Real-Time Face Mask Detection and Alerting System

Neil Uri Nowgaonkar[1], Rutuja Vasantrao Nevlikar[2]

[1]UG Computer Science Student, VIT Pune, Maharashtra
[2]UG Mechanical Student, VIT Pune, Maharashtra

---***---

Abstract *—In this paper, we have a proposed a system where we check if a person is wearing a mask or not using a webcam and if he is not wearing a mask then we alert the authorities by sending an email as well as sending the image of the person in real-time, for this functionality we are using SMTP library. This system is very helpful in public places such as schools, colleges, theatres, etc which will help to stop the spread of COVID-19. The dataset that we are using for training our model was created by PyImageSearch reader Prajna Bhandary the dataset consists of 1,376 out of which 690 images are of people wearing a mask and 686 images of people not wearing a mask. Here, we have used Python, OpenCV, Keras with TensorFlow as a backend for building the Deep Learning Convolution Neural Network model. Our model is giving an accuracy of 95.35%.*

Key Words: OpenCV, Face mask detection, Keras, Deep learning, CNN, SMTP

1.INTRODUCTION

Considering the situation of the COVID-19 pandemic many people are getting infected due to the covid-19 virus. The COVID-19 virus spreads primarily by nasal discharge or droplets of saliva when an infected person sneezes or coughs, so it is necessary to take care of our health by doing simple things like cleaning hands, wearing a mask, or gargling with hot water, etc. Right now, we all are in a do-or-die situation because of these pandemics. Masks are a crucial factor to stop the transmission of coronavirus and save humankind. We all should wear a face mask while going to a crowded place. Also, we should avoid closed and close-contact settings and many more.

In this covid-19 pandemic, wearing a face mask is important for everyone but few people tend to forget to wear a mask which may be risky for him as well as for others. In large population countries, it is hard to check manually for people wearing a mask or not. The main objective is to identify whether the person on the image/video stream is wearing a face mask or not with the help of computer vision and deep learning and also to keep an eye on people to maintain safety standards at public places. Monitoring people without masks and sending an alert to notify someone is not wearing a mask through email with an image of that person attached with it. WHO stresses prioritizing medical masks and respirators for everyone [4]. Therefore, face mask detection has become a crucial task in today's world

1.1 LITERATURE REVIEW

A study on using facemask to limit the expansion of COVID-19 is introduced in [6]. The study indicated that the masks that are perfectly fit, effectively stop the spread of droplets expelled when coughing or sneezing. Masks that are not perfectly fitted, also able to retain airborne particles and viruses. Allam and Jones [7] proposed a framework on smart city networks that specialize in how data sharing should be performed during the outbreak of COVID-19. The proposed system discussed the prospects of Urban Health Data regarding the protection problems with the economy and national security. within the system, the data is collected from various points of the town using sensors, trackers, and from laboratories.

Face recognition can be done with various algorithms like Haarcascade_frontalface_default.xml B. Eigenfaces C. Fisher faces D. Local Binary Patterns Histograms, etc. The authors in [2] have detected faces by using PCA and they found challenges in facial detection as it is difficult to plan a picture before use of the tool. The paper by Vinitha. V, Valantina. V suggests an idea of MobileNetV2 by which we can deploy the model on mobile devices[3]. According to the work in [4], CNN can identify pixel-level data when compared to many algorithms available, CNN works more accurately and that paper gives an idea of object detection by using HOG and HOG+SVM techniques. In the paper on Masked face detection using viola jones algorithm, the proposed system reduces the time for detecting faces but the problem is with the brightness of images and face orientations. The system could not detect the faces with real-world problems.[1]

1.2 DATASET DESCRIPTION

The dataset that we are using for training our model was created by PyImageSearch reader Prajna Bhandary. The dataset contains a total of 1376 images out of which 690 images are of people wearing a mask and 686 images of people not wearing a mask. Dataset is available freely on Kaggle.

Fig -1: Sample Images from the used dataset

2. METHODOLOGY

2.1 Data pre-processing

Dataset pre-processing is a very essential thing to build any project. There are various data pre-processing techniques. We have used grayscaling and resizing images in our project. The motto behind using grayscaling is to reduce computational requirements and simplify algorithms. By providing grayscaled images instead of RGB to model increases the good performance of training data. Giving an RGB image may increase the confusion of the model and result in poor performance

a.) Conversion of RGB image to gray image: -

The images captured by the CCTV cameras require pre-processing before visiting the subsequent step. During the pre-processing, the image is transformed into a greyscale image as said earlier, the RGB image contains a lot of redundant information which is not necessary. On the opposite hand, the grayscale image stored 8 bits for each pixel and it contained enough information for classification.

Fig -2: Samples of converted greyscale images

b.) Image Reshaping: -

Convolution Neural Networks require a fixed size of an input image. Therefore, to maintain the uniformity of input images we reshape the image into (112 x 112) shape. Later, we normalize the images so that the value of the pixel resides in the range from 0 to 1. Using the command: images=np.reshape(images,(images.shape[0],img_rows,img_cols,1)) with 1 signifying a grayscale image. CNN can only accept fine-tuned images. This engenders several problems throughout data collection and implementation of the model. However, reconfiguring the input images before adding them to the network can help to overcome this constraint. Therefore, we use normalization which also helps the learning algorithm to learn faster and capture necessary features from the images.

2.2 Training of the model

a.) Build a Convolution Neural Network Model

To build the CNN model we have used the Sequential model of Keras. Two convolution layers will deal with input images and see them as 2- Dimensional matrices. The two layers used group Convolution, Rectified Linear activation (ReLU), and Max Pooling layers. Kernel size is set to 3 x 3 to specify the height and width of the 2D convolution window. We use the Rectified Linear Activation function (ReLU) which helps to overcome the vanishing gradient problem, allowing models to perform better and learn faster.

Max Pooling layers are used to reduce the size of the data as it removes the noise from the data and extracting only the significant one, we reduce overfitting and speed up the computation. Next, we flatten the data to convert the data into a 1-dimensional array for inputting it to the next layer. We flatten the output of the convolutional layers to create a single vector. To avoid overfitting, we use a dropout layer which has the probability of 0.5 to setting inputs to zero is added to the model. The final dense layer with two outputs for categorization uses the SoftMax function which assigns decimal probabilities to each class which helps training converge more quickly than it otherwise would.

Next, Adam optimizer is used as it combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy images. As this is a multi-class classification problem, we use categorical cross-entropy loss function for optimization during the training of the model. As this is a classification problem we set 'accuracy' as the metric.

b.) Splitting the dataset and training the Convolution Neural Network model: -

After creating a model, we need to train it using a specific dataset and then test it on a different dataset. The validation split is set to 0.25 i.e., 75% of the data is used for training and the rest 25% is used for testing our model. The model is trained for 100 epochs (iterations) which maintains a trade-off between accuracy and chances of overfitting.

c.) Sending mail to authorities

When CCTV is detecting a person, we make a rectangle near the face to show the interest of region if the person is wearing a mask green rectangle is shown else, we make a red rectangle and throw a warning message to tell the user to wear a mask. Also, trigger an email to the administrator with an image attached of the person who isn't wearing a mask. Before sending the email, we specify to whom should we send the email here we declare the email of the authorities and also a message stating a warning that there is a person. Imported EmailMessage class from email.message Python module as EmailMessage is the base class for the email object model and offers the functionality for accessing message bodies. We use the EmailMessage class so that we don't have to use the f-string to concatenate our body and subject. We have created its object to define each component and then send it using the SendMail() function. We have used Context Managers as they ensure efficient resource management to open the file and store its data.

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 09 | Sep 2021 www.irjet.net p-ISSN: 2395-0072

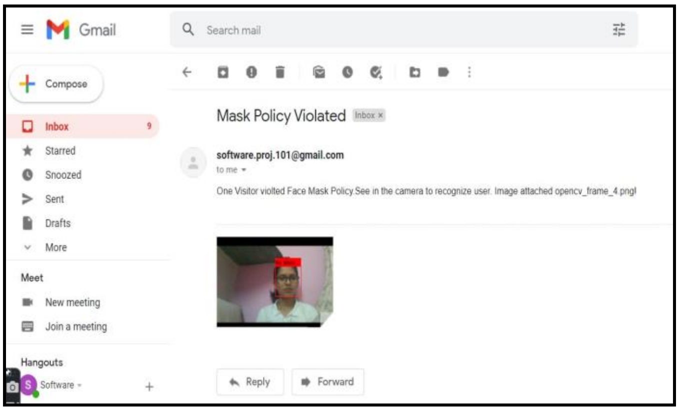

Fig -3: Image of the mail sent to authorities

Table -1: Description of various layers and variables

Description	Value
Num_classes	2
Batch_size	32
Nodes in First Layer	128
Nodes in Second Layer	64
Dropout Probability	0.5
Nodes in Dense Layer	2
Learning_rate	0.001
Epoch	100

2.3 Flowchart

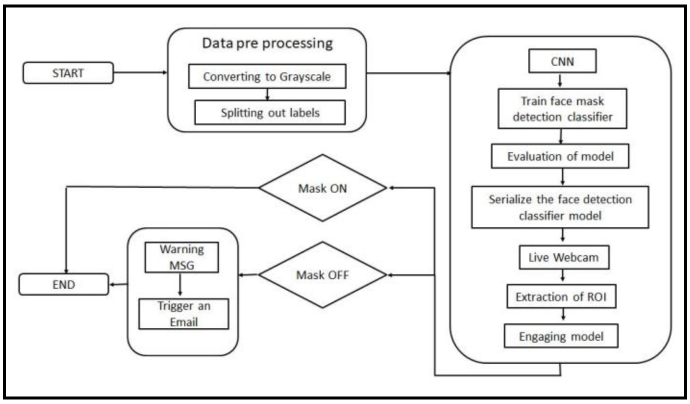

Fig -4: Flowchart

3.1 Results and Analysis

The model is trained, validated, and tested upon datasets. The number of epochs used here is 100. Increasing epochs will give better accuracy but also increases the training time of the dataset. The graph of training and validation loss gives an idea about the fitting of the model. The gap between the lines should be minimal. The minimal gap talks about the perfect fit of the model. Another graph of accuracy gives an idea about the accuracy of the model. These graphs are plotted to check the underfitting or overfitting of the model and to adjust hyperparameters. The model should be generalized to give accurate output. If the model is under fitted or overfitted then it will not produce a good output. Hyperparameters mean many nodes per layer of neural network and the number of layers in the network can make a significant impact on the performance of the model. Both the graphs of the model are good and the accuracy comes to be 95.35 %.

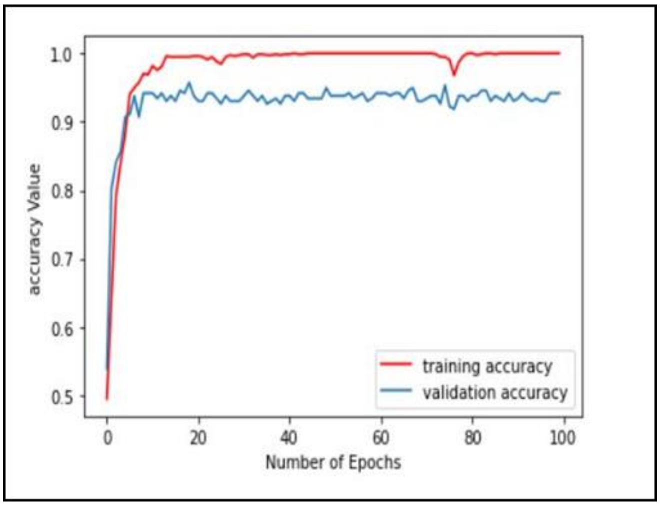

Fig -5: Accuracy of the developed system for training and testing phase

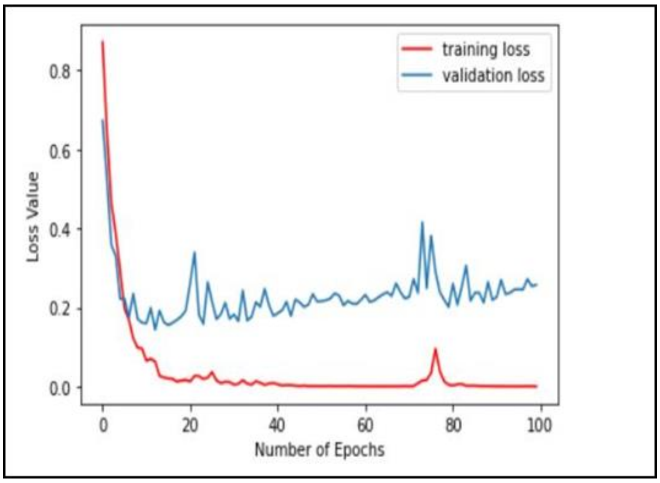

Fig -6: Loss of the developed system for training and testing phase

3.2 Limitations

As each system has limitations, so has this system its limitations such as it might classify sometimes a person to be not wearing a mask even though he is wearing a mask as well as if there are many people in the same camera frame it might not classify correctly. Also, if a person is moving very fast it might be difficult for the proposed system to classify him correctly.

3.3 Future Scope

In the future, this system can also be merged with an advanced retina scanner so that we can identify people in companies, schools, and colleges and create an automatic attendance system. Also, while using this system in public places such as various streets, we can add the geographic coordinates of that CCTV in the email so that we can inform the authorities exactly where unmasked people are located.

4. CONCLUSIONS

This paper presents a system to reduce the spread of coronavirus by informing the authority about the person who is not wearing a facial mask that is a precautionary measure of COVID-19. The system contains a face mask detection architecture where a deep learning algorithm is used to detect the mask and if no mask is detected with help of the SMTP library, we are automatically emailing the authorities with the image of the person who has not worn it. To train the model, labelled image data are used where the images were facial images with masks and without a mask. The proposed system detects a face mask with an accuracy of 99.87%. The system proposed is a valuable tool to strictly impose the use of masks in public places.

REFERENCES

[1] Aishwarya Radhakrishnan Nair, Dr. Amol D. Potgantwar," Masked face detection using viola jones algorithm: A progressive approach to less time consumption", December 2018International Journal of Recent Contributions from Engineering Science & IT (iJES) 6(4):4 DOI:10.3991/ijes.v6i4.9317

[2] "Face Detection and Recognition Using OpenCV",2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)

[3] Vinitha.V, Velantina.V." COVID-19 FACEMASK DETECTION WITH DEEP LEARNING AND COMPUTER VISION", International Research Journal of Engineering and Technology (IRJET)Volume: 07 Issue: 08 | Aug 2020

[4] T Subhamastan Rao, S Anjali Devi, P Dileep, M Sitha Ram," A Novel Approach To Detect Face Mask To Control Covid Using Deep Learning", European Journal of Molecular & Clinical Medicine ISSN 2515-8260 Volume 07, Issue 06, 2020

[5] A. Khan, A. Sohail, U. Zahoora and A. S. Qureshi, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks", Artif. Intell. Rev., Jan. 2019.

[6] L. P. Garcia, "Uso de máscara facial para limitar a transmissão da COVID-19", Epidemiol. e Serv. saude Rev. do Sist. Unico Saude do Bras., vol. 29, no. 2, pp. e2020023, 2020.

[7] Z. Allam and D. S. Jones, "On the Coronavirus (COVID-19) Outbreak and the Smart City Network: Universal Data Sharing Standards Coupled with Artificial Intelligence (AI) to Benefit Urban Health

BIOGRAPHIES

Neil Uri Nowgaonkar
Third year Computer engineering student from Vishwakarma Institute of Technology Pune, Maharashtra

Rutuja Vasantrao Nevlikar
Third year Mechanical engineering student from Vishwakarma Institute of Technology Pune, Maharashtra