Real-Time Face Mask Detection

Project report in partial fulfilment of the requirement for the award of the degree of

Bachelor of Technology

In

Computer Science

Submitted By

Animesh Prasadi	University Roll No. 12019009023015
Subham Bhattacharyya	University Roll No. 12019009001401
Sudipta Narayan Dhar	University Roll No. 12019009022032
Sourav Sarkar	University Roll No. 12019009022024
Swarnava Halder	University Roll No. 12019009022031
MD. Risher Ali	University Roll No. 12019009023069

Under the guidance of

Prof. Sukanya Roy

&

Prof. Prasenjit Kumar Das

Department of Computer Science

UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA

University Area, Plot No. III – B/5, New Town, Action Area – III, Kolkata – 700160

CERTIFICATE

This is to certify that the project titled **Real-Time Face Mask Detection** submitted by **Animesh Prasadi (University Roll No. 12019009023015), Subham Bhattacharyya (University Roll No. 12019009001401)**, **Sudipta Narayan Dhar (University Roll No**. **12019009022032), Sourav Sarkar (University Roll No. 12019009022024)**, **Swarnava Halder (University Roll No. 12019009022031) and MD. Risher Ali (University Roll No. 12019009023069)** students of UNIVERSITY OF ENGINEERING & MANAGEMENT, KOLKATA, in partial fulfilment of requirement for the degree of Bachelor of Computer Science Engineering, is a bonafide work carried out by them under the supervision and guidance of **Prof. Sukanya Roy** & **Prof. Prasenjit Kumar Das** during 5th Semester of academic session of 2021 - 2022. The content of this report has not been submitted to any other university or institute. I am glad to inform that the work is entirely original and its performance is found to be quite satisfactory.

Prof. Sukanya Roy	Prof. Prasenjit Kumar Das
Assistant Professor	Assistant Professor
Department of Computer Science	Department of Computer Science
UEM, Kolkata	UEM, Kolkata

Prof. (Dr.)Sukalyan Goswami

Head of the Department

Department of Computer Science

UEM, Kolkata

ACKNOWLEDGEMENT

We would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to us.

We are sincerely grateful to our **Guide Prof. Sukanya Roy** and **Co-guide Prof. Prasenjit Kumar Das** of the Department of Computer Science, UEM, Kolkata, for his wisdom, guidance and inspiration that helped us to go through with this project and take it to where it stands now.

We would also like to express our sincere gratitude to **Prof. (Dr.)Sukalyan Goswami**, HOD, Computer Science, UEM-Kolkata and all other departmental faculties for their ever-present assistant and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Animesh Prasadi

Subham Bhattacharyya

Sudipta Narayan Dhar

Sourav Sarkar

Swarnava Halder

MD. Risher Ali

TABLE OF CONTENTS

Abstract

Since the infectious coronavirus disease (COVID-19) was first reported in Wuhan, it has become a public health problem in China and even around the world. This pandemic is having devastating effects on societies and economies around the world. The increase in the number of COVID-19 tests gives more information about the epidemic spread, which may lead to the possibility of surrounding it to prevent further infections. However, wearing a face mask that prevents the transmission of droplets in the air and maintaining an appropriate physical distance between people, and reducing close contact with each other can still be beneficial in combating this pandemic. Therefore, this research paper focuses on implementing a Face Mask and Social Distancing Detection model as an embedded vision system. The pretrained models such as the MobileNet, ResNet Classifier, and VGG are used in our context. People violating social distancing or not wearing masks were detected. After implementing and deploying the models, the selected one achieved a confidence score of 100%. This paper also provides a comparative study of different face detection and face mask classification models. The system performance is evaluated in terms of precision, recall, F1-score, support, sensitivity, specificity, and accuracy that demonstrate the practical applicability. The system performs with F1-score of 99%, sensitivity of 99%, specificity of 99%, and an accuracy of 100%. Hence, this solution tracks the people with or without masks in a real-time scenario and ensures social distancing by generating an alarm if there is a violation in the scene or in public places. This can be used with the existing embedded camera infrastructure to enable these analytics which can be applied to various verticals, as well as in an office building or at airport terminals/gates.

Introduction

The COVID19 virus can be spread through contact and contaminated surfaces. There are so many essential equipments needed to fight against the Corona virus. One of such most essential is Face Mask. Firstly, a face mask was not mandatory for everyone but as the day progresses scientists and Doctors have recommended everyone to wear a face mask. So to detect whether a person is wearing Face Mask or not is an essential process to implement in the society currently which can be used for various applications like at the airport, hospitals, offices, schools, etc. This system can be of great importance at airports to detect travelers whether they are wearing a mask or not and at schools to ensure students are wearing a face mask for their safety. However, wearing the mask face causes the following problems: i) fraudsters and thieves take advantage of the mask, stealing and committing crimes without being identified. ii) community access control and face authentication have become very difficult tasks when the most part of the face is hidden by a mask. Hence, detecting the face mask and recognizing the person behind the face mask is very important.

In this paper, we will address the different approaches tried to implement the face detection and face recognition system along with the presence of face masks from various papers. To study the existing techniques and analyze which approach is feasible and efficient enough to implement to the current state of the society, we have organized this paper into different sections. Later, by considering the constraints and drawbacks of each approach the best techniques will be concluded. Recognition has to classify a given face, and there are as many classes as candidates. Consequently, many face detection methods are very similar to face recognition algorithms.

Literature Survey

1. Related Works

Deep learning is an important breakthrough in the AI field. It has recently shown enormous potential for extracting tiny features in image analysis. Due to the COVID-19 epidemic, some deep learning approaches have been proposed to detect patients infected with coronavirus. In this context, and unlike bacterial pneumonia, many other types of lung infections caused by viruses are called viral pneumonia. These viruses, such as the COVID-19, infect the lungs by blocking the oxygen flow, which can be life-threatening. This motivated researchers to develop many frameworks and schemes based on AI tools in the fight against this dangerous virus.

2. Deep Learning Tools and CT Image-Based COVID-19 Detection

Computed tomography scan or CT scan is a medical imaging technique utilized in radiology in order to get detailed images of the body for diagnosis purposes. Accurate and fast COVID-19 screening is achievable using CT scan images. Various works have been carried out in this context. In, Shah et al. proposed distinct deep learning techniques to differentiate CT scan images of both COVID-19 and non-COVID-19, which helps in diagnosis. In the dataset, we find 349 images corresponding to patients with COVID-19 and 463 images corresponding to patients without COVID-19. These images were divided into three sets: 80% of them for training set, 10% for validation, and 10% for testing. From the different techniques presented in this work, we cite CTnet-10, which is a self-developed model having an accuracy of 82.1%. We can also cite VGG-16, ResNet-50, InceptionV3, VGG-19, and DenseNet-169, having an accuracy of 89%, 60%, 53.4%, 94.52%, and 93.15%, respectively. The accuracy of VGG-19 is the best as compared to other models. To predict the results, CTnet-10 takes only 12.33 ms. This method is well-organized. It is useful for doctors, especially in mass screening. All the automatic diagnosis methods presented previously can be used by doctors for COVID-19 screening.

3. Deep Learning Tools and CXR Image-Based COVID-19 Detection

Radiography is a technique used to quantify the functional and structural consequences of chest diseases, to provide high-resolution images on disease progression. Several works have been carried out in this context. Echtioui et al., in , proposed a new CNN-based method for COVID-19 recognition, through analyzing radiographic images of a patient's lungs. The aim of this scheme is to provide clinical decision support for healthcare workers and also for researchers. Hence, performance results, as well as the accuracy value of about 91.34%, and the other metrics in terms of recall, precision, and F1-score, prove the efficiency of the method. In the same context, Ozturk et al. in, introduced a new automatic COVID-19 detection model using CXR images denoted by the "DarkCovidNet." It is used to provide correct diagnosis for both a binary classification (COVID-19 VS no findings) and a multiclass classification (COVID-19 VS pneumonia VS no findings). For binary classes, the classification accuracy produced by this model is about 98.08%, but, for multiclass cases, the accuracy is 87.02%. To validate their initial screening, radiologists can use the model to assist them. This model can be employed also via cloud to screen patients immediately. As a solution to the shortage of radiologists, this method can be used in remote places especially in countries affected by COVID-19. The most important advantage of this method is that such models can be used to diagnose supplementary chest-related diseases such as tuberculosis and pneumonia. However, the proposed work fits well into the COVID-19 detection phase, but to ensure its efficiency and model reliability, the authors may augment the dataset and retrain the proposed model. With the same idea, Wang et al. proposed a deep CNN model which is presented in. Their model, called COVID-Net, is open source and accessible to the general public. The test accuracy achieved by this model is 93.3%. Therefore, this model makes predictions which can assist clinicians in improving screening, transparency, and trust.

4. MobileNet Masking Model

All governments around the world are struggling against COVID-19, which causes serious health crises. Therefore, the use of face masks regulatory can slow down the high spread of this virus. In this context, Dey et al. proposed in [6] a deep learning-based model for detecting face mask. This model named "MobileNet Mask" is multiphase. A pretrained model of the ResNet-10 architecture is utilized to find faces in video stream. Also, numerous steps are used such as charging the classifier (MobileNet), building the FC layer, and testing phase. All the experimental cases are supervised on Google Colab that runs in the cloud and is provided with over 12 GB of RAM. Different performance metrics (accuracy, F1-score, precision, and recall) are used to judge the performance of the proposed model.

Two distinct face mask datasets are used to train and test the model. The first dataset, named IDS1, consists of 3835 images, divided into two classes: 1916 images of faces with masks and 1919 images without masks. Kaggle dataset, RMFD, and Bing Search API are the source of the typical images of this dataset. The second dataset, named IDS2, consists of 1376 images, divided into two classes: 690 images of faces with masks and 686 images without masks. The sample images of this dataset are gathered from the SMFD.

For all experiments, 80% of the datasets are dedicated for training and 20% for testing. When testing the model, it achieved an accuracy of 93% in IDS1, but almost 100% in IDS2. Comparing the results of their model with those of state-of-art models available in the literature, Dey et al. found that the accuracy of theirs is higher. The main advantage is that their model can be implemented on light-weight embedded computing devices.

5. Hybrid Model

All over the world, the trend of wearing masks is rising because of COVID-19 pandemic. In this context, a hybrid model utilizing deep learning with classical machine learning (ML) to detect masked faces is presented by Loey et al. in [1]. The model proposed by the authors includes two phases: the feature extraction process applying ResNet-50, and the classification process. The ResNet-50 model used like a feature extraction is composed of 50 deep layers. A convolutional layer (CNVL) is the start of the model, a fully connected layer (FCL) is the end of the model, and 16 residual bottleneck blocks are in between them.

To improve the performance of the model, three traditional classifiers replaced the last layer of the ResNet-50 during the classification process. The classification of face masks is released by three algorithms: the DT, the SVM, and the ensemble algorithm. The SVM is a machine learning algorithm designed for classification. It is one of the most popular supervised learning techniques. The DT is a model of classification based on information gain and entropy function. Ensemble methods are a combination of algorithms of machine learning which generate a collection of classifiers.

Problem Statement & Requirement

1. **Problem Statement**

In this Project we will be conducting a AI based face recognition model that is capable enough to detect if a person wearing a Face-Mask or not.

This Program is made in mind that in future we can invest more time on it to make it a easy to plug-in AI based model which can be used in everywhere i.e. Security camera, public camera, various institution , Hospital etc

Although the process may sound simple but there goes a lot of understanding of AI based system and how they work. Training an AI maybe the 1st step but training what to recognise might be the challenging part.

2. Requirement to run the program

To Run this program, 1st of all we need a webcam or any form of capture device that can send facial data to the data model of this Program. 2nd we will need a compiler that can run Python based program code. 3rd we will need a PC or Laptop, capable enough to run this program although this program is made in mind that every lower end devices can run the algorithms thus compatibility will not be a issue.

We will using Anaconda and jupyter Notebook to compile this Program.

Source Code & Results Analysis

1. Source Code

Filename - 1.0 data preprocessing.ipynb

```
import cv2,os


data_path='dataset'       #Giving the path of the dataset

categories=os.listdir(data_path)      #Loading all the folders in dataset

labels=[i for i in range(len(categories))]  #setting labels for the categories


label_dict=dict(zip(categories,labels))


print(label_dict)

print(categories)

print(labels)
```

```
img_size=100

data=[]

target=[]


for category in categories:

    folder_path=os.path.join(data_path,category)

    #Loading the images in the categories(folders)

    img_names=os.listdir(folder_path)


    for img_name in img_names:

        img_path=os.path.join(folder_path,img_name)  #Reading every image

        img=cv2.imread(img_path)

        try:
```

```
            gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

            #Coverting the image into gray scale

            resized=cv2.resize(gray,(img_size,img_size))

            #resizing the gray scale into 100x100, since we need a fixed common size for
             all the images in the dataset

            data.append(resized)

            target.append(label_dict[category])

            #appending the image and the label(categorized) into the list (dataset)


        except Exception as e:

            print('Exception:',e)

            #if any exception rasied, the exception will be printed here. And pass to
             the next image
```

```
import numpy as np


data=np.array(data)/255.0  #normalizing and converting the pixel range to 0 and 1

data=np.reshape(data,(data.shape[0],img_size,img_size,1)) #Creating 4D array

target=np.array(target)


from keras.utils import np_utils

new_target=np_utils.to_categorical(target)

#converting the target into categorical representations
```

```
np.save('data',data)    #contains images

np.save('target',new_target)  #contains whether image is with mask or without mask
```

Filename - 2.0 training the CNN.ipynb

```
import numpy as np


data=np.load('data.npy')
```

```
target=np.load('target.npy')


#loading the save numpy arrays in the previous code
```

```
from keras.models import Sequential

from keras.layers import Dense,Activation,Flatten,Dropout

from keras.layers import Conv2D,MaxPooling2D

from keras.callbacks import ModelCheckpoint


model=Sequential()


model.add(Conv2D(200,(3,3),input_shape=data.shape[1:]))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

#The first CNN layer followed by Relu and MaxPooling layers


model.add(Conv2D(100,(3,3)))

model.add(Activation('relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

#The second convolution layer followed by Relu and MaxPooling layers


model.add(Flatten())

model.add(Dropout(0.5)) #To reduce Overfitting

#Flatten layer to stack the output convolutions from second convolution layer

model.add(Dense(50,activation='relu'))

#Dense layer of 64 neurons

model.add(Dense(2,activation='softmax'))

#The Final layer with two outputs for two categories


model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```python
from sklearn.model_selection import train_test_split

train_data,test_data,train_target,test_target=train_test_split(data,target,test_size=0.1)

checkpoint = ModelCheckpoint('model-
{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,mode='auto')

history=model.fit(train_data,train_target,epochs=20,callbacks=[checkpoint],validation_sp
lit=0.2)
```

```python
from matplotlib import pyplot as plt


plt.plot(history.history['loss'],'r',label='training loss')

plt.plot(history.history['val_loss'],label='validation loss')

plt.xlabel('# epochs')

plt.ylabel('loss')

plt.legend()

plt.show()
```

```python
plt.plot(history.history['accuracy'],'r',label='training accuracy')

plt.plot(history.history['val_accuracy'],label='validation accuracy')

plt.xlabel('# epochs')

plt.ylabel('loss')

plt.legend()

plt.show()
```

```python
print(model.evaluate(test_data,test_target))
```

Filename - 3.0 detecting Masks.ipynb

```python
from keras.models import load_model

import cv2

import numpy as np
```

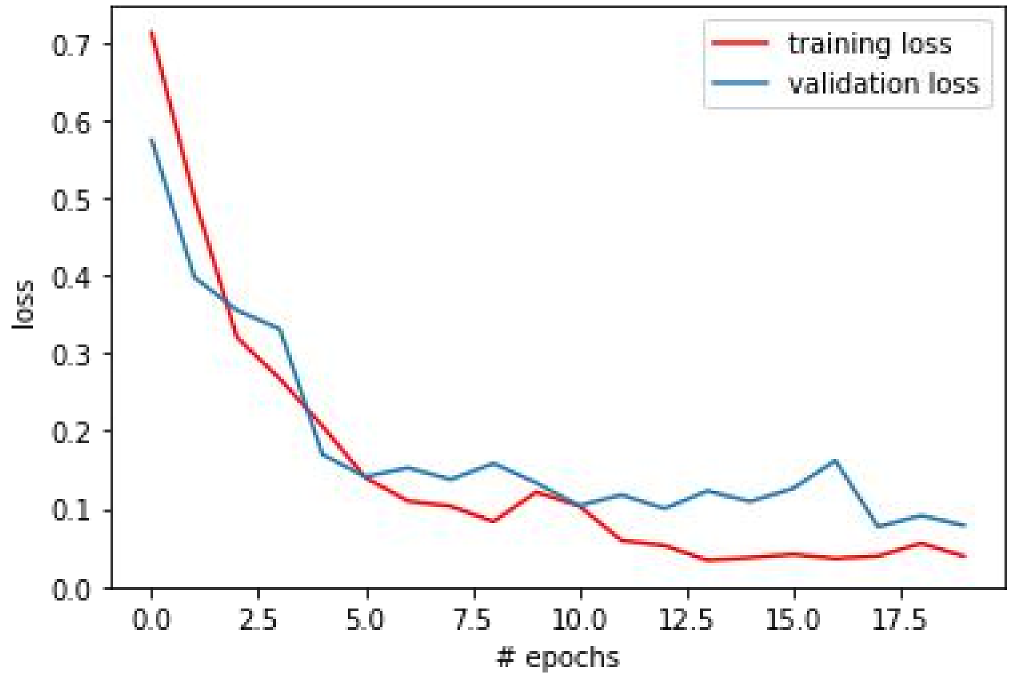
```
model = load_model('model-018.model')


face_clsfr=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')


source=cv2.VideoCapture(0)


labels_dict={0:'WITH MASK',1:'NO MASK'}

color_dict={0:(0,255,0),1:(0,0,255)}
```

```
while(True):


    ret,img=source.read()

    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces=face_clsfr.detectMultiScale(gray,1.3,5)


    for x,y,w,h in faces:


        face_img=gray[y:y+w,x:x+w]

        resized=cv2.resize(face_img,(100,100))

        normalized=resized/255.0

        reshaped=np.reshape(normalized,(1,100,100,1))

        result=model.predict(reshaped)


        label=np.argmax(result,axis=1)[0]

        #This returns the column for the maximum probability


        cv2.rectangle(img,(x,y),(x+w,y+h),color_dict[label],2)

        cv2.rectangle(img,(x,y-40),(x+w,y),color_dict[label],-1)

        cv2.putText(img, labels_dict[label], (x, y-10),
            cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)
```

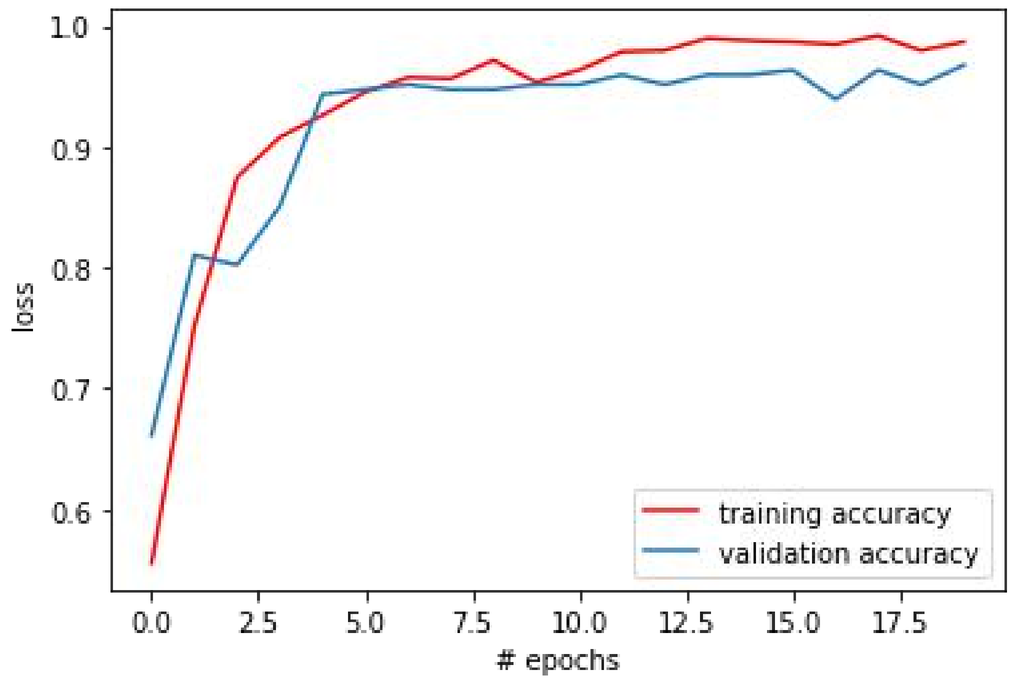
```
    cv2.imshow('LIVE',img)

    key=cv2.waitKey(1)


    if(key==27):

        break


    cv2.destroyAllWindows()

source.release()
```

2. Result Analysis

We got the following graph of the losses that occurred:

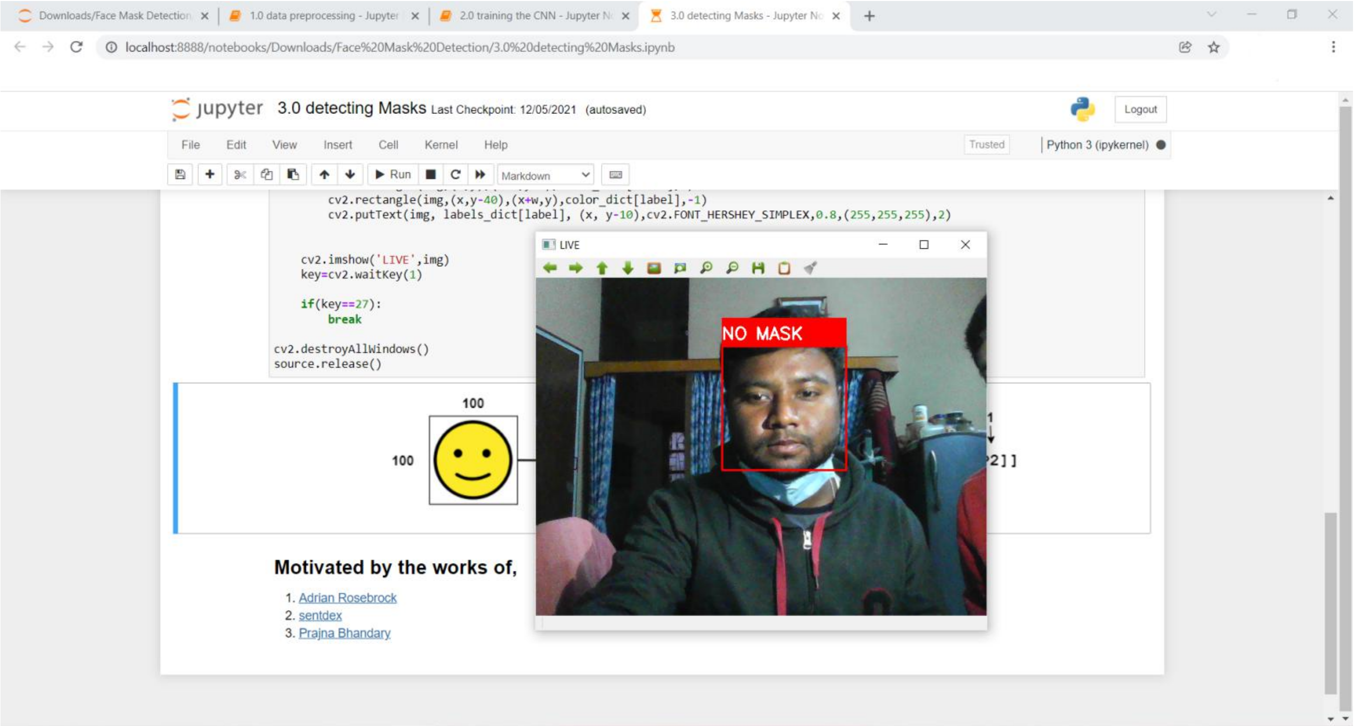


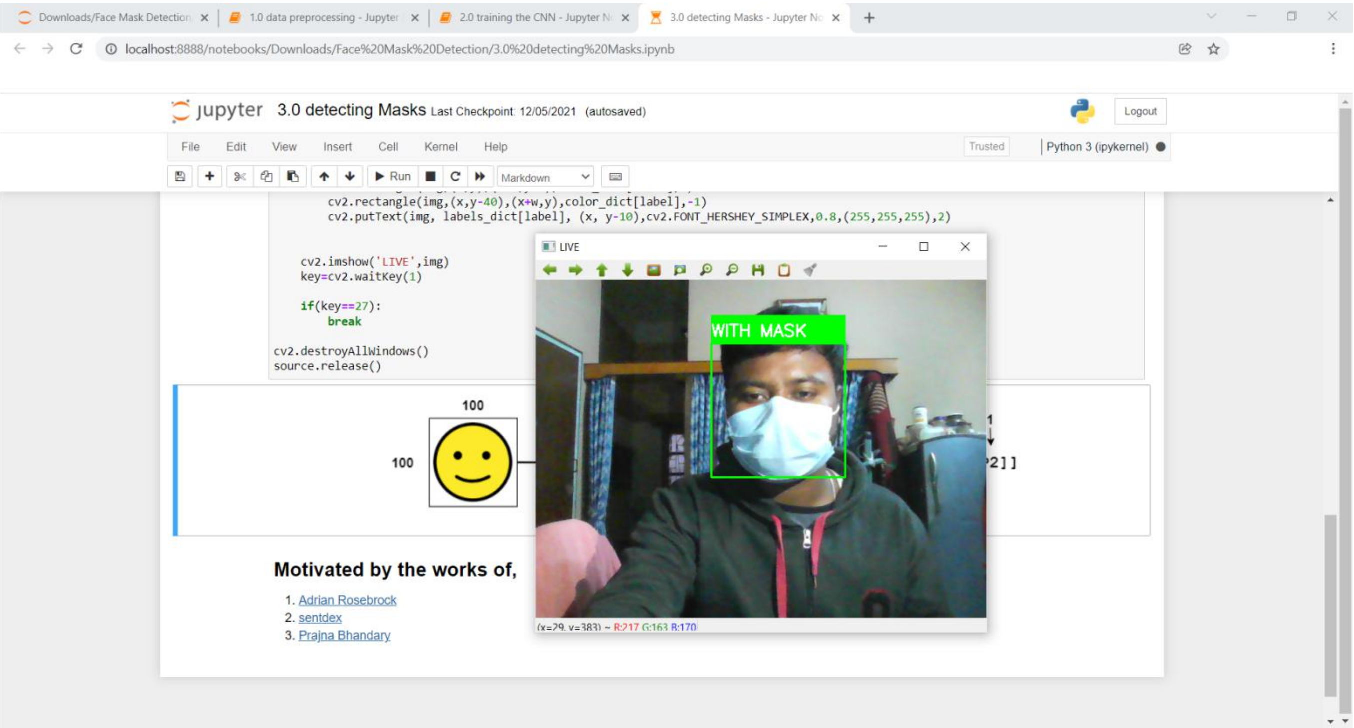
This is the graph of the accuracy we got after training:

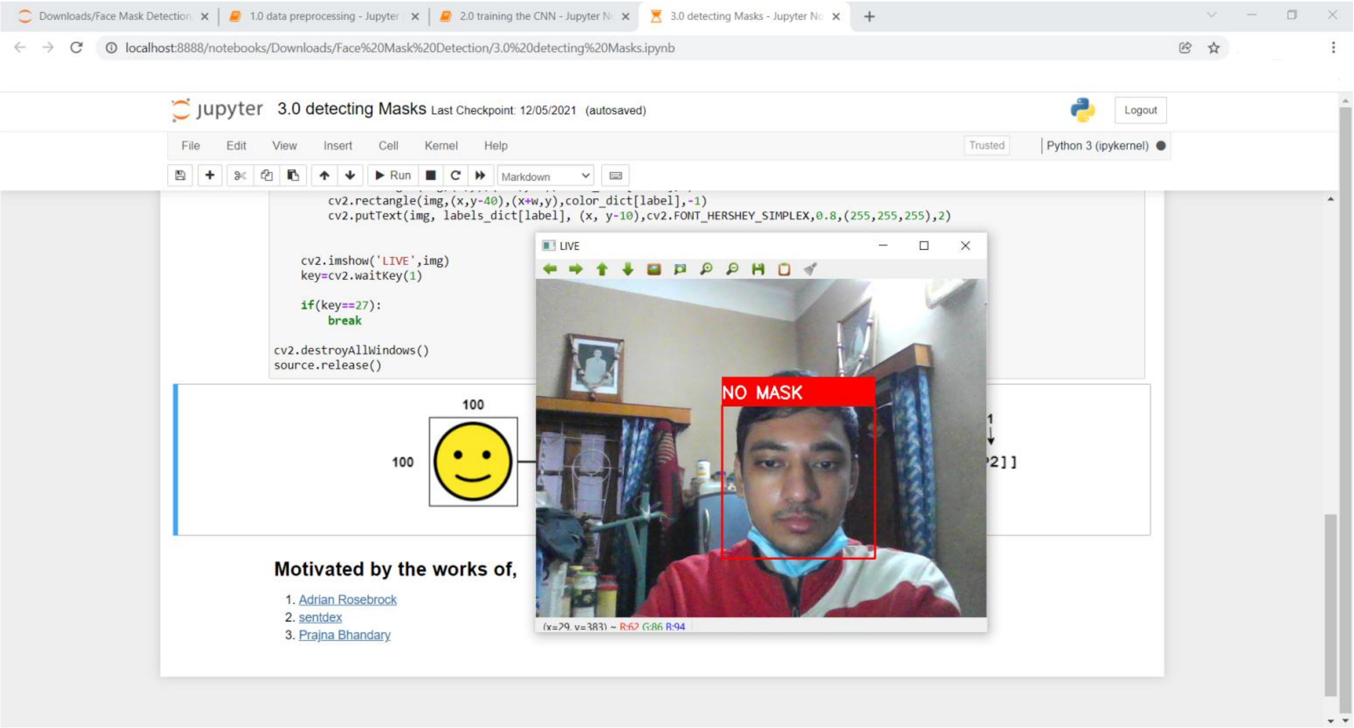


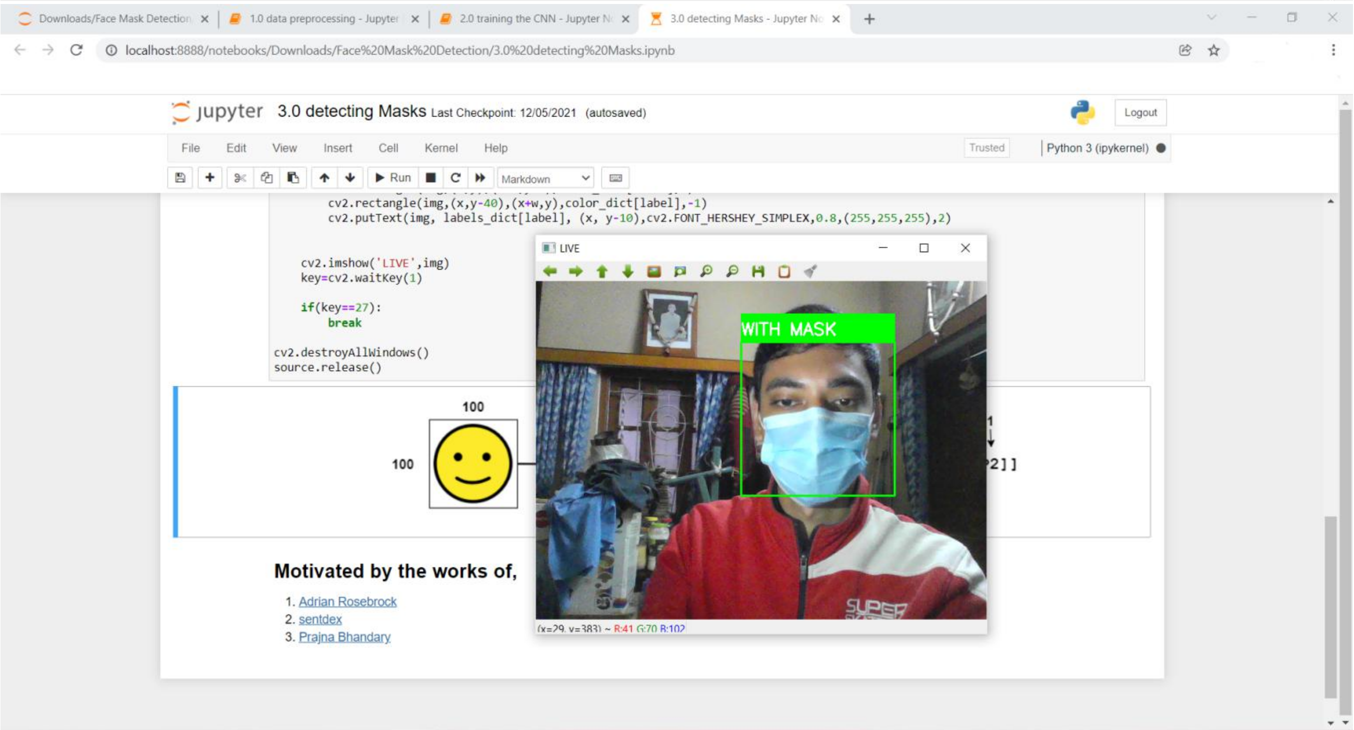
As we can see from the above graphs, the losses gradually decreased with each model and the accuracy gradually increased. After training, we chose the best model which got the maximum accuracy among the other trained models.

The Following the Test Result of the Source code:









Conclusion

Due to the urgency of controlling COVID-19, the application value and importance of real-time mask and social distancing detection are increasing. This work reviewed, firstly, many research works that seek to surround COVID-19 outbreak. Then, it clarified the basic concepts of deep CNN models. After that, this paper reproduced the training and testing of the most used deep pretrained-based CNN models (DenseNet, InceptionV3, MobileNet, MobileNetV2, ResNet-50, VGG-16, and VGG-19) on the face mask dataset. Finally and after evaluated the numerical results, best models are tested on an embedded vision system consisted of Raspberry Pi board and webcam where efficient real-time deep learning-based techniques are implemented with a social distancing task to automate the process of detecting masked faces and violated or maintained distance between peoples.

This embedded vision-based application can be used in any working environment such as public place, station, corporate environment, streets, shopping malls, and examination centers, where accuracy and precision are highly desired to serve the purpose. It can be used in smart city innovation, and it would boost up the development process in many developing countries. Our framework presents a chance to be more ready for the next crisis or to evaluate the effects of huge scope social change in respecting sanitary protection rules.

Future Scope

Human recognition with face mask has various applications in different domains. The various methodologies discussed in this paper can be based on the particular demands of the application. As every approach has its very own pros and cons we need to determine the best approach according to the necessity. Face detection is gaining the interest of marketers. It can be used at various domains like airports where this system can be of great importance at airports to detect travellers whether they are wearing mask or not. Travellers data can be captured as videos in the system at the entrance. Hospitals – This system can be integrated with CCTV cameras and that data may be administered to see if their staff is wearing mask or not. Offices – This system can help in maintaining safety standards to prevent the spread of Covid- 19, to detect whether the person is wearing mask or not. The scope of this system extends to security systems of wide range right from Malls, hospitals, IT companies and in many such public areas.

BIBLIOGRAPHY

1. Mamata s. Kalas, REAL TIME FACE DETECTION AND TRACKING USING OPENCV , International Journal of Soft Computing and Artificial Intelligence, ISSN: 2321-404X, Volume-2, Issue-1, May-2014

2. Walid Hariri, Efficient Masked Face Recognition Method during the COVID-19 Pandemic, pp.1-7, July 2020

3. Vinitha.V1, Velantina.V2, Covid-19 facemask detection with deep learning and computer vision International Research Journal of Engineering and Technology (IRJET), Volume: 07, pp.1-6, Aug 2020.

4. Yassin Kortli, Maher Jridi, Ayman Al Falou, and Mohamed Atri, face recognition systems: a survey, 20(2): 342, pp.1-10, 2020 Jan 7

5. Vinitha.V1, Velantina.V2, COVID-19 FACEMASK DETECTION WITH DEEP LEARNING AND COMPUTER VISION International Research Journal of Engineering and Technology (IRJET), Volume: 07, pp.1-6, Aug 2020.