0.9456	0.3226	0.8779
10/20	0.1363	0.9497	0.2606	0.8974
11/20	0.1180	0.9583	0.2140	0.9065
12/20	0.1204	0.9596	0.3547	0.8766
13/20	0.1065	0.9632	0.1792	0.9195
14/20	0.1189	0.9560	0.3814	0.8727
15/20	0.1286	0.9524	0.3104	0.8831
16/20	0.1081	0.9622	0.2735	0.8948
17/20	0.1074	0.9570	0.2102	0.9143
18/20	0.1084	0.9576	0.2578	0.8974
19/20	0.1068	0.9593	0.2178	0.9117
20/20	0.0915	0.9685	0.2502	0.9052

Tabel 1

From Table 1, we can see that the accuracy is increasing at the start of the second epoch, and loss is decreasing after it. The table then can be shown in the graph shown in Figure 2. When the accuracy line is being stable, it means that there is no need for more iteration for increasing the accuracy of the model. So then, the next step is making the model evaluation as shown in Table 2.

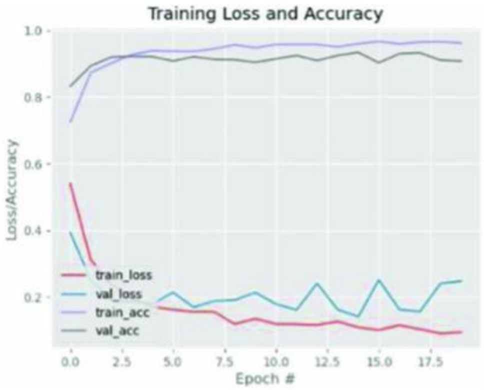


Figure9 Plot diagram

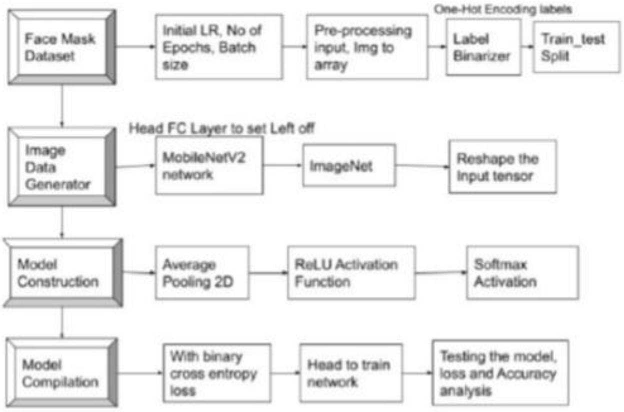
5.5 Implementing the model:

The model implemented in the video. The video reads from frame to frame, then the face detection algorithm works. If a face is detected, it proceeds to the next process. From detected frames containing faces, reprocessing will be carried out including resizing the image size, converting to the array, pre-processing input using MobileNetV2. The next step is predicting input data from the saved model. Predict the input image that has been processed using a previously built model. Besides, the video frame will also be labeled that the person is wearing a mask or not along with the predictive percentage. For Mask Recognition, following are the details of the model: MobileNet V2 along with some incremental modifications:

- Max-Pool of 7 x 7
- Fully connected layer of 128 neurons. Followed by Dropout of 0.5
- Last layer maps to the 2 classes with softmax activation.

Trained with a learning rate of 0.01,Batch Size of 32 and with 20 to 40 epochs. Used Adam optimizer with decay as the division of learning rate by the number of epochs Used OpenCV for creating montages of the output and creating the blob for Face Detection. Used Tensorflow&#39;s Image Data Generator feature for Data Augmentation for better results.

Phase 2:



5.6 Image data generator:

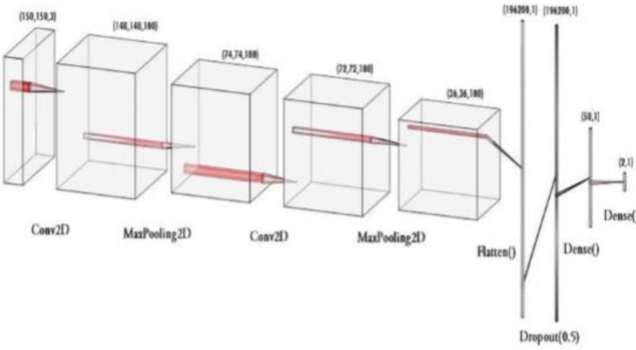
Before start building the model, there is a small step which is making the every image of dataset to be understood by the model in many different ways using the image data generator. Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples. Image data augmentation is perhaps the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image. Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more.

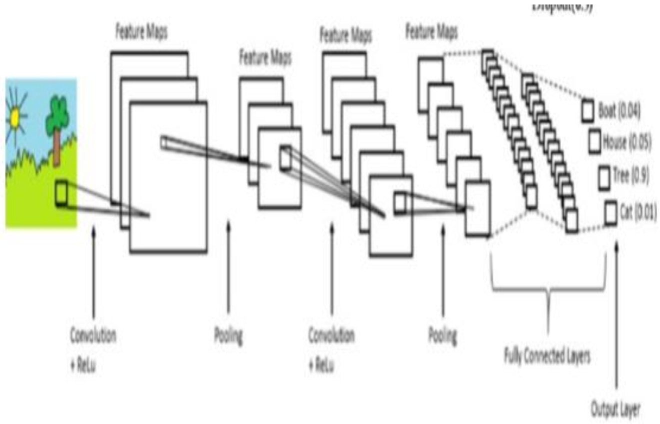
5.7 Label binarizer:

The transform method in the label binarizer makes this operation simple. When it comes to prediction, the matching model provides the highest level of performance by assigning the class. The inverse transform method in label binarizer makes this simple. Negative labels must be encoded with this value.

5.8 Relu activation:

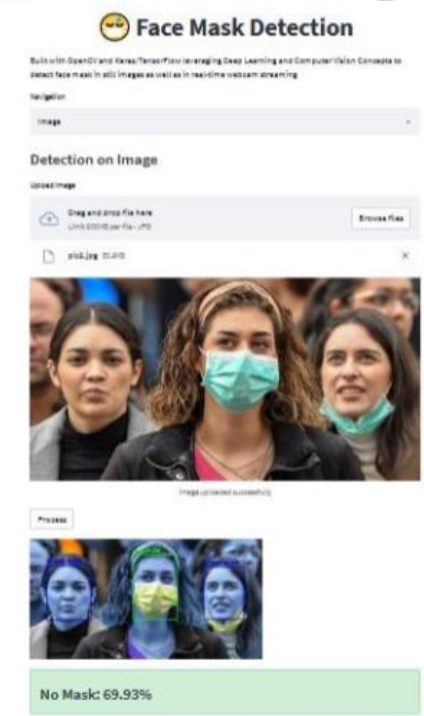
For short, relu is a piece-wise linear transformation that outputs the input directly if it is positive; else, it outputs zero. It became the standard activation function for several neural network models since it is willing to implement and typically results in high quality. A rectified linear activation unit, or relu for short, is a cluster or component that performs the activation function. Resolved networks are networks that employ the rectifier function for their hidden neurons. Acceptance of relu is readily regarded as one of the few watershed moments in the deep learning renaissance, alongside approaches that today allow for the everyday construction of incredibly deep learning techniques. The functions involved in the relu activation function are computational simplicity, linear behavior, representational sparsity, and train deep networks. By using Transfer Learning I am making use of the feature detection capabilities of the pre-trained MobileNetV2 and applying it to our rather simple model. The MobileNetV2 is followed by our DNN composed of GlobalAveragePooling, Dense and Dropout layers. As ours is a binary classification problem, the final layer has 2 neurons and softmax activation.





6. RESULTS

6.1 Using uploading images:



Face Mask Detection

Built with OpenCV and Keras/TensorFlow leveraging Deep Learning and Computer Vision Concepts to detect face mask in still images as well as in real-time webcam streaming

Navigation

Image

Detection on Image

Upload Image

Drag and drop file here
Limit 200MB per file • JPG

Browse files

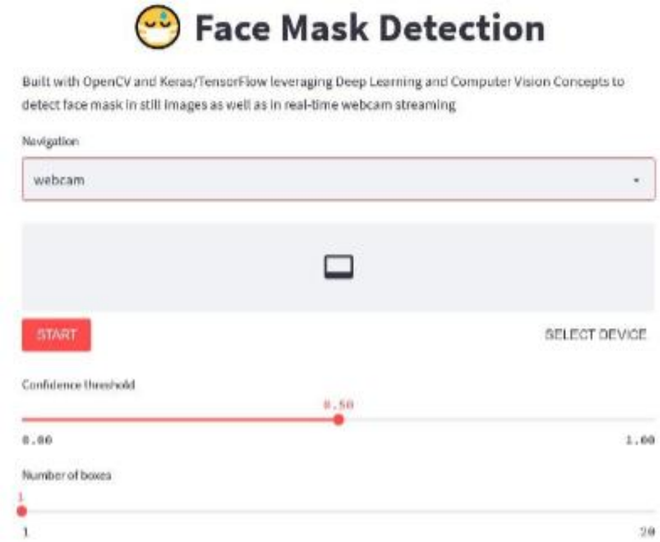
sick.jpg

Image uploaded successfully

Process

No Mask: 69.93%

using live web cam:



Face Mask Detection

Built with OpenCV and Keras/TensorFlow leveraging Deep Learning and Computer Vision Concepts to detect face mask in still images as well as in real-time webcam streaming

Navigation

webcam

START

SELECT DEVICE

Confidence threshold
0.50
0.00
1.00

Number of boxes
1
1
20

7. CONCLUSION

As the technology is booming with emerging trends therefore the novel face mask detector which can possibly contribute to public healthcare. The model is trained on an authentic dataset. We used OpenCV, tensor flow, keras and CNN to detect whether people were wearing face masks or not. The models were tested with images and real-time video. The accuracy of the model is achieved and the optimization of the model is a continuous process and we are building an accurate solution by tuning the hyper parameters. This specific model could be used as a use case for edge analytics. We are able to capture images in an efficient manner and are able to detect face masks successfully. With the help of developing this system, we can detect if the person is wearing a face mask and allow their entry would be of great help to the society.

8. FUTURE SCOPE:

This system can be implemented as mobile applications in the near future and also can be developed as an individual application program interface i.e API which can become quite crucial in our day to day lives.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] Zheng Jun, Hua Jizhao, Tang Zhenglan, Wang Feng “Face detection based on LBP&,2017 IEEE 13th International Conference on Electronic Measurement &amp; Instruments.

[2] Q. B. Sun, W. M. Huang, and J. K. Wu &Face DetectionBased on Color and Local Symmetry Information”, National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore.

[3] Based on Color and Local Symmetry Information”, National University of Singapore Heng Mui Keng Terrace, Kent Ridge Singapore.

[4] Wang Yang, Zheng Jiachun &Real-time face detection based on YOLO&,1st IEEE International Conference on Knowledge Innovation and Invention 2018.

[5] Dr. P. Shanmugavadivu, Ashish Kumar, &Rapid Face Detection and Annotation with Loosely Face Geometry&,2016 2nd International Conference on Contemporary Computing and Informatics (ic3i).

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor, &Active appearance models,& IEEE Transactions on pattern analysis and machine intelligence.