International Journal of Modern Agriculture
ISSN: 2305-7246
Volume 10 Issue 2, 2021

Real Time Face Mask Detection and Alert System Using Convolutional Neural Network

Tejaswini N
Electronics and Communication Engineering
Reva University
Bangalore, India
Email : teju2851999@gmail.com

R Pooja Sri
Electronics and Communication Engineering
Reva University
Bangalore, India
Email : rpoojasri2707@gmail.com

Neha G M
Electronics and Communication Engineering
Reva University
Bangalore, India
Email : 1901nehagm@gmail.com

Rajeshwar Reddy
Electronics and CommunicationEngineering
RevaUniversity
Bangalore,India
Email : rajeswarpathakota@gmail.com

AllaSrija
Electronics and communication Engineering RevaUniversity
Bangalore,India
Email : allasrija@reva.edu.in

1. ABSTRACT

We all know how Covid-19 pandemic has affected our lives. During this time of constant fear and uncertainty it is important that all individuals wear a protective face mask to prevent ourselves from the exposure to the deadly virus. Therefore, face mask detection has become a crucial task to avoid spreading of virus and to help the global society. This paper presents a real time face mask detection system which detects if a person is wearing a face mask or not and gives an alert to the authorities if he/she fails to wear a face mask, which can be implemented in places where public gather frequently such as entrance and exit gates of shopping malls, offices, hotels, hospitals, railway stations and other public places. The methodology that we haveimplemented involves collection of data set which includes two kinds of data, one is jpeg images of faces with mask and another jpeg images of faces without mask, preprocessing the collected dataset, training the processed data using Convolutional Neural Network (CNN) model and also by using some Machine learning packages like KERAS, Tensor flow, Open CV and Scikit-learn. Finally, a real time detection is done and an alert message is sent to the authorities via e-mail if a person is detected not wearing a face mask.

Keywords: Convolutional Neural Network, face mask detection, Keras

2. INTRODUCTION

According to the World Health Organization (WHO)'s official Situation Report there were 20 million people infected and 0.7 million deaths due to the coronavirus disease 2019(COVID-19). As of June 10, 2020 the corona virus reached nearly 8 million infected patients. Individuals with COVID-19 have more than one symptoms like going from mellow manifestations to serious illness, Respiratory problems like shortness of breath and difficulty in breathing.

people above 30 having lung disease can possess serious complications from COVID-19 illness as they have at higher risk .important and much needed points of interest of the utilization of masks lie in reducing vulnerability of risk from a one individual to another and also during the "pre-symptomatic" period and stigmatization of discrete persons putting on masks is necessary to restraint the spread of virus. To combat it's transmission and take control over the spread of virus, there are enforced protocols set by the WHO like compulsory wearing of face mask, observing strict social distancing, sanitizing hands with disinfectants frequently and so on. Wearing masks is an inexpensive way to mitigate fatalities and respiratory disorders. But the efficacy of facemasks in preventing disease transmission in the public has generally been lessened due to inadequate facemask use.

various Studies show that wearing a facemask is important to prevent the spread of the virus and the effectiveness of N95 and surgical masks are 91% and 68% respectively. Therefore, It is essential to develop an automatic detection system for wearing facemask which will provide individual protection and prevent local epidemics. Therefore, face mask detection has become a crucial task in our present global society.

Features of Face mask detection involves detecting the location of the face and then determining whether it has a mask on it or not. The issue here is proximately cognate to general object detection, that is to detect the classes of objects. Face identification here specifically deals with distinguishing a specific group of entities i.e. Face. In this paper we have used basic Machine Learning (ML) packages such as TensorFlow, Keras, OpenCV and Scikit-Learn. It has numerous applications, such as autonomous driving, education, surveillance, and so on.

3. PACKAGES IMPORTED*:*

3.1 Numpy: It is a library for the Python programming language. It supports operation on large, multi-dimensional arrays and matrices.The library contains a large number of mathematical, algebraic and transformation functions.

3.2 TensorFlow: TensorFlow is an end-to-end open source platform for machine learning which has a comprehensive, flexible ecosystem of tools, libraries and community resources. It is used for implementing ML systems into fabrication over a bunch of areas of computer science, which includes sentiment analysis, voice recognition, geographic information extraction, computer vision, etc.

3.3 Keras: Keras is a deep learning API written using Python programming language. It uses TensorFlow in the backened hence makes complete use of the scalability and cross-platform capabilities of TensorFlow. The core data structures of Keras are layers and models . Using keras, the layers of sequential CNN model in the model are implemented. It also helps in the conversion of the class vector to the binary class matrix in data processing, and to compile the overall model.

3.4 O**pen Cv**: OpenCV (Open-Source Computer Vision Library), is an opensource computer vision and ML software library, which is used to differentiate and recognize faces, recognize objects, group movements in recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, perceive landscape and set up markers to overlay it with increased reality and so forth . In the proposed method, open cv is also used Grey-scaling and resizing of images.

3.5 Matplotlib: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It provides an object-orientedAPI for embedding plots into applications using general-purpose GUI toolkits like Tkinter. In the proposed method. Pyplot, a matplotlib module, which provides a MATLAB-like interface is imported in the proposed method. Matplotlib is designed to be as usable as MATLAB, with the ability to use Python, and the advantage of being free and open-source.

3.6 Smptlib: Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers. Python provides **smtplib** module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

4. METHODOLOGY:

The proposed methodology consists of collection of data, pre-processing of data, training a CNN model consisting of two 2D convolution layers connected to layers of dense neuron, real-time detection of face mask using the trained model and an alert system(as shown in fig:1)

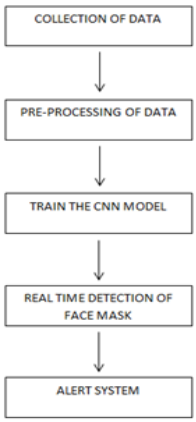

Fig.1 methodology

4.1 Collecting data set:

Collected two kinds of dataset one with faces without mask(shown in fig:2) and another with faces with mask(shown in fig:3)

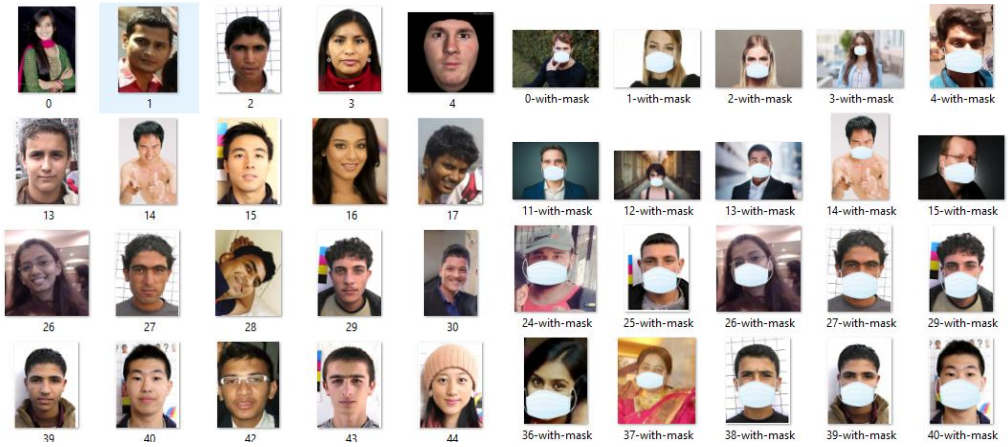

Fig.2:Without Mask **Fig.3:With Mask**

4.2 Pre-Processing the data:

Data pre-processing is done to extract only the useful information required for our application and also converts the data from a given format to much more convenient format.

Steps in data pre-processing include:

4.2.1 Visualization of data:

data visualization is the process of representing the data in the form of arrays, graph or other visual format. It tells the relationships of the data with the images.The total number of images in the dataset is visualized in both categories – 'with mask' and 'without mask'.

4.2.2 Conversion of RGB image to Gray image:

this step involves conversion of color image-to-grayscale image, as color dose not play a important role in our application. Introducing nonessential information could increase the size of training data required to achieve good performance. As grayscale rationalizes the algorithm and diminishes the computational requisites, it is utilized for extracting descriptors instead of working on color images instantaneously.

4.2.3 Image Reshaping:

Inorder to bring uniformity among all the images, reshaping of image is done. Most CNNs can only accept fine-tuned images. This gives rise to several problems throughout data collection and implementation of model. However, reconfiguring the input images before implementing them into the model can help to overcome this constraint. gray-scale images are resized into 100 x 100. The images are normalized to converge the pixel range between 0 and 1. Then they are converted to 4 dimensional array. As, the final layer of the neural network has 2 outputs – with mask and without mask i.e. it has categorical representation, the data is converted to categorical labels.

4.3 Building and Training the deep learning model:

4.3.1 Building the model:

We start by instantiating the sequential model to initiate the neural network by making use of sequential class and then start adding the various CNN layers. The convolutional layer uses a feature detector or a filter to detect significant features from the image data. Feature detector is a small weight matrix that is convolved with the input array of pixel values to give a Feature map. The first convolution layer uses 200 filters of kernel size 3X3, which is the height and width of a 2D convolution window. Since it is important for the model to know the shape of input data the first layer is provided with the dimensions of the input array from index 1 given by data.shape[1:]. The Following layers can perform instinctive shape reckoning . The next layer is the activation layer where Rectified linear unit (ReLu) is used as activation function. ReLu introduces non-linearity to the linear output obtained from the convolution layer. This layer is followed by the max-pooling layer which is used to

progressively reduce the spatial size of the representation in order to reduce to parameters and the time taken for computation in the network. This layer removes noise from the data and thus helps in faster computation and reduces overfitting. It acts on each feature map independently. Two hyper-parameters of the pooling layer ,filter and stride, are set to 2 respectively. Pool size is set to 3 x 3 and the resulting output has a shape (number of rows or columns) of: shape of output = (input shape - pool size + 1) / strides), where strides has default value (1,1) . The second convolution layer has 100 filters with kernel size 3X3, followed by the ReLu and max pooling layer.

The last stage consists of the flattening layer and the fully connected layer. Flattening is done to convert the data from the convolutional layer into a 1-dimensional feature vector since 2-D or 3-D data cannot be given as an input to the fully connected layer. In order to reduce overfitting, a dropout layer with 50% probability is used in the model. The output from this layer is then fed to a fully connected dense layer of 64 neurons along with a ReLu activation function. The final dense layer, which is the output layer consists of 2 neurons representing the two categories, one with facemask and the other without facemask, respectively. Softmax activation function is used on this last layer as it is a multi-class classification problem where the sum of all vectors is 1.

4.3.2 Splitting the dataset and Training the model:

The learning process is required to be configured first with the compile method . Adam optimizer is used here as it is computationally efficient and its implementation is straightforward. Adam is an adaptive learning rate method which means it computes individual learning rates for different parameters. The loss function used is categorical crossentropy as it is a multi-class classification. Loss function is responsible for optimizing the value of parameters during back-propagation. Metrics is set to accuracy in order to compute the accuracy rate across all predictions. To produce better accuracy, an optimized train and split is needed. The test size is set to 0.1, which means 90% of the dataset is used for training the model and the trained model is tested on 10% of the dataset which was not used for training. ModelCheckpoint is used for maintaining the validation loss where the validation data is set to 20% of the training data. Number of epochs are set to 20 to get a better optimized accuracy. Fit method is used to fit the model and train it which takes in training dataset, number of epochs and validation split as arguments.

4.4 Real-Time Detection of face using the trained model and sending email :

I. Live video capture: OpenCV is imported in order to capture live video. tkinter is used to display a dialog box when a face without a mask is detected. smtplib is used to define a smtp client session object which can be used to send an alert mail to the concerned authorities if a person without a mask is detected.

We first load the trained model into a variable using the load_modelobject.Our model is implemented using Haar Cascade Classifier to detect the front face of an individual. the cascade classifier object of OpenCV library is used to point to location of the Haar cascade classifier which is saved as an xml file. Live video is captured using VideoCapture method of the OpenCV library and the argument inside this method is set to 0. We then define two dictionaries, one containing the details of wearing a mask and not wearing a mask where the keys 0 and 1 are given the values 'Mask On' and 'No Mask' respectively. The other dictionary contains details of colour of the rectangle that are drawn around the face after being detected where green colour, with key 0 and value (0,255,0), depicts 'Mask On' and red colour, with key 1 and value (0,0,255), depicts 'No Mask'.

An infinite while loop is initiated. Inside the loop, the colour image is converted to grey using cvtColor() function and each frame is passed as an argument inside this function along with the parameter COLOR_BGR2GRAY. This grey colour image is saved in a variable. In order to get the region of interest on the face inside the frame, detectMultiScale() methods is used. It takes in several input parameters such as: the gray scale image, scale factor, which depicts how much the image size is reduced at each image scale, it is set to 1.3 and the last parameter is min neighbour, which depicts the minimum number of neighbours that a rectangle frame can have, it is set to 5. {A for loop is used to iterate through all the dimensions of the image array represented as x, y, w, h. Inside the for loop, the image Is being cropped to limit it just to the face and rescaled to size 112 X 112. This resized image is normalised by dividing it by 255}. Prediction for face mask is done on this reshaped image using the predict() associated with our trained deep learning model. The label of the image is obtained by using the argmax() method and the axis argument is assigned with value 1. Rectangle() function from the opencv library is used to draw the rectangle around the face.IT takes in four parameters: i)input image, ii) starting point of the rectangle which is x and y, iii)ending point of the rectangle which is x+w and y+h where w and h are width and height of the image respectively, iv) colour dictionary which was created before rect_colour_dict,it stores the colour of the rectangle in the BGR band and v) thickness of the

rectangle line which is set to 2. (To write text on the rectangle, such as 'Mask On' and 'No Mask', we use the puttext() function of the open cv library.)

A subject and message are defined with the text that needs to be the contents of the alert email. An if-else loop is used to send an alert e-mail if a face without a mask is detected based on the value of label. If the label is equal to 0, the person is wearing a mask and if the label is 1, the individual is not wearing a mask and an alert e-mail is sent to the concerned authorities. Smptlib is used to send a mail. If label is equal to 1, then the control into the if condition. First the message format is defined having a combination of subject and the message, which was defined earlier in the body of the email. The login() function is used to login to the Gmail account from which the mail is supposed to be sent. The parameters are the Gmail user ID and password. Sendmail() function is used to send mail to the concerned authorities. The arguments of this function are the sender email id, the receiver email id and the message to be sent. Finally, the connection is closed using the close() method.

5. CONCLUSION

To reduce the spread of covid-19 pandemic, measures must be taken. We have developed face mask detection using convolution neural network with the help of machine learning packages like TensorFlow, Keras and open CV. This face mask detector can be deployed in many areas like airport ,shopping malls and many other public places to detect if a person is wearing a mask or no and to avoid the spread of virus by checking who is following necessary rules and who is not. An alert will be sent to an official if an individual without a mask is being detected and necessary action can be taken.

The model can further be developed to detect faces wearing different kinds of masks and to detect multiple faces in motion. A better accuracy of the model can be achieved by training the model with a dataset containing large number of images wearing different kinds of masks.

6. APPLICATIONS

- It decreases the spread of virus and avoids contact of virus in various institutions like colleges and other public places etc.
- Installed in malls and various social places where the doors open, only if a person is wearing mask.
- It helps spread of virus all over the country.

7. RESULT

The model is trained and tested upon the dataset. We are able to get an accuracy of 92.74% (as shown in fig:4)for training data and an accuracy of 93.67%(as shown in fig:5)for testing data. The model is able to classify the image for with and without mask successfully(as show in fig:6 and fig:7) real time and an alert mail is sent to the mail id of the authorities when a face without mask is detected.

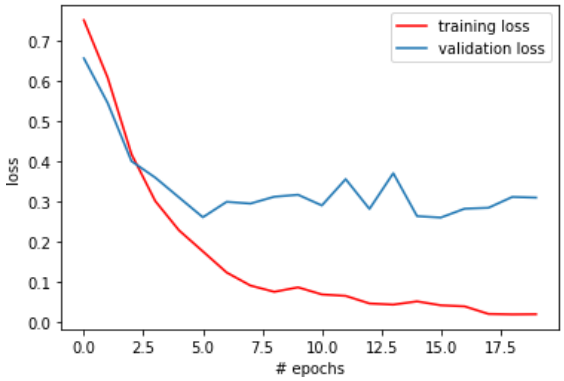

Fig:4

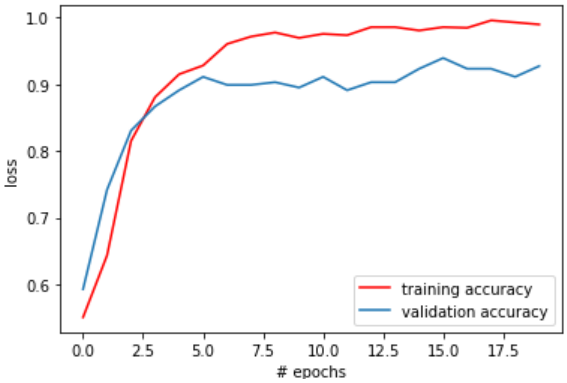

Fig:5

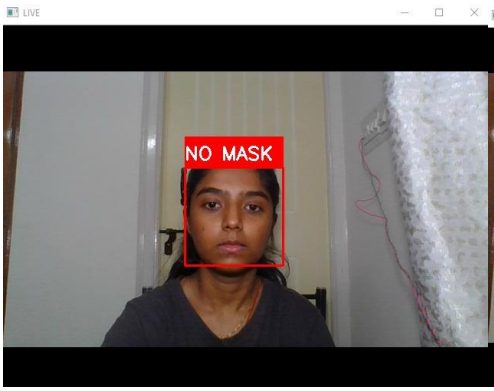

Fig:6

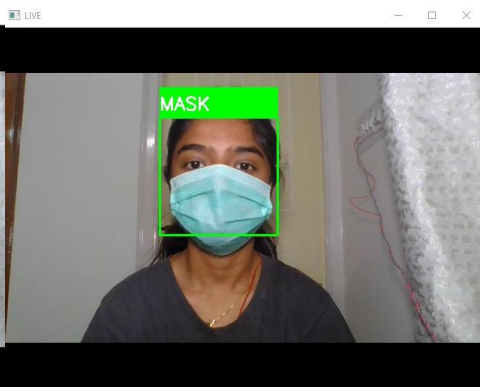

Fig:7

References:

[1] W.H.O., "Coronavirus disease 2019 (covid-19): situation report, 205". 2020.
[2] "Coronavirus Disease 2019 (COVID-19) – Symptoms", Centers for Disease Control and Prevention, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/2019-ncov/symptomstesting/symptoms.html. 2020. [3] "Coronavirus — Human Coronavirus Types — CDC", Cdc.gov, 2020. [Online]. Available: https://www.cdc.gov/coronavirus/types.html. 2020
[3] W.H.O., "Advice on the use of masks in the context of COVID-19: interim guidance", 2020.
[4] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask detector", arXiv.org, 2020.
[5]Opencv-python-tutroals.readthedocs.io. 2020. Changing Colorspaces — Opencv-Python Tutorials 1 Documentation. [online] Available at:https://opencv-python-tutroals.readthedocs.io/en/latest/py tutorials/ pyimgproc/pycolorspaces/py colorspaces.html. 2020.

[6] "Guide to the Sequential model - Keras Documentation", Faroit.com, 2020. [Online]. Available: https://faroit.com/keras-docs/1.0.1/gettingstarted/sequential-model-guide/. 2020
[7]S. Ghosh, N. Das and M. Nasipuri, "Reshaping inputs for convolutional neural network: Some common and uncommon methods", Pattern Recognition, vol. 93, pp. 79-94, 2019. Available: 10.1016/j.patcog.2019.04.009.
[8] D. Meena and R. Sharan, "An approach to face detection and recognition," 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, 2016, pp. 1-6, doi: 10.1109/ICRAIE.2016.7939462.
[9] T. VenkatNarayanaRao1 ,Gadige Vishal Sai2 , Manideep Chenna3 , M. Sahas Reddy4- An Application for the Detection of COVID-19 Face Mask Using OpenCV
[10]K. Team, "Keras documentation: MaxPooling2D layer", Keras.io, 2020. [Online]. Available: https://keras.io/api/layers/pooling layers/ max pooling2d/. 2020.
[11] ADDITIVE MANUFACTURING AND RECENT DEVELOPMENTS IN POLYMER NANOCOMPOSITES, Dr. Yogesh P. Deshmukh, Mr. Amol Bhelkar , International Journal Of Advance Research In Science And Engineering http://www.ijarse.com IJARSE, Volume No. 10, Issue No. 03, March 2021 ISSN-2319-8354(E).
[12] "TensorFlow White Papers", TensorFlow, 2020. [Online]. Available: https://www.tensorflow.org/about/bib. 2020.
[13] K. Team, "Keras documentation: About Keras", Keras.io, 2020. [Online]. Available: https://keras.io/about. 2020.
[14]"OpenCV", Opencv.org, 2020. [Online]. Available: https://opencv.org/. 2020