A Project Report

on

FACE MASK DETECTION

Submitted in partial fulfillment of the

requirement for the award of the degree of

Bachelor of Technology in Computer Science and Engineering

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. Soumalya Ghosh
Assistant Professor
Department of Computer Science and Engineering

Submitted By

189SCSE1010056 – MANAV SAXENA

19SCSE1010266 – HIMANSHU SINGH

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA

DECEMBER - 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the project, entitled " FACE MASK DETECTION " in partial fulfillment of the requirements for the award of the BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of JULY-2021 to DECEMBER-2021, under the supervision of Mr.Soumalya Ghosh, Assistant Professor, Department of Computer Science and Engineering of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the project has not been submitted by me/us for the award of any other degree of this or any other places.

19SCSE1010056-MANAV SAXENA

19SCSE1010266-HIMANSHU SINGH

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor

(Mr.Soumalya Ghosh,
Assistant Professor)

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of 19SCSE1010056– MANAV SAXENA, 19SCSE1010266 – HIMANSHU SINGH has been held on _____ and his/her work is recommended for the award of BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date:

Place:

Table of Contents

FACE MASK DETECTION

ABSTRACT

After the predictions of the new (3^{rd}) wave of one of the deadliest corona virus could be spread in the nation pretty soon so wearing the mask is now very essential while going outside or to meet somebody. However there exists so many irresponsible fellows who refuses to wear the mask with many excuses . So, developing the face mask detector is very-very important in this type of scenarios. The project basically aims to develop the face mask detector which will be able to detect the any kind of masks . In order to detect whether the person wear face mask or not deep learning is used as the mask detection algorithm. Results have been done in the real – time – application. Basically it can detect if the person is wearing a mask or not using the cctv camera with quite a good accuracy . After that they denied access to places where the authorities has strictly asked to wear the face mask . After denied access to person , authorities will get the alert email in real-time , where the person's photo can or cannot be attached . In a way screen panel could be installed at the entrance where the person when denied access can see a pop up warning message where he/she is advised to wear the mask before getting the access .

This kind of face mask application can be useful at the airports , hotels , universities , public buildings , DL offices or many other places where we expect the large public gathering . The face mask detector that we designed could not only detect that whether a person has wearing mask or not during the entry time but it also able to detect that in any particular area in how much time a person could remove the mask and if the time surpasses to the time we fix then the warning again send to the person .

For example ; we had fixed that if a person remains without mask for more than 5 minutes in one hour . So the warning will send to the person and the mail will be send to the concerned authority with the proof of image of that person .

We use CNN Algorithm in the proposed system . Collecting number of data sets with face mask and without face mask . We could get accuracy depending on collecting the number of the images . Convolutional Neural Network is neural network that has one or more convolutional layers and mainly used for image processing , classification , segmentation and for other auto correlated data because of its high accuracy .

We use OpenCV (that is computer vision framework helps to do all sorts of processing on the images and videos . It is a widely used tool for image processing task .

) , Keras framework (because it is neural network based deep learning model) , python language , Tensorflow (end to end ,open source software for Artificial Intellenge and Machine Learning , create large-scale neural network , mainly used for the classification; perception;understanding;prediction;creation) as backend .

Also , we will able to detect that how many such detectors should we need in any particular area.

For example ; one detector will cover 500 meter distance and the total area is of 2km .

Then we have needed the 4 such detectors to cover that entire area.

LIST OF TABLE :

1. STUDENTS DATA :

STUDENT' NAME	ADMSISSION NO .	ENROLLMENTNO.	SIGN
MANAV SAXENA	19SCSE1010056	19021011264	
HIMANSHUSINGH	19SCSE1010266	19021011455	

2 . FACULTY DATA :

FACLTY NAME
MR . SOUMALYA GHOSH

INTRODUCTION

Face mask detection refers to detect whether a person is wearing a mask 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 [6], [7]. 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 [8], [9], [10], [11] 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 a develop technique that can accurately detect mask 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 45,000 images, our technique achieves outstanding accuracy of 98.2%. The major contribution of the proposed work is given below:

1.

Develop a novel object detection method that combines one-stage and two-stage detectors for accurately detecting the object in real-time from video streams with transfer learning at the back end.

2.

Improved affine transformation is developed to crop the facial areas from uncontrolled real-time images having differences in face size, orientation and background. This step helps in better localizing the person who is violating the facemask norms in public areas/ offices.

3.

Creation of unbiased facemask dataset with imbalance ratio equals to nearly one.

4.

The proposed model required low memory, making it easily deployable for embedded devices use for surveillance purposes.

LITERATURE SURVEY

STUDY OF MASKED FACE DETECTION APPROACH IN VIDEO ANALATICS :

Description :

The paper outlines the principle used in each of these steps and the use of commonly available algorithm of people detection and face detection . This unique approach for the problem has created a simple in complexity thereby making real time implementation feasible .
Analysis of the algorithm’s performance on the test video sequences gives useful insights to further improvements in the marked face detection performance .

Limitions :

The approach requires further analysis for the improvement in the performance of the technique.

A CASCADE FRAMEWORK FOR MASKED FACE DETECTION

DESCRIPTION :

They propose a new CNN-based cascade framework , which consists of three carefully designed Convolutional neural networks to detect masked faces . Besides because of the shortage of the masked face . Besides , because of the shortage of masked faced training samples , a new dataset called “ MASKED FACED DATASET ” to fine tune our CNN models .

The evaluation proposed masked face detection algorithm on the Masked face testing set , And it achieves the satisfactory performance .

LIMITATION :

The main limitation is that the author have not utilize the real time morphing technique for the corresponding pixels .

UTILIZING SKIN MASK AND FACE ORGANS DETECTION FOR IMPROVING THE VIOLA FACE DETECTION METHOD :

DESCRIPTION :

Approach to detect eyes and more makes possible face detecting only with two eyes or one eye and nose , in addition to decrease the false detection rate rather than the viola approach . Implementing of this approach shows , the presented algorithm could reduce the false negative From 10 percent to 2.4 percent and reduce false positive from 4.4 percent to 2.4 .

LIMITATION :

There is an increased computational complexity that is observed in this research .

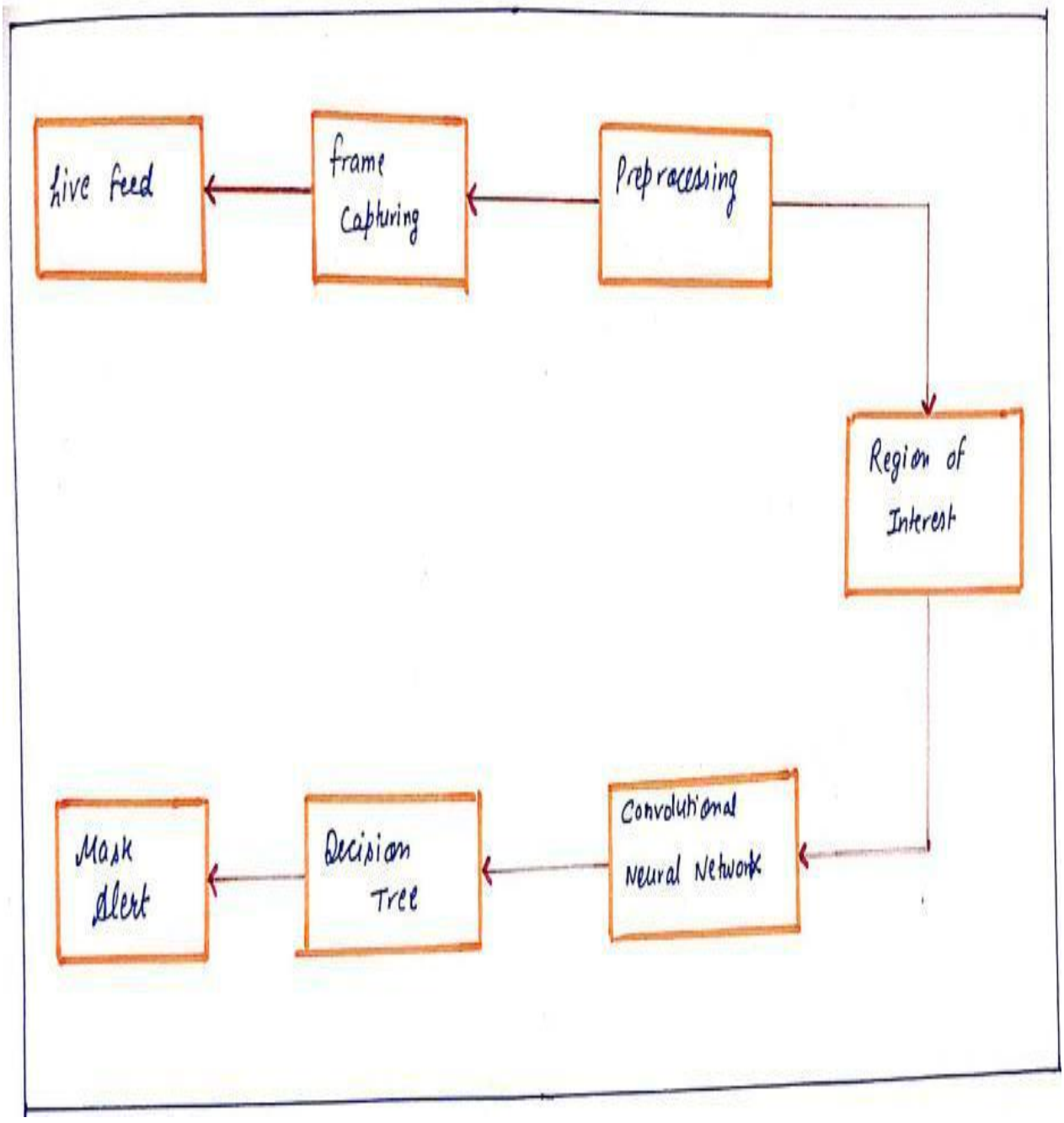
DATA FLOW DIAGRAM :

DATA FLOW DIAGRAM (DFD) represent flow of the data and transformation in Face Mask Detection (FSD) System.

System Features :

- Setting the face mask detection
- Records of person with or without mask
- Secure Real-Time Records
- Count the violators in the real-time
- Count the non-violators in the real-time

SYSTEM ARCHITECTURE



LIST OF FIGURES

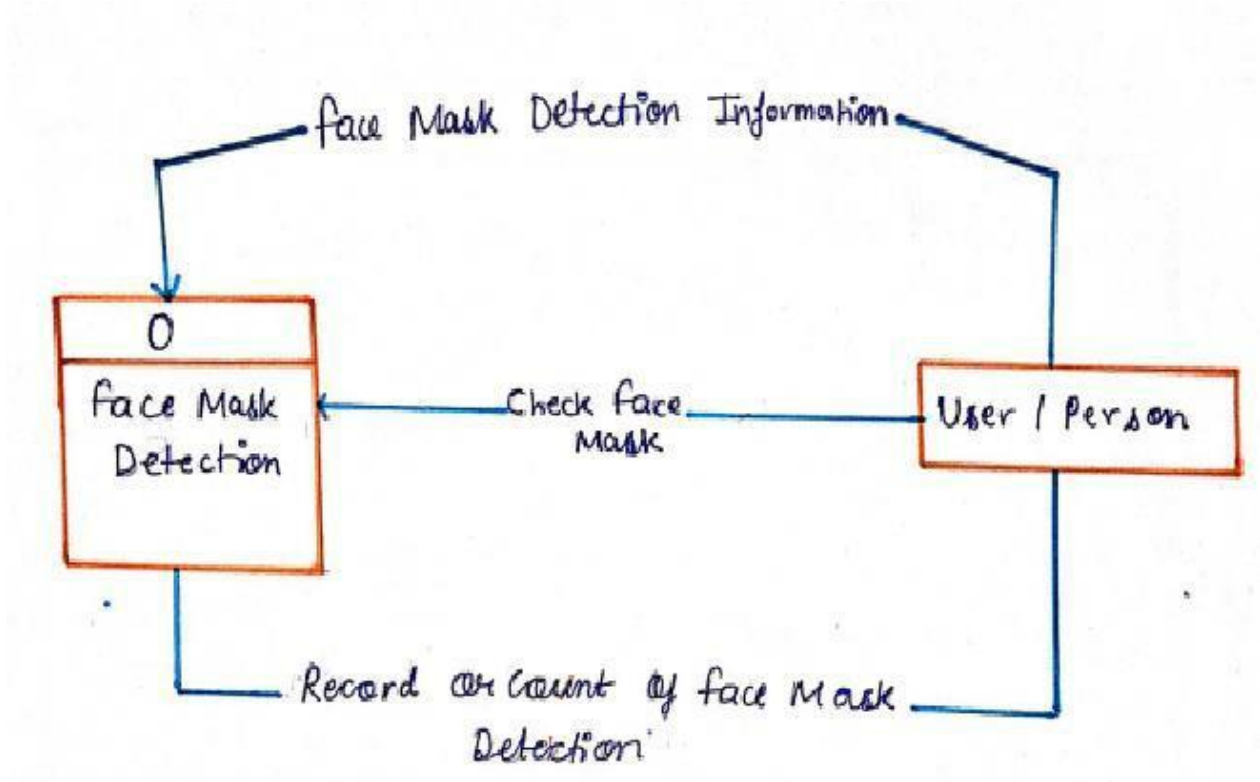
DFD (LEVEL – 0)

The face mask detection system (FMDS) level 0 , it is also known as the context diagram .

It is supposed to be as an abstract view , with the mechanism that represented as the single process with external parties.

It depicts the overall structure of FMD as a single bubble , it consist of incoming and outgoing indicator showing input and output data.

It is the general process works as a initial guide for the overall system.



DFD (LEVEL – 1)

The content of FMDS DFD level 1 must be single process node from the context diagram and is broken down into sub processes . System must display and reveal further processing information . These are essential data for accomadate by the system.

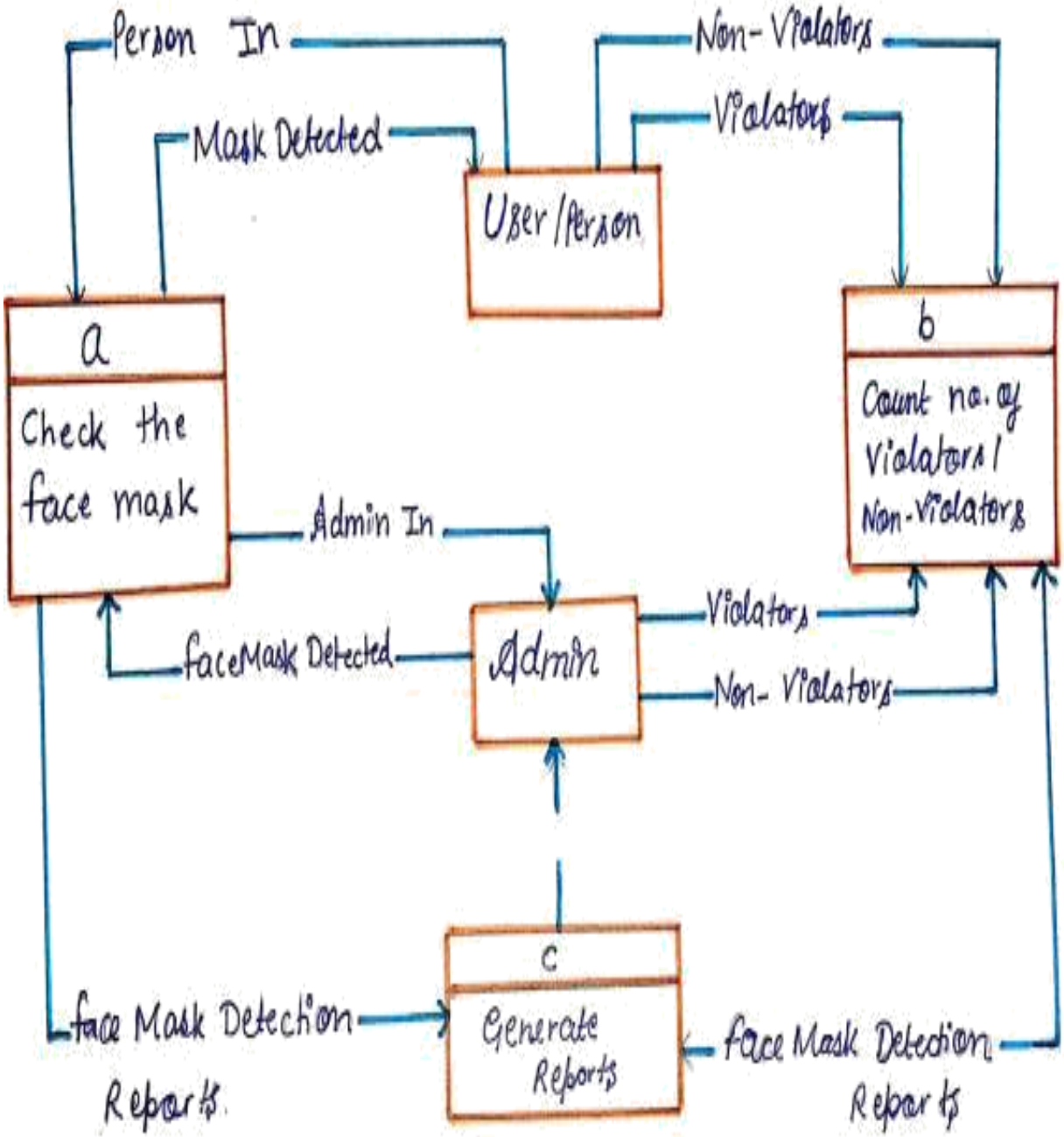
Procedures :

- FMD records
- Violator records
- Non-Violator Records

It depicts the overall structure of FMD as a single bubble , it consist of incoming and outgoing indicator showing input and output data.

Process work :

1. Records of person with or without mask
2. Total count of individual enters in the firm as without mask
3. It is the basis for admin to manage the FMDS.


Person In
Mask Detected
Non-Violators
Violators
User/Person
a
Check the face mask
b
Count no. of Violators/ Non-Violators
Admin In
FaceMask Detected
Admin
Violators
Non-Violators
c
Generate Reports
face Mask Detection Reports
face Mask Detection Reports

DFD (LEVEL – 2)

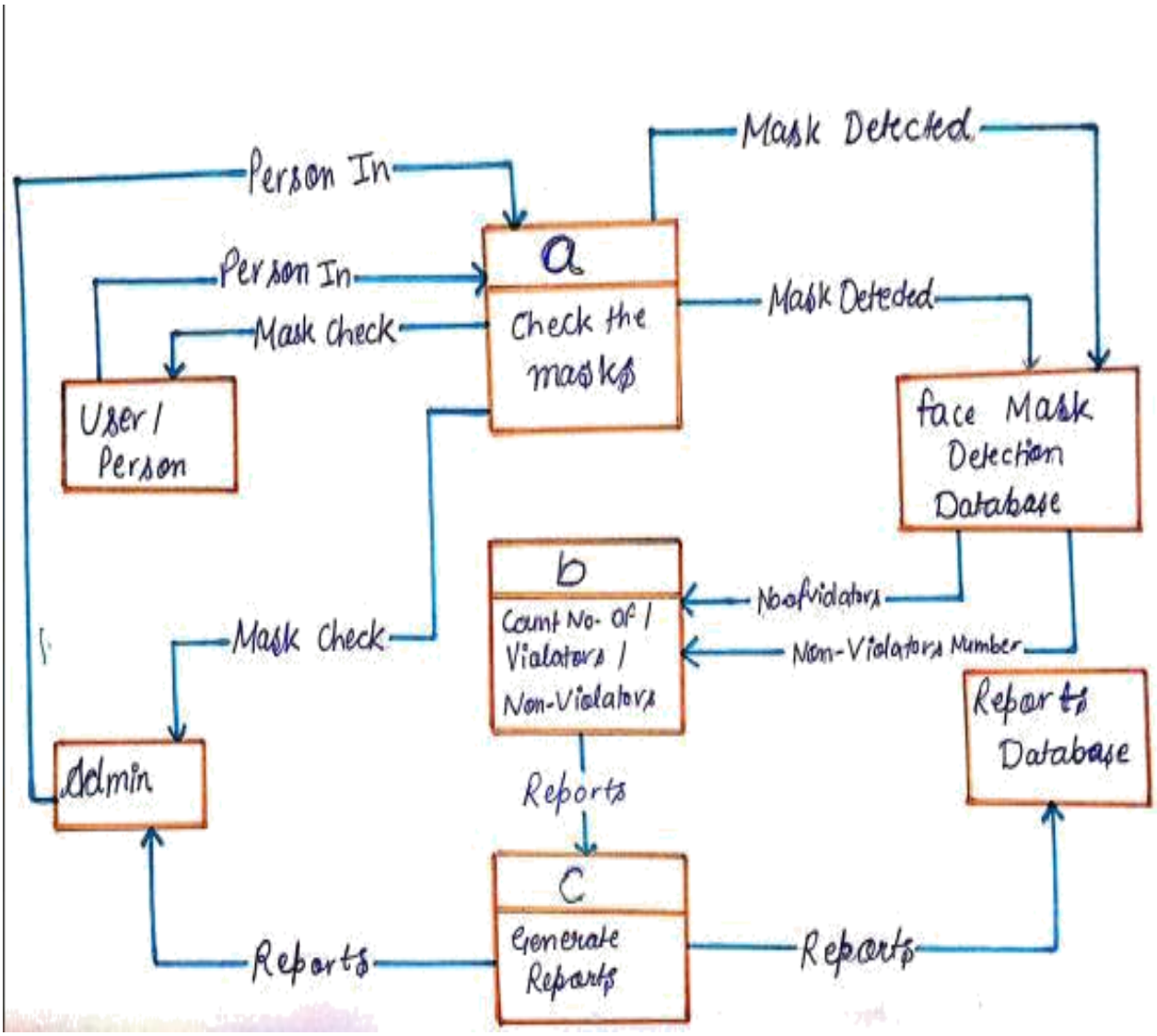
The system should represent the basic modules as well as data that flows between them .

DFD level-2 is the highest abstraction level of the system so the processes must be detailed that is based on the DFD level .

Here the user will know how the request being processed

Processed Involved :

- Setting face mask protection
- Records of person with mask
- Records of person without mask
- Real-Time records
- Count of violators in real time
- Count of non - violators in real time


Person In
Mask Detected
Person In
a
Check the masks
Mask Detected
Mask Check
User / Person
face Mask Detection Database
b
Count No. of / Violators / Non-Violators
No of violators
Non-Violators Number
Mask Check
Admin
Reports
Reports Database
Reports
c
Generate Reports
Reports

ABOUT THE ENHANCED WORK :

If we have a detector that detect or covers some distance let's say (y). And we want to know to cover a whole particular area, how much detectors we need, So it could be generated as :-

So; let's we have a x km of area. We want to know how many detectors we need and we stated that

① machine covered y m distance.

①st step is to convert the total area of x km into metres.

So; (x' = x × 1000 m)

(X' = area of particular place in metres)

(ii). now, no. of machines required is the round off of $\left(\frac{X'}{y}\right)$

No. of machines = roundoff $\left(\frac{X'}{y}\right)$

EXAMPLE – 1

eg. 1 detector machine covered 500 m distance and we have the total area of 4km

Now, we want to know how many detector (machines) required to cover the area;

So; (i). Total area = 4 km $\Rightarrow$ (X)

$\therefore X' = 4 \times 1000$

$X' = 4000\ m$

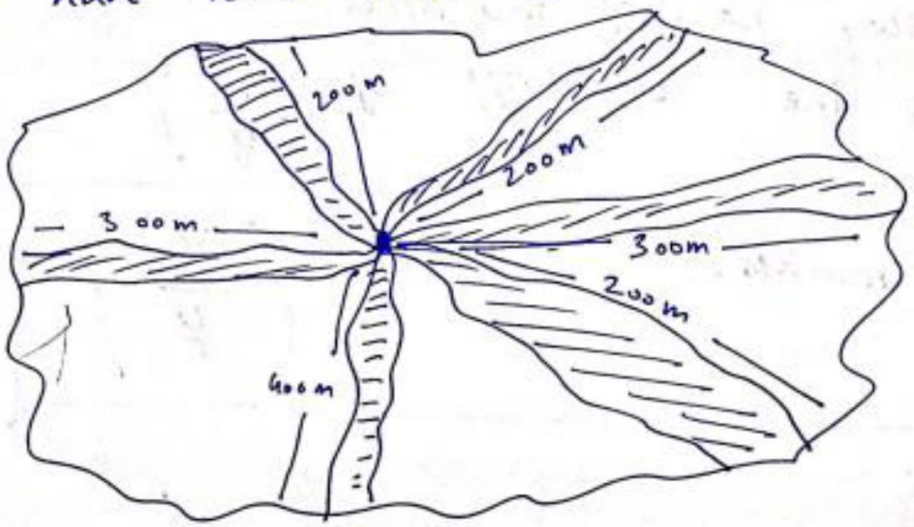
(ii). No. of machines = $\frac{X'}{4}$

($y = 500$ m); $\hookrightarrow \frac{4000^{8}}{500}$ machines

8 machines.

EXAMPLE – 2

(b) 1 detector machine covered 200 m distance we have total area of 1.5 km :



Total area = 1.5 km. (x)
(1 machine covered 200 m.)

(i) $x = 1.5$ km , $y = 200$ m

$x' = 1.5 \times 1000$

$(x' = 1500 \text{ m})$

(ii) No. of machines $\Rightarrow$

roundoff $\left(\frac{1500}{200} \right)$

roundoff (7.5)

↳ 8. machines.

So; (we need 8 machines to cover that area.)

It is restricted that if one person is at a particular area for one hour and if he/she removes mask for more than 5 minutes; if they remove mask for more than 5 minutes in particular area, so the, warning would sent to the person, and the mail will send to the concerned authority.

How to calculate the above scenerio:-

60 min → 5 min.
30 min → 2.5 min
15 min → 1.25 min
2.5 min → 0.625 min

Here; we knowss that person can remove mask for 5 min. in 1 hrs not more than that;

1 hour → 3600 sec.
5 min → 300 sec.

① We have to find in 1 min for how much time a person can remove mask.

1 min → 60 sec; x → seconds for how much a person can remove mask.

$$3600 \times x = 300 \times 60$$

$$x = \left(\frac{\cancel{3}\cancel{0}\cancel{0} \times 60}{\cancel{3}\cancel{6}\cancel{0}\cancel{0}}\right) \quad \boxed{x = 5 \text{ seconds}}$$

Now; if a person stays in a particular area for 50 minutes, so for how many minutes he/she could remove the mask. And after that the warning would be sent!!!

50 minutes → 3000 second

(1 sec → 0.083 sec. allowed.)

(3000 seconds) → 3000 × 0.083 sec.

→ (249 seconds) allowed.

3000 * 0.083 < 249 seconds.

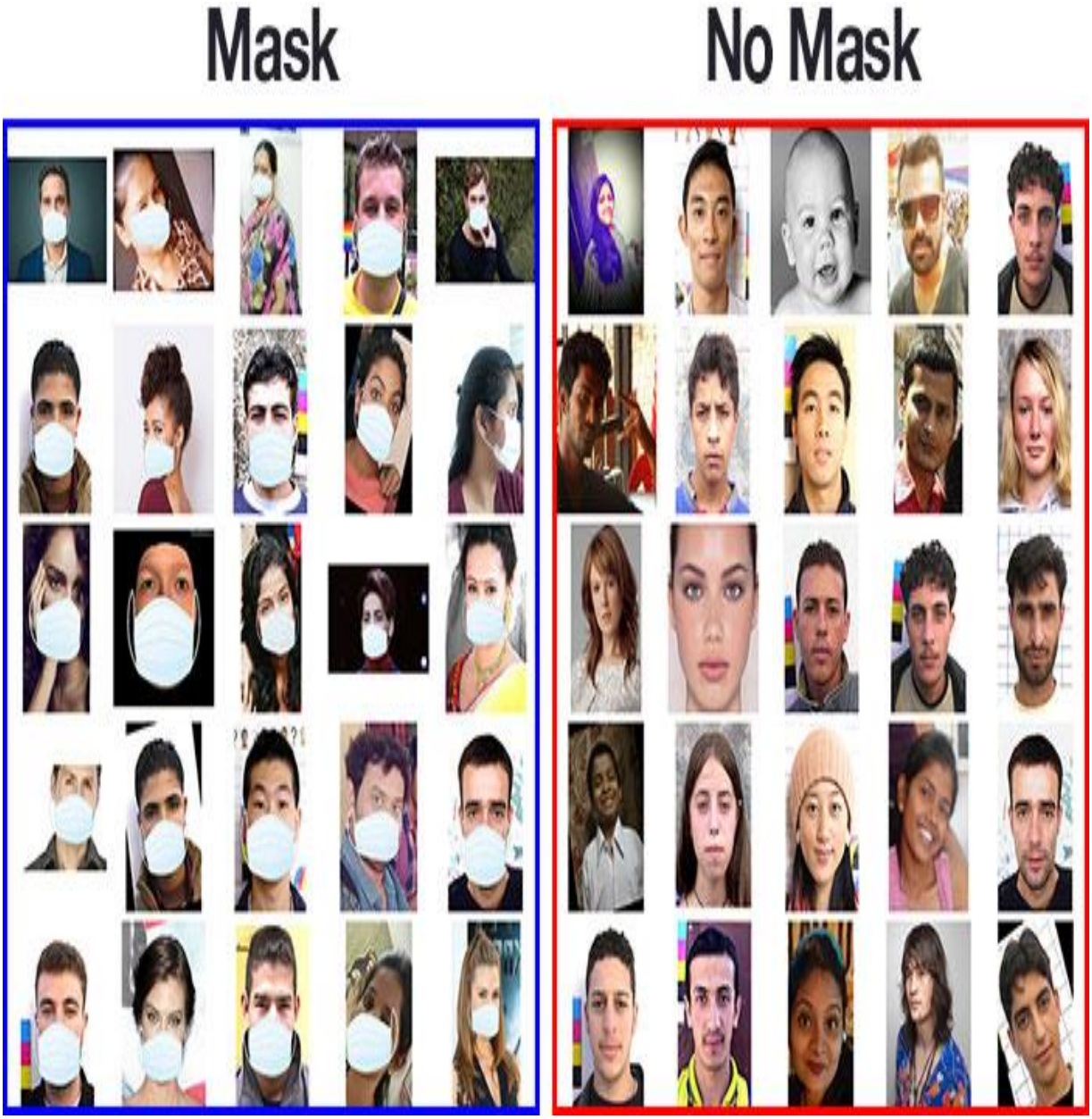
Here; x * 0.083 < y seconds

[x → for how much time the person stay in a particular area (second)]

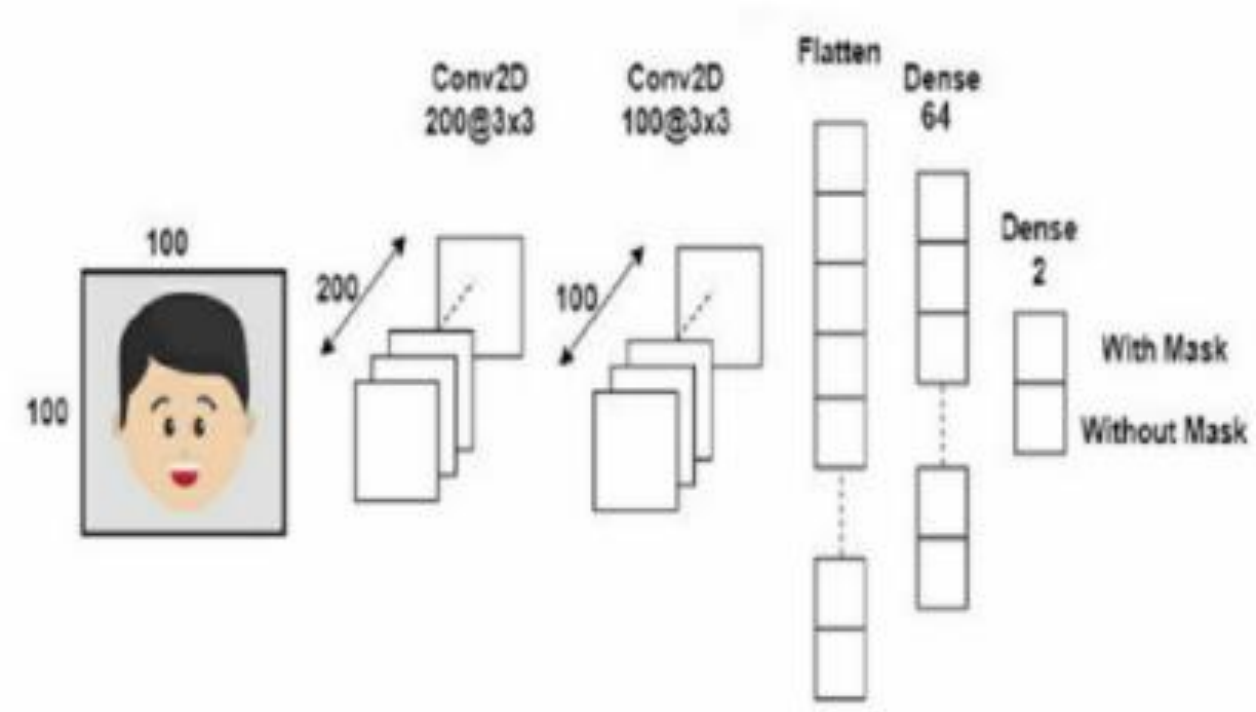
[y (second) → for how much time it is allowed to the person to remove mask]

[0.083 → the value in 1 second (it is for 5 minutes in 1 hrs) (second.)]

Dataset



CNN Architecture



OpenCv

OpenCV is a multi-platform library that we can use to develop real-time computer vision applications. It focuses on image processing, video capture and analysis which includes features such as face detection and object detection. In this tutorial, we explain how to use OpenCV in your applications. OpenCV is a Python open source library, used for computer-assisted visualization, Machine Learning, Face Recognition, etc.

In OpenCV, a CV is a type of computer vision, defined as a field of study that helps computers to understand the content of digital images such as photos and videos.The purpose of computer vision is to understand the content of images. Extracts a picture from a picture, possibly an object, a text description, and a three-dimensional model, and so on. For example, cars can be simplified with a computer view, which will be able to identify various objects around the road, such as robots, pedestrians, road signs, etc., and do the right thing.

Keras

Keras is an open source library of state-of-the-art Neural Network, written via Python capable enough to work on Theano, TensorFlow, or CNTK. Created by one of Google's engineers, Francois Chollet. It is designed to be user-friendly, flexible, and a module to aid rapid exploration through deep neural networks. It not only supports Convolutional Networks and Continuous Networks individually but also a combination of them.Can't handle low-level statistics, so it uses the Backend library to solve it. The back library serves as a high-level API wrap for low-level API, allowing us to work on TensorFlow, CNTK, or Theano.

TensorFlow

TensorFlow is an open source machine learning framework for all developers. It is used to implement machine learning and deep learning applications. To develop and research interesting ideas in the field of artificial intelligence, the Google team created TensorFlow. Since TensorFlow was developed in the Python programming language, it is considered an easy-to-

understand framework. The
TensorFlow tutorial is intended for beginners and experts alike. This guide introduces you to all the basic and advanced concepts of machine learning and deep learning concepts such as deep
neural networks, image processing, and sentiment analysis.
TenorFlow is one of the most popular deep
learning frameworks developed by the Google team. This is a free and open source software library developed in the Python
programming language. This tutorial is designed to help you implement deep learning projects with TensorFlow easily and efficiently.

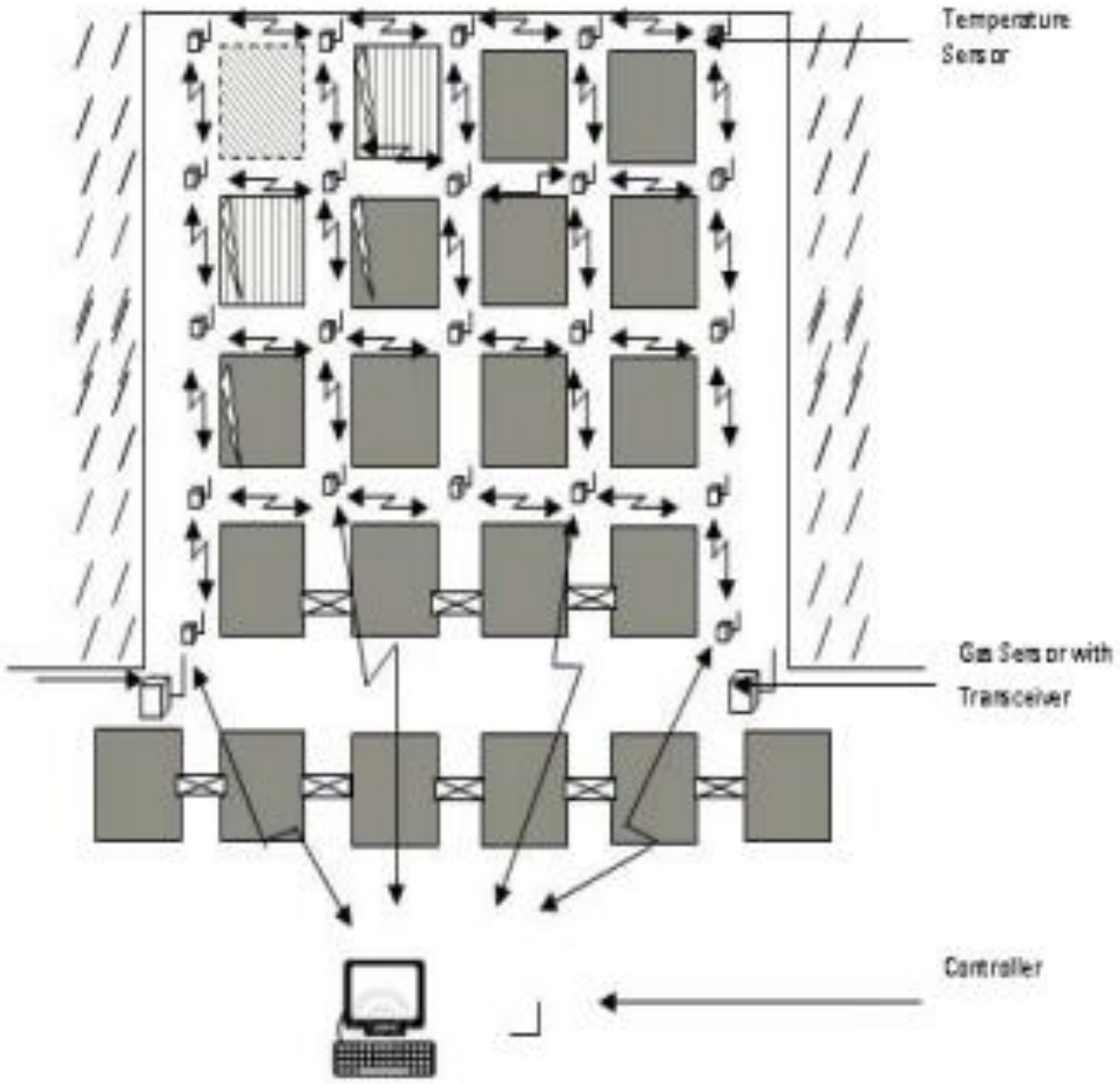
Algorithm

Algorithm 1: Face Mask Detection

Input: Dataset including faces with and without masks
Output: Categorized image depicting the presence of face mask

```
for each image in the dataset do
    Visualize the image in two categories and label them
    Convert the RGB image to Gray-scale image
    Resize the gray-scale image into 100 x 100
    Normalize the image and convert it into 4 dimensional array
end
for building the CNN model do
    Add a Convolution layer of 200 filters
    Add the second Convolution layer of 100 filters
    Insert a Flatten layer to the network classifier
    Add a Dense layer of 64 neurons
    Add the final Dense layer with 2 outputs for 2 categories
end
Split the data and train the model
```

Working Plan



INSTALLED LIBRARIES :

```
from tensorflow.keras.applications.mobilenet_v2 import
preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
import os
```

COMPLETE SOURCE CODE :

```
# import the necessary packages
from tensorflow.keras.applications.mobilenet_v2 import
preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.models import load_model
from imutils.video import VideoStream
import numpy as np
import imutils
import time
import cv2
```

```
import os

def detect_and_predict_mask(frame, faceNet, maskNet):
# grab the dimensions of the frame and then construct a blob
# from it
(h, w) = frame.shape[:2]
blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),
(104.0, 177.0, 123.0))

# pass the blob through the network and obtain the face detections
faceNet.setInput(blob)
detections = faceNet.forward()
print(detections.shape)

# initialize our list of faces, their corresponding locations,
# and the list of predictions from our face mask network
faces = []
locs = []
preds = []

# loop over the detections
for i in range(0, detections.shape[2]):
# extract the confidence (i.e., probability) associated with
# the detection
confidence = detections[0, 0, i, 2]

# filter out weak detections by ensuring the confidence is
# greater than the minimum confidence
if confidence > 0.5:
# compute the (x, y)-coordinates of the bounding box for
# the object
box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
(startX, startY, endX, endY) = box.astype("int")
```

```
# ensure the bounding boxes fall within the dimensions of
# the frame
(startX, startY) = (max(0, startX), max(0, startY))
(endX, endY) = (min(w - 1, endX), min(h - 1, endY))

# extract the face ROI, convert it from BGR to RGB channel
# ordering, resize 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)

# add the face and bounding boxes to their respective
# lists
faces.append(face)
locs.append((startX, startY, endX, endY))

# only make a predictions if at least one face was detected
if len(faces) > 0:
# for faster inference we'll make batch predictions on *all*
# faces at the same time rather than one-by-one predictions
# in the above `for` loop
faces = np.array(faces, dtype="float32")
preds = maskNet.predict(faces, batch_size=32)

# return a 2-tuple of the face locations and their corresponding
# locations
return (locs, preds)

# load our serialized face detector model from disk
prototxtPath = r"face_detector\deploy.prototxt"
weightsPath =
r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
```

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

# load the face mask detector model from disk
maskNet = load_model("mask_detector.model")

# initialize the video stream
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()

# loop over the frames from the video stream
while True:
# grab the frame from the threaded video stream and resize it
# to have a maximum width of 400 pixels
frame = vs.read()
frame = imutils.resize(frame, width=400)

# detect faces in the frame and determine if they are wearing a
# face mask or not
(locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)

# loop over the detected face locations and their corresponding
# locations
for (box, pred) in zip(locs, preds):
# unpack the bounding box and predictions
(startX, startY, endX, endY) = box
(mask, withoutMask) = pred

# determine the class label and color we'll use to draw
# the bounding box and text
label = "Mask" if mask > withoutMask else "No Mask"
color = (0, 255, 0) if label == "Mask" else (0, 0, 255)

# include the probability in the label
label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
```

```
# display the label and bounding box rectangle on the output
# frame
cv2.putText(frame, label, (startX, startY - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

# show the output frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

# if the `q` key was pressed, break from the loop
if key == ord("q"):
break

# do a bit of cleanup
cv2.destroyAllWindows()
vs.stop()
```

REFERENCES

www.google.com
www.researchgate.com
www.kaggle.com
www.youtube.com
www.javatpoint.com
www.tutorialspoint.com