North South University
Department of Electrical & Computer Engineering

Project Report

(Face Mask Detector)

Course Code: CSE299

Course Title: Junior Design Course

Course Instructor: Dr. Sifat Momen (SfM1)

Section: 02

GROUP-06

GROUP MEMBERS:

- 1931024042 **Shadman Sakib**
- 1931132042 **Rahul Deb Roy**
- 1931333042 **Moriom Islam Mou**
- 1931795642 **Akil Hossain Prottoy**

Date: 17/12/22

Abstract

The world has been severely impacted by the coronavirus epidemic in 2019. After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one. Additionally, a lot of public service providers only permit users to access the service when the proper masks are worn. Only a small number of studies have examined face mask identification using image analysis, nevertheless. The basic aim of the project is to detect the presence of a face mask on human faces on live streaming video as well as on images. We have used deep learning to develop our face detector model. The architecture used for the object detection purpose is Single Shot Detector (SSD) because of its good performance accuracy and high speed. Alongside this, we have used basic concepts of transfer learning in neural networks to finally output presence or absence of a face mask in an image or a video stream. Experimental results show that our model performs well on the test data with 100% and 99% precision and recall, respectively. In this project, we propose automating the manual mask detector by applying image processing. Convolutional Neural Network is the method utilized for image segmentation and classification.

Contents

Chapter 1

1 Introduction

1.1 Overview

According to the World Health Organization's (WHO) Situation Report No. 96, the coronavirus illness of 2019 (COVID-19) has killed more than 180,000 people and infected more than 2.7 million people worldwide. Additionally, there are several other comparable severe respiratory illnesses that affect a large population, such as the Middle East respiratory syndrome (MERS) and the Severe Acute Respiratory Syndrome (SARS), which have emerged in recent years. It has been reported that COVID-19 has a higher reproductive rate than SARS. As a result, public health is seen as governments' top priority since people are becoming more and more worried about their own health. Thankfully, research indicated that surgical face masks helped slow the transmission of coronavirus. The World Health Organization currently advises using a face mask if you have respiratory symptoms or are caring for someone who does. Additionally, a lot of public service providers only allow users to use the service if they are disguised. Consequently, face mask detection has emerged as a critical computer vision problem to support the global society, yet there is little existing research in this area.

Face mask detection is the process of determining if someone is wearing a mask or not and where their face is. The issue is closely related to generic object detection, which is used to identify classes of items, and face detection, which is used to identify a specific class of objects, namely faces. Object and face detection have various uses, including autonomous driving, teaching, spying, and other fields. Traditionally used feature extractors serve as the foundation for object detectors. While other works employ alternative feature extractors, such as the histogram of oriented gradients (HOG), scale-invariant feature transform (SIFT), and so on. Deep learning-based object detectors have recently displayed great performance and are dominating the creation of contemporary object detectors. Deep learning enables neural networks to learn features end-to-end without the need for prior knowledge to create feature extractors. There are one-stage and two-stage deep learning based object detectors. One-stage detectors use a single neural network to detect objects, such as single shot detector (SSD) and you only look once (YOLO). In contrast, two stage detectors utilize two networks to perform a coarse-to-fine detection, such as region-based convolutional neural network (R-CNN) and faster R-CNN. However, there is rare research focusing on face mask detection.

The basic steps of this system are:

- Preprocessing
- Weighted Training
- Object Recognition

1.2 Project Profile

- **Title:** Face Mask Detection System
- **Domain:** Deep Learning
- **Language:** Python
- **Library:** TensorFlow and Keras

1.3 Contributions

The major contributions of this system are:

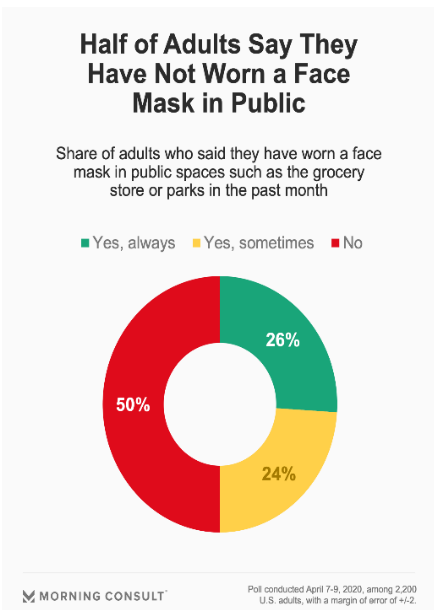
- Develop a training model that uses deep learning methods to learn the dataset's face mask features.
- An output unit that generates the data whether there is a facemask or not.
- An IOT unit to display the message.

Chapter 2

2 Problem Definition and Methodology

2.1 Problem definition

To stop the spreading of the contagious corona virus.



Wearing face masks is recommended as part of personal protective equipment and as a public health measure to prevent the spread of coronavirus disease 2019 (COVID-19) pandemic. Their use, however, is deeply connected to social and cultural practices and has acquired a variety of personal and social meanings.

According to survey, we can see how this has become a problem by seeing the picture mentioned below.

The large scale of losses that have been noticed across the world due to the COVID-19 pandemic have been highly shocking that leads to a lot of loss of property and life. The pandemic was sudden where the people and governments could not prepare themselves effectively beforehand to mitigate the effects of this epidemic. This virus is highly deadly and has caused multiple casualties which could be prevented through effective preventive measures.

Therefore use of mask enables effective prevention and further spread of the virus which can be the main ingredient for stopping the infections in their path.

1. To ensure that the mask rule is been followed, there needs to be an automatic technique that provide highly accurate intelligent system for mask detection through image processing.

2. Wearing a mask in public settings is an effective way to keep communities safe. As a response to the COVID -19 pandemic, we open-sourced a face mask detection application that uses AI to detect if people are wearing masks or not. We focused on making our face mask detector ready for real-world applications, such as CCTV cameras, where faces are small, blurry, and far from the camera.

2.1 Objectives

To propose a method that will help the authorities and officials to recognize persons who are not wearing a face mask, that is by:

1. Enforcing the mandate for wearing masks in public places following the COVID-19 pandemic.
2. Effectively providing a working model for accurate mask detection.
3. Utilizing image processing approaches to identify the presence of a mask on face.
4. Developing an efficient computer vision based system focused on the real time automated monitoring of people to detect face mask in public places.

2.3 Motivation

The model can identify people who are not wearing masks, and when it does, a message is sent to the officials in charge of the system.

2.4 Solution strategies and the best solution adopted

2 Solution strategies:

Existing Strategy:

It is required that someone, such as security or a volunteer, guides a person entering a public area if the person is not wearing a mask. The current approach is completely manual, and if someone is caught without a face mask, they are told to put one on.

There is no alternative for it. Covid-19 can occasionally spread when someone enters a public area without alerting anyone to it. There'll be two issues,

- No one will be there for controlling the passing of people without mask every moment and thus spreading of COVID- 19 will be easy.

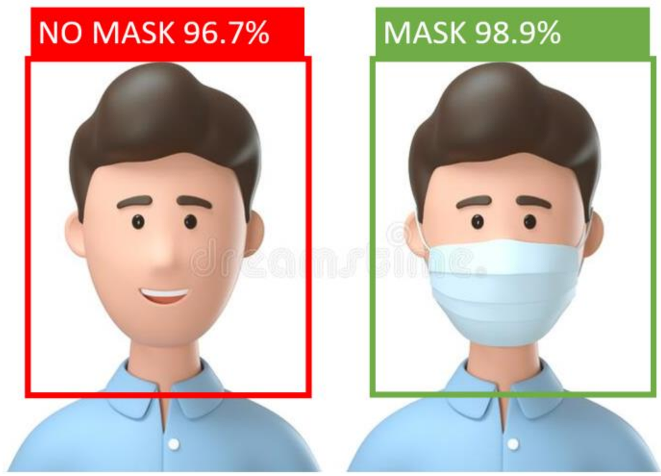
- ➢ Manual method is expensive.

Proposed Strategy:

An automated system that uses a camera to record video, identify somebody wearing no mask, and then notify the appropriate authority of the incident.

- ➢ It is possible to train the images and techniques by using CNN. So, it can be the best solution adopted.

FRONT END:



2.5 Scope

The system is easy to operate and it can be used in crowded areas. It also ensures the compliance for wearing mask and the system provides accurate assessment of the individual in public areas, whether the person is wearing a mask or not.

- Public will face the camera to get detected.
- Detect COVID-19 face masks in images.
- Detect face masks in real-time video streams.

Our goal is to train a custom deep learning model to detect whether a person is or is not wearing a mask. Video captioning, which aims to automatically detect face mask of a person and with further research a message can be sent to the concerned authority if the mask is missing. This will help to prevent the spreading of the pandemic Covid-19 and it can be used variety of public gathering areas such as,

- Theaters
- Seminar Halls
- Shopping Malls
- Education Institutions
- Convention centers

2.6 Methodology

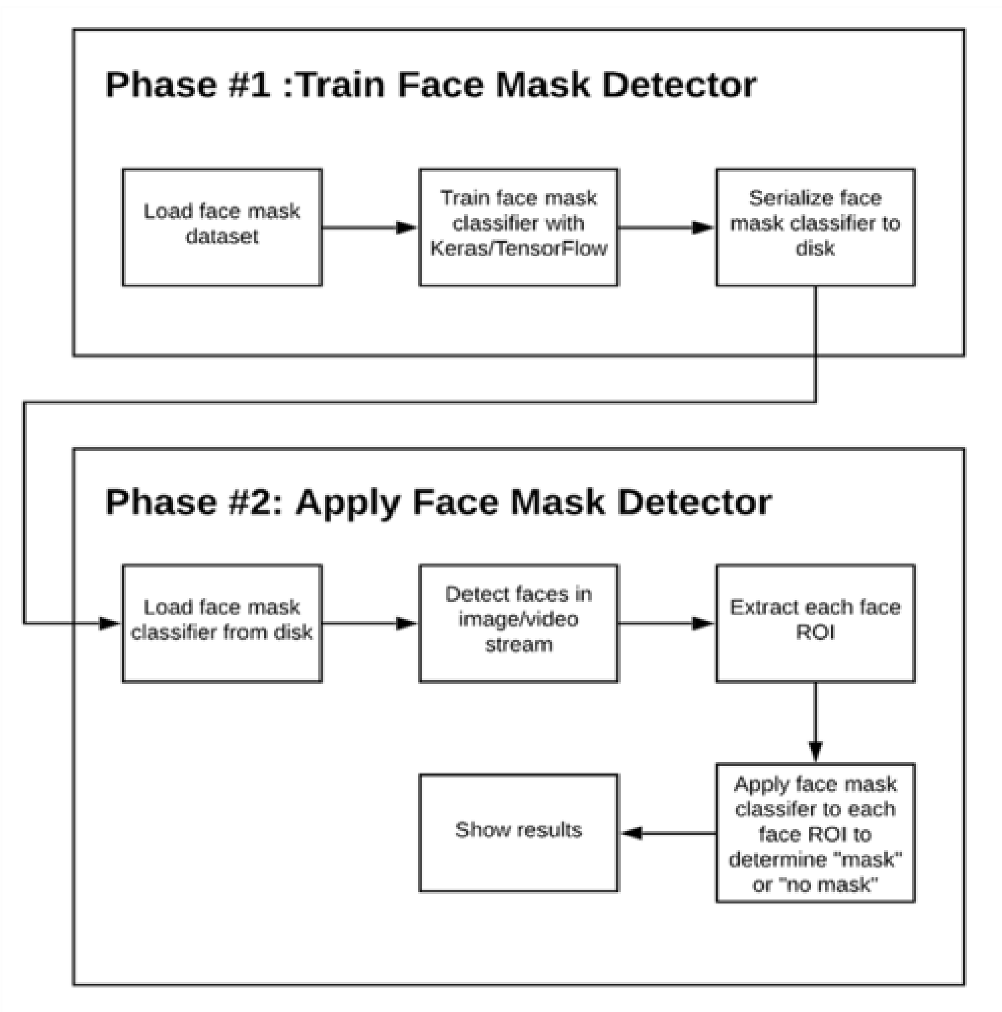
Deep learning Technique is used in this system with CNN. Convolutional Neural networks are used for image extraction. The system focuses on detecting a person wearing a mask on the video stream with the aid of computer vision and deep learning algorithms by using tensorflow, OpenCV, Keras library. The development of the system consists is divided into two main sections:

1. Train deep learning model
2. Apply facemask detector on video stream.

Chapter 3

3 System Design

3.1 Flow Diagram



3.2 Gantt Chart Flow Diagram

Tasks \ Weeks	1	2	3	4	5	6	7	8	9	10	11
Information Gathering											
Requirement Analysis											
preprocessing											
Training the images											
Face Mask detection model											
Testing the model											

3.3 CNN (Convolutional Neural Network)

CNN is a Convolutional Neural Network or CNN is a type of artificial neural network, which is widely used for image/object recognition and classification. CNN takes input as a two-dimensional array and works directly on the images rather than concentrating on feature extraction which other neural networks emphasize on.

3.4 Required Packages

3.4.1 *Keras*

Keras is a neural network API. It is a library written specifically in python. Also, it acts as an interface for the TensorFlow library which makes deep learning convenient. All the layers used in the CNN model are implemented using Keras. Along with the conversion of the image data to the binary class matrix in data processing, it helps to compile the overall model.

3.4.2 *TensorFlow*

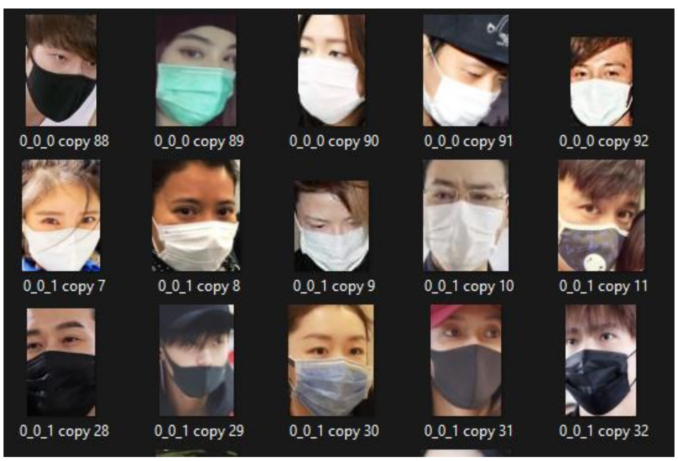
TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It is used in voice recognition, geographic information extraction, computer vision, text summarization, information retrieval, etc. in this project the CNN architecture uses TensorFlow as backend. It is also used to reshape the data (image) in the data processing.

3.4.3 *OpenCV*

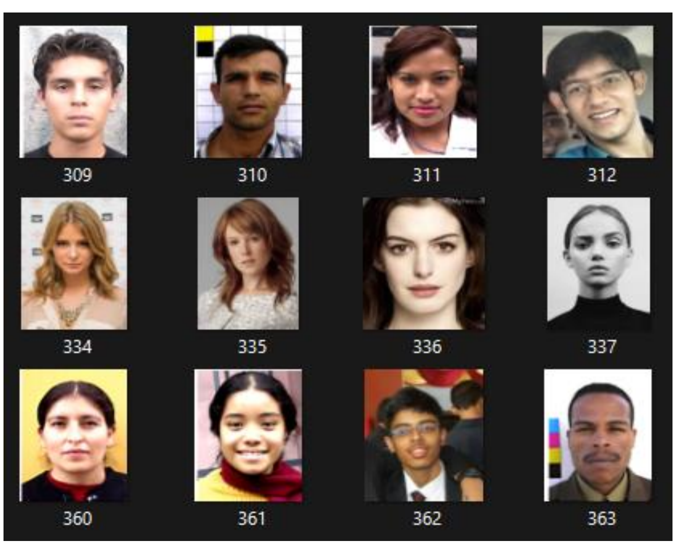
OpenCV (Open Source Computer Vision Library), an open source computer vision and ML software library, is applied to distinguish and identify faces, objects, group movements in recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken utilizing flash, find comparative pictures from an image database, etc. in this project OpenCV is used for resizing and color conversion of data images.

3.5 Dataset

The dataset was organized through google images and Kaggle. Two datasets were built labeled with_mask and without_mask. There are 800 masked faces and 750 unmasked faces. A training image generator was used for data augmentation to create more images with alteration making the dataset larger with around 1900 images on each dataset.



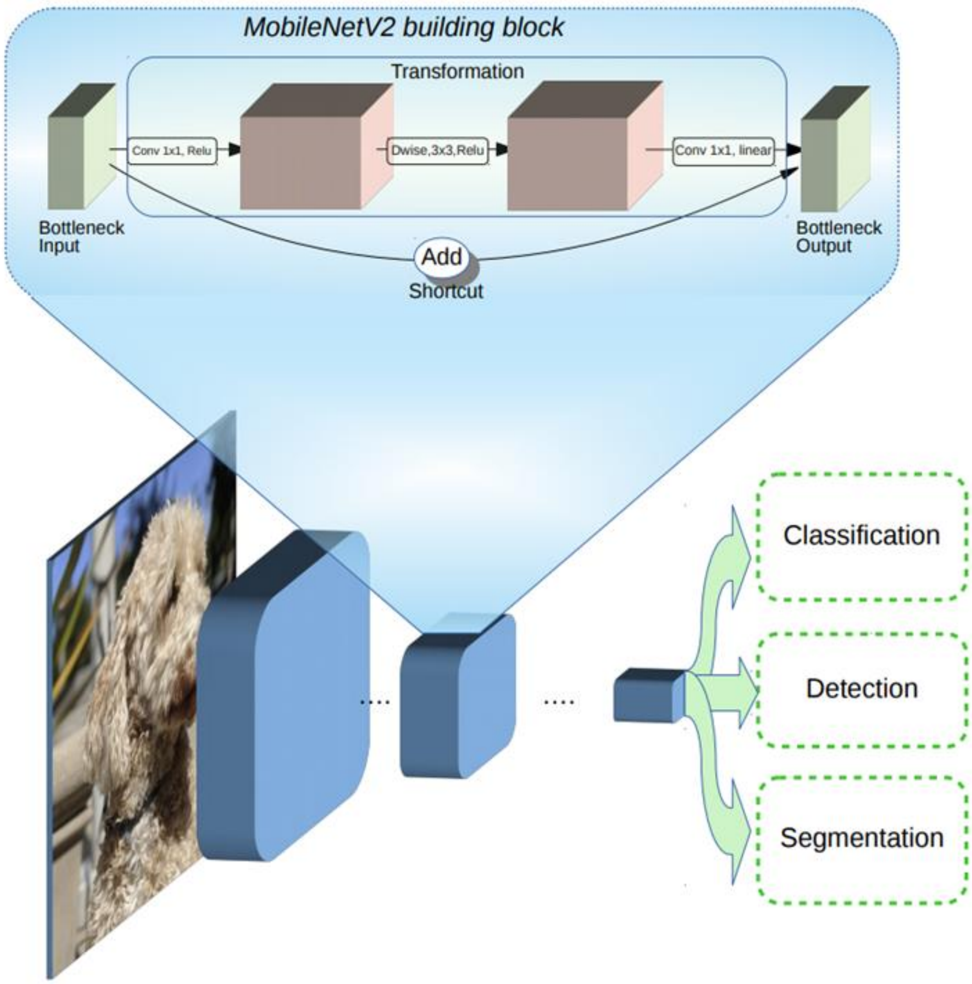
With_Mask



Without_Mask

3.6 Mobilenet-v2

MobileNet-v2 is a convolutional neural network that is 53 layers deep. We can load a pre-trained version of the network trained on more than a million images from the ImageNet database. The pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images, which is ideal for object detection. The network has an image input size of 224-by-224.



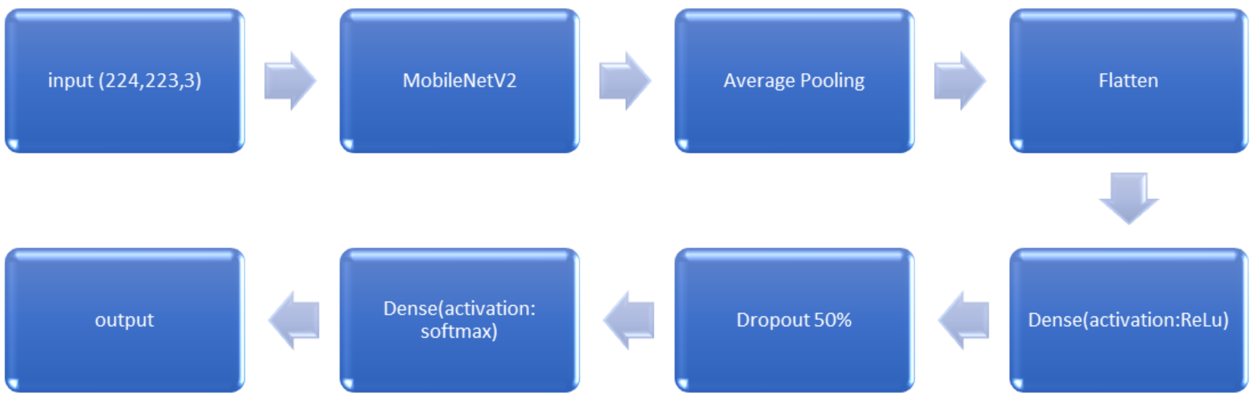
Chapter 4

4 Implementation

4.1 Training the face detector model

The software was developed using python. Along with TensorFlow and keras other python packages like sklearn, matplotlib and numpy was used.

The initial learning rate was 1e-4 and EPOCH set to 20 and batch size of 32.



Process flow of the train model

4.2 Splitting data

We divided the dataset into a training set and test set. CNN model will be trained using the training set and the model will be tested using the test set.

We assigned 80% of the total data for training and 20% for testing.

4.3 Data Augmentation

In order to expand the data set to include a larger number of images for training.

We rotate, zoom and tilt each image in the data set. A training image generator was used for data augmentation to create more images with alteration making the dataset larger with around 1900 images on each dataset.

4.4 Building the model

Here with the use of mobilenetV2 the base model was loaded using the pertained imagenet function of mobilenetv2.

```
baseModel = MobileNetV2(weights="imagenet", include_top=False,
        input_tensor=Input(shape=(224, 224, 3)))
```

the head model is constructed by placing it on top of the base model. We used AveragePooling2D, Flatten, Dense and Dropout. We used “softmax” activation function in the last dense layer.

```
headModel = baseModel.output
headModel = AveragePooling2D(pool_size=(7, 7))(headModel)
headModel = Flatten(name="flatten")(headModel)
headModel = Dense(128, activation="relu")(headModel)
headModel = Dropout(0.5)(headModel)
headModel = Dense(2, activation="softmax")(headModel)
model = Model(inputs=baseModel.input, outputs=headModel)
```

-

for compilation we use “Adam” optimizer and “binary_crossentropy” as our loss function. A loss function is a function that compares the target and predicted output values; measures how well the neural network models the training data. When training, our goal is to minimize this loss between the predicted and target outputs. An optimizer helps us to reduce the losses.

```
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
model.compile(loss="binary_crossentropy", optimizer=opt,
        metrics=["accuracy"])
```

4.5 Training the model

In this section the images are trained using the model and keras library. EPOCH was set to 20 which gave the optimum results.

```
H = model.fit(  aug.flow(trainX, trainY, batch_size=BS),
                steps_per_epoch=len(trainX) // BS,
                validation_data=(testX, testY),
                validation_steps=len(testX) // BS,
                epochs=EPOCHS)
```

4.6 Result of the training model

	precision	recall	f1-score	support
with_mask	0.99	0.99	0.99	383
without_mask	0.99	0.99	0.99	384
accuracy			0.99	767
macro avg	0.99	0.99	0.99	767
weighted avg	0.99	0.99	0.99	767

4.7 Building and implementing the facemask detector using the trained model

Our aim in this section is to use the webcam of our device and give result according to the trained model. Here the mechanism is to create single snaps of images from the video and compare it with the trained model to give the result during the live stream.

With the help of OpenCV we extract the face region of interest.

Convert it from BGR to RGB channel ordering, resizing it to 224x224, and preprocess it.

```
face = frame[startY:endY, startX:endX]

                                    face = cv2.cvtColor(face, cv2.COLOR_BGR2RGB)

                                    face = cv2.resize(face, (224, 224))

                                    face = img_to_array(face)

                                    face = preprocess_input(face)
```

We then load our serialized face detector model from disk. As it is running a face detector model it can detect multiple faces.

```
prototxtPath = r"deploy.protext"

weightsPath = r"res10_300x300_ssd_iter_140000.caffemodel"
```

faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)

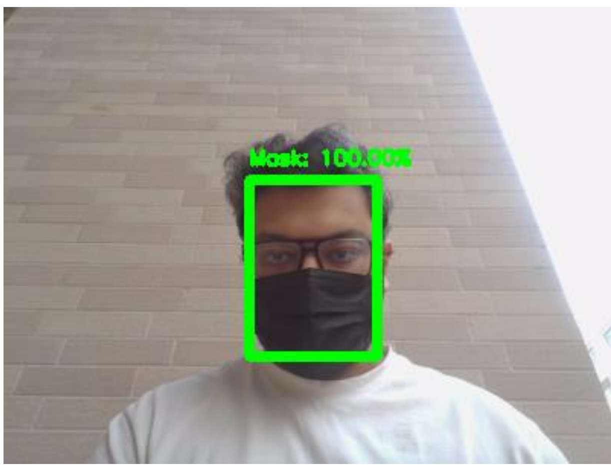
maskNet = load_model("mask_detector.model")

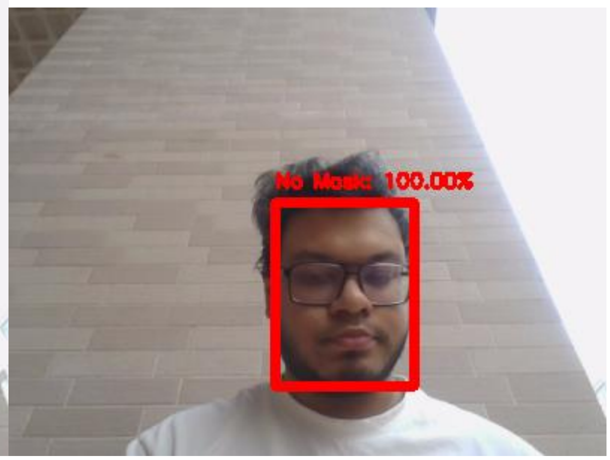
then we initialize the video stream and run an infinite loop from the stream

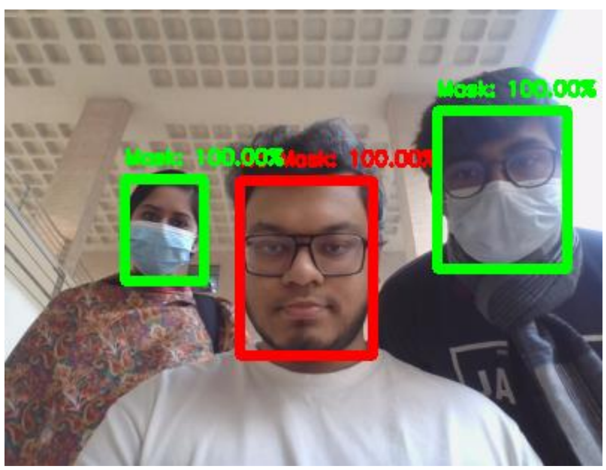
vs = VideoStream(src=0).start()

labes and probabilities are included in the video stream which detects if person is wearing mask or not.

4.8 Final Output







Chapter 5

5 Budgeting

Coefficient<Effort Factor>: 2.4 [where, P =1.05 & T =0.38]

SLOC = 250 Lines

Effort = Person Months, PM = $\mathbf{2.4} \times (250/\mathbf{1000})^\wedge\mathbf{1.05}$ = 0.56

Development Time = DM = 2.50 × (0.56)^0.38 = 7.994 ≈ 2 Months = 420 Working Hour

[Working hours per day = 7 hours]

Required number of People = 4

Per Developer Salary Per Working Hour = 100 Taka

Developer Salary in 2 months

Total Developer Salary = 100 ×420 × 4 = 1,68,000 Taka

Requirement Analysis:

Time needed: 1 week (3 Working Days = 21 Working Hour)

Total Requirement Analysis Expense = 200 × 21 = 4,200 Taka

Transportation Cost Estimation: 2,000 Taka

Training & Hardware Expenses Estimation: 5,000 Taka

Rent Expenses:

Room rent Per Month 12,000 Taka

Total in 2 Months = 24,000 Taka

Total Utilities in 2 Months: 6,000 Taka

Maintenance (Till 3 Months after Delivery):

Expense Per Hour = 1000 taka

Total Estimated Time Needed for Maintenance = 30 Hours

Total Estimated Maintenance Expense = 30 × 1000 = 30,000 Taka

Total Estimated Expense: 1,68,000 + 4,200 + 2,000 + 5,000 + 24,000 +6,000 +30,000

= 2,39,200 Taka

Profit:

20% of Total Estimated Expense = 2,39,200 × 20% = 47,840 Taka

Project Budget: 2,39,200 + 47,840 = 2,87,040 Taka

Chapter 6

6 Conclusion

As one of the best systems that will be implemented together with the interface of the alarm and alerting system in the future, the current ongoing system is recognized by MobileNet-v2 classifier. This system will be combined with the system for enforcing social distance to create a comprehensive system that can have a significant impact on the spread of. The demand for masks will be extremely high in the new world since it will have no faces, which poses serious security risks. According to experts, by CNN, wearing a face mask is the greatest way to stop the transmission of airborne viruses like Corona. However, this possesses a serious security risk to the country because it would present a huge opportunity for those who hide their faces for evil purposes. Experts also warn that the widespread use of masks could make it more difficult to solve crimes in the future because facial recognition is crucial for locating culprits. When all the components of our architecture are combined, the mask observation system usually works as intended. In this system, the MobileNetV2 classifier is used with Adam optimizer. The system's performance as a result of its ability to detect face masks in images with several faces from various perspectives.

We now have a face mask detector, which may help the public health care system. Low computation scenarios and the MobileNetV2 classifier serve as the framework for the architecture. Our face mask detection is based on the CNN (Convolutional neural network) model, and to determine if a person is wearing a mask or not, we use OpenCV, TensorFlow, Keras, and Python. The model was image-tested, and the results were accurate in real time video stream.

6.1 Future scope

When the pandemic covid-19 getting over, then this system comes into play for chemical factories, bank, glass factories etc. If a person enters the bank while wearing a mask he would be not allowed to enter and also if the person does not wear masks in glass factories chemical factories and etc. then the person would not be allowed to enter to the industry. A mind concept of human being has been proved out to be very good at recognizing familiar faces and facial recognizing familiar faces and facial recognition algorithms are getting better in identifying pattern. So thus, this challenge would create a scope to new face detection algorithms which can identify aces which are covered with greater accuracies and precisions. Moreover, the door of further scopes opens when we will use more CNN models other than Mobilenet-v2 such as, DenseNet-121, VGG-19, ResNet50, etc.

Chapter 7

7 Reference

1. W.H.O., “Coronavirus disease 2019 (covid-19): situation report, 205” 2020.
2. 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.rs-28668/v1.
3. A. Nieto-Rodríguez, M. Mucientes, V.M. Brea System for medical mask detection in the operating room through facial attributes Pattern Recognition. Image Anal. Cham (2015), pp. 138-145, 10.1007/978-3-319-19390-8_16
4. “Face Mask Detection”, Kaggle.com, 2020. [Online]. Available: https://www.kaggle.com/andrewmvd/face-mask-detection. 2020.
5. “TensorFlow White Papers”, TensorFlow, 2020. [Online].
6. Opencv-python-tutroals.readthedocs.io. 2020. Changing Color spaces — OpenCV-Python.
7. Hands on machine learning with scikit-learn and TensorFlow 2nd edition
8. M. Hashemi, “Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation”
9. Brownlee, J. (2020, October 18). *Softmax Activation Function with Python*. Machine Learning Mastery. https://machinelearningmastery.com/softmax-activation-function-with-python/
10. *7 Types of Activation Functions in Neural Network | Analytics Steps*. (n.d.). Www.analyticssteps.com. https://www.analyticssteps.com/blogs/7-types-activation-functions-neural-network
11. Xin, M., Wang, Y. Research on image classification model based on deep convolution neural network. J Image Video Proc. 2019, 40 (2019).
12. Image Classification Using Convolutional Neural Networks Deepika Jaiswal, Sowmya.V, K.P.Soman
13. D. Meena and R. Sharan, “An approach to face detection and recognition,” 2016 International Conference on Recent Advances and Innovations in Engineering
14. “Guide to the Sequential model - Keras Documentation”, Faroit.com, 2020.
15. R. Yamashita, M. Nishio, R. Do and K. Togashi, “Convolutional neural networks: an overview and application in radiology