In [28]:
```
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
import face_recognition
from datetime import datetime
```

In [29]:
```
# Mask Detector model
```

In [30]:
```
def detectMask(frame, faceNet, maskNet):

    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),(104.0, 177.0, 123.0))

    faceNet.setInput(blob)
    detections = faceNet.forward()
    print(detections.shape)

    faces = []
    locs = []
    preds = []


    for i in range(0, detections.shape[2]):

        confidence = detections[0, 0, i, 2]

        if confidence > 0.5:

            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")
            (startX, startY) = (max(0, startX), max(0, startY))
            (endX, endY) = (min(w - 1, endX), min(h - 1, endY))

            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)
            faces.append(face)
            locs.append((startX, startY, endX, endY))


    if len(faces) > 0:

        faces = np.array(faces, dtype="float32")
        preds = maskNet.predict(faces, batch_size=32)
    return (locs, preds)
```

In [31]:
```
pPath = r"face_detector\deploy.prototxt"
wPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceDetect = cv2.dnn.readNet(pPath, wPath)
```

In [32]:
```
maskDetect = load_model("maskdetector.model")
```

In [33]:
```
# Fetch Database images
```

In [34]:
```
from sqlalchemy import create_engine
engine = create_engine("mysql://root:@localhost/test",echo = True)
conn = engine.connect()
```

```
2022-03-16 14:58:48,971 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'sq
l_mode'
2022-03-16 14:58:48,975 INFO sqlalchemy.engine.base.Engine ()
2022-03-16 14:58:48,981 INFO sqlalchemy.engine.base.Engine SHOW VARIABLES LIKE 'lo
wer_case_table_names'
2022-03-16 14:58:48,985 INFO sqlalchemy.engine.base.Engine ()
2022-03-16 14:58:48,993 INFO sqlalchemy.engine.base.Engine SELECT DATABASE()
2022-03-16 14:58:48,997 INFO sqlalchemy.engine.base.Engine ()
2022-03-16 14:58:49,002 INFO sqlalchemy.engine.base.Engine show collation where `C
harset` = 'utf8mb4' and `Collation` = 'utf8mb4_bin'
2022-03-16 14:58:49,007 INFO sqlalchemy.engine.base.Engine ()
2022-03-16 14:58:49,016 INFO sqlalchemy.engine.base.Engine SELECT CAST('test plain
returns' AS CHAR(60)) AS anon_1
2022-03-16 14:58:49,019 INFO sqlalchemy.engine.base.Engine ()
2022-03-16 14:58:49,022 INFO sqlalchemy.engine.base.Engine SELECT CAST('test unico
de returns' AS CHAR(60)) AS anon_1
2022-03-16 14:58:49,024 INFO sqlalchemy.engine.base.Engine ()
2022-03-16 14:58:49,027 INFO sqlalchemy.engine.base.Engine SELECT CAST('test colla
ted returns' AS CHAR CHARACTER SET utf8mb4) COLLATE utf8mb4_bin AS anon_1
2022-03-16 14:58:49,029 INFO sqlalchemy.engine.base.Engine ()
```

In [35]:
```
fob=open(r'C:\Users\Dell\Desktop\Jainam docs\Deteccion de mascara\Recognition images
fob=fob.read()
data=('bunhakunj86@gmail.com',fob) # tuple with data
mail=conn.execute("INSERT INTO  images(email,img) VALUES (%s,%s)",data)
print("Row Added  = ",mail.rowcount)
```

```
2022-03-16 14:58:49,130 INFO sqlalchemy.engine.base.Engine INSERT INTO  images(ema
il,img) VALUES (%s,%s)
2022-03-16 14:58:49,132 INFO sqlalchemy.engine.base.Engine ('bunhakunj86@gmail.co
m', b'\xff\xd8\xff\xe1x9Exif\x00\x00II*\x00\x08\x00\x00\x00\x0c\x00\x00\x01\x04\x0
0\x01\x00\x00\x00\xf0\x0e\x00\x00\x01\x01\x04\x00\x01\x00\x00\x004\x0b\x ... (2386
024 characters truncated) ... 0\x00\x00\x00\x00\x01\nh\x00\x00\x00#\x00\x00\x00\x0
0\x00\xa1\nE\x00\x00\x00\x13\x00\x00\x00\x00\x00\x10\t2\x00\x00\x002\x00\x00\x000
\x00\x00\x00SEFT')
2022-03-16 14:58:49,284 INFO sqlalchemy.engine.base.Engine COMMIT
Row Added  =  1
```

In [36]:
```
fob=open(r'C:\Users\Dell\Desktop\Jainam docs\Deteccion de mascara\Recognition images
fob=fob.read()
data=('jainamkothari14@gmail.com',fob) # tuple with data
mail=conn.execute("INSERT INTO  images(email,img) VALUES (%s,%s)",data)
print("Row Added  = ",mail.rowcount)
```

```
2022-03-16 14:58:49,426 INFO sqlalchemy.engine.base.Engine INSERT INTO  images(ema
il,img) VALUES (%s,%s)
2022-03-16 14:58:49,430 INFO sqlalchemy.engine.base.Engine ('jainamkothari14@gmai
l.com', b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x01\x00`\x00`\x00\x00\xff\xdb\x
00C\x00\x03\x02\x02\x03\x02\x02\x03\x03\x03\x03\x04\x03\x03\x04\x05\x08\x05\ ...
(1438008 characters truncated) ... f\xfc\x04Uf\xe9\xff\x00\x02\xa2\x8a]@\xb3\xfc4
[\xff\x00\x1d\x14Q\xd4}P\xd6\xea)\xc3\xfdq\xa2\x8a}\x00\x07\xdd\xa7[\xff\x00\xad4Q
R"\xbd\x14QLg\xff\xd9')
2022-03-16 14:58:49,521 INFO sqlalchemy.engine.base.Engine COMMIT
Row Added  =  1
```

In [37]:
```
my_cursor=conn.execute("SELECT * FROM  images")
my_result=my_cursor.fetchall()
for row  in my_result:
    print(row)
    fob=open(r'C:\Users\Dell\Desktop\Jainam docs\Deteccion de mascara\Database Image
    fob=fob.write(row[1])
```

```
2022-03-16 14:58:49,689 INFO sqlalchemy.engine.base.Engine SELECT * FROM  images
2022-03-16 14:58:49,692 INFO sqlalchemy.engine.base.Engine ()
('bunhakunj86@gmail.com', b'\xff\xd8\xff\xe1x9Exif\x00\x00II*\x00\x08\x00\x00\x00
\x0c\x00\x00\x01\x04\x00\x01\x00\x00\x00\xf0\x0e\x00\x00\x01\x01\x04\x00\x01\x00\x
00\x004\x0b\x ... (2386024 characters truncated) ... 0\x00\x00\x00\x00\x01\nh\x00
\x00\x00#\x00\x00\x00\x00\x00\xa1\nE\x00\x00\x00\x13\x00\x00\x00\x00\x00\x10t2\x0
0\x00\x002\x00\x00\x000\x00\x00\x00SEFT')
('jainamkothari14@gmail.com', b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x01\x00`
\x00`\x00\x00\xff\xdb\x00C\x00\x03\x02\x02\x03\x02\x02\x03\x03\x03\x03\x04\x03\x03
\x04\x05\x08\x05\ ... (1438008 characters truncated) ... f\xfc\x04Uf\xe9\xff\x00\x
02\xa2\x8a]@\xb3\xfc4[\xff\x00\x1d\x14Q\xd4}P\xd6\xea)\xc3\xfdq\xa2\x8a}\x00\x07\x
dd\xa7[\xff\x00\xad4QR"\xbd\x14QLg\xff\xd9')
('bunhakunj86@gmail.com', b'\xff\xd8\xff\xe1x9Exif\x00\x00II*\x00\x08\x00\x00\x00
\x0c\x00\x00\x01\x04\x00\x01\x00\x00\x00\xf0\x0e\x00\x00\x01\x01\x04\x00\x01\x00\x
00\x004\x0b\x ... (2386024 characters truncated) ... 0\x00\x00\x00\x00\x01\nh\x00
\x00\x00#\x00\x00\x00\x00\x00\xa1\nE\x00\x00\x00\x13\x00\x00\x00\x00\x00\x10t2\x0
0\x00\x002\x00\x00\x000\x00\x00\x00SEFT')
('jainamkothari14@gmail.com', b'\xff\xd8\xff\xe0\x00\x10JFIF\x00\x01\x01\x01\x00`
\x00`\x00\x00\xff\xdb\x00C\x00\x03\x02\x02\x03\x02\x02\x03\x03\x03\x03\x04\x03\x03
\x04\x05\x08\x05\ ... (1438008 characters truncated) ... f\xfc\x04Uf\xe9\xff\x00\x
02\xa2\x8a]@\xb3\xfc4[\xff\x00\x1d\x14Q\xd4}P\xd6\xea)\xc3\xfdq\xa2\x8a}\x00\x07\x
dd\xa7[\xff\x00\xad4QR"\xbd\x14QLg\xff\xd9')
```

In [38]:
```
# Face Recognition
```

In [39]:
```
path=r"C:\Users\Dell\Desktop\Jainam docs\Deteccion de mascara\Database Images"
personimages=[]
emails=[]
imagelist=os.listdir(path)
print(imagelist)
```

```
['bunhakunj86@gmail.com.jpeg', 'jainamkothari14@gmail.com.jpeg']
```

In [40]:
```
for current_img in imagelist:
    current=cv2.imread(f'{path}/{current_img}')
#     print(current)
    personimages.append(current)
    name=os.path.splitext(current_img)
#     print(name[0])
    emails.append(name[0])

print(emails)
```

```
['bunhakunj86@gmail.com', 'jainamkothari14@gmail.com']
```

In [41]:
```
def faceEncodings(images):
    encodeList=[]
    count=0
    for img in images:
        print("Generating Encodings for Image No : ",count)
        count=count+1
        img=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode=face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
```

In [48]:

```python
def addLog(email):

    with open('Log.csv','r+') as f:
        myDataList = f.readlines()
        emailList = []

        for line in f:
            entry = line.split(',')
            emailList.append(entry[0])

        if email not in emailList:
            time_now = datetime.now()
            time= time_now.strftime('%H:%M:%S')
            date= time_now.strftime('%d/%m/%Y')
            f.writelines(f'\n{email},{time},{date}')
```

In [43]:

```python
enc=faceEncodings(personimages)
print("All Encodings completed : ")
```

```
Generating Encodings for Image No :  0
Generating Encodings for Image No :  1
All Encodings completed :
```

In [49]:

```
capture=cv2.VideoCapture(0)
while True:
    flag,frame=capture.read()
    if flag:
        frame=imutils.resize(frame,width=600)
        (locs, preds) = detectMask(frame, faceDetect, maskDetect)
        for (box, pred) in zip(locs, preds):

            (startX, startY, endX, endY) = box
            (mask, withoutMask) = pred

            if mask > withoutMask:
                label = "Mask"
            else:
                label="No Mask"

            if label == "Mask":
                color = (0, 255, 0)
                cv2.putText(frame, label, (startX, startY - 10), cv2.FONT_HERSHEY_SI
                cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)

            else :
                color = (0, 0, 255)
                faces=cv2.resize(frame,(0,0),None,0.25,0.25)
                faces=cv2.cvtColor(faces,cv2.COLOR_BGR2RGB)

                facesCurrentFrame = face_recognition.face_locations(faces)
                encodesCurrentFrame = face_recognition.face_encodings(faces,facesCur

                for encodeFace,faceLoc in zip(encodesCurrentFrame,facesCurrentFrame)

                    matches = face_recognition.compare_faces(enc,encodeFace)
                    faceDis = face_recognition.face_distance(enc,encodeFace)

                    matchIndex = np.argmin(faceDis)

                    if matches[matchIndex] :
                        name=emails[matchIndex].upper()
                        print(name)
                        text=label+" "+name
                        cv2.putText(frame, text, (startX, startY - 10), cv2.FONT_HER
                        cv2.rectangle(frame, (startX, startY), (endX, endY), color,
                        addLog(name)

        cv2.imshow("Live Video",frame)
        if cv2.waitKey(2) == 27:
            break
capture.release()
cv2.destroyAllWindows()
```

```
(1, 1, 200, 7)
(1, 1, 200, 7)
JAINAMKOTHARI14@GMAIL.COM
(1, 1, 200, 7)
JAINAMKOTHARI14@GMAIL.COM
(1, 1, 200, 7)
JAINAMKOTHARI14@GMAIL.COM
(1, 1, 200, 7)
JAINAMKOTHARI14@GMAIL.COM
(1, 1, 200, 7)
(1, 1, 200, 7)
JAINAMKOTHARI14@GMAIL.COM
(1, 1, 200, 7)
JAINAMKOTHARI14@GMAIL.COM
```

```
(1, 1, 200, 7)
JAINAMKOTHARI14@GMAIL.COM
```

In [ ]:

In [ ]: