Presentation on

FACE MASK DETECTION

OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINNERING

BY

PAWAN KUMAR
(2018UGEC040R)

ROHAN GUPTA
(2018UGEC043R)

SHUBHANSHU KUMAR
(2018UGEC037R)

Submitted to
Prof Dr. Rashmi Panda

Dept. of Electronics and communication Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, RANCHI

INTRODUCTION

First of all to understand my project you have to understand computer vision. Face tracking and detection features in sequence is an important and fundamental problem in computer vision. This area of research has a lot of applications in face identification systems, model based coding, gaze detection, human computer, interaction, teleconferencing, etc. human-computer interaction, teleconferencing, etc.

Open CV

OpenCV means Intel® Open Source Computer Vision Library. It is a collection of C functions and a few C++ classes that implement some popular Image Processing and Computer Vision algorithms. OpenCV has cross-platform middle-to-high level API that consists of a few hundreds C functions. It does not rely on external libraries, though it can use some when it is possible. OpenCV is free for both non-commercial and commercial use. OpenCV provides transparent interface to Intel Integrated Performance Primitives (IPP). That is, it loads automatically IPP libraries optimized for specific processor at runtime, if they are available.

Image Processing

Computer manipulation of images. Some of the many algorithms used in image processing include convolution (on which many others are based), FFT, DCT, thinning (or skeletonisation), edge detection and contrast enhancement. These are usually implemented in software but may also use special purpose hardware for speed. Image processing contrasts with computer graphics, which is usually more concerned with the generation of artificial images, and visualisation, which attempts to understand (real-world) data by displaying it as an artificial image (e.g. a graph). Image processing is used in image recognition and computer vision. Silicon Graphics manufacture workstations which are often used for image processing. There are a few programming languages designed for image processing, e.g. CELIP, VPL, C++.

Face Tracking

Face detection and tracking are important in video content analysis since the most important objects in most video are human beings. Research on face tracking and animation techniques has been improved due to its wide range of applications in security, entertainment industry, gaming, psychological facial expression analysis and human computer interaction. Recent advances in face video processing and compression have made face-to-face communication be practical in real world applications. However, higher bandwidth is still highly demanded due to the increasing intensive communication. Model based low bit rate transmission with high quality video offers a great potential to mitigate the problem raised by limited communication resources. However, after a decade's effort, robust and realistic real time face tracking and generation still pose a big challenge. The difficulty lies in a number of issues including the real time face feature tracking under a variety of imaging conditions such as lighting variation, pose change, self-occlusion and multiple non-rigid features deformation and the real time realistic face modeling using a very limited number of feature parameters. Traditionally, the head motion is modeled as a 3D rigid motion with the local skin deformation, the linear motion tracking method cannot represent the rapid head motion and dramatic expression change accurately.

DEVELOPMENT TOOLS

1. we make our tool with Anaconda
2. language: python
3.we use sublime text as our editor
4. web-Browser:
    google chrome /Microsoft Edge

System Requirements

On client side

. Operating system
. Web-Browser : google chrome /microsoft edge

On Server side

. Python

Features

1. An airtight matching algorithm

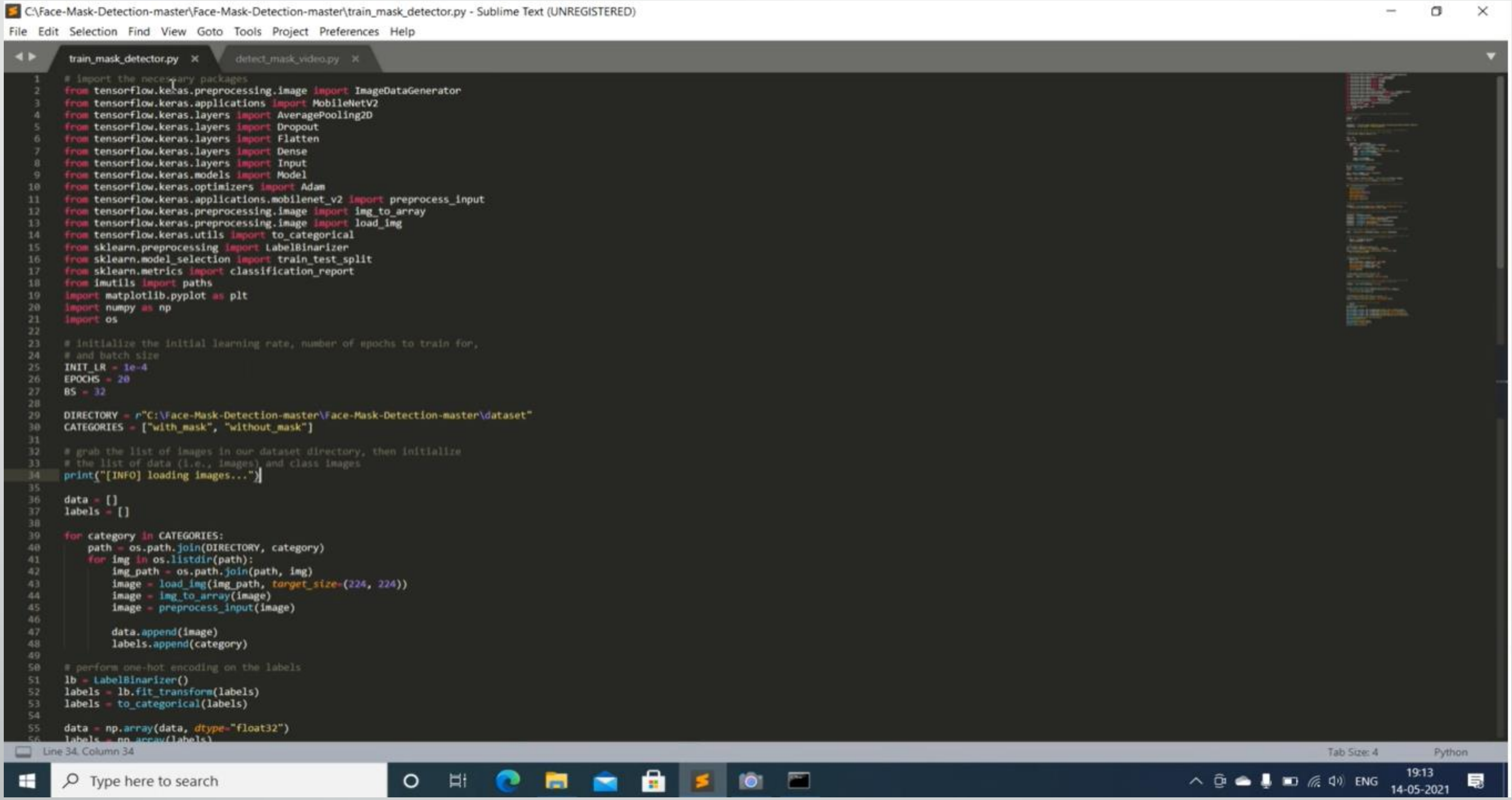
2.Scalability

3.Built-in privacy protection

4. Predictive Analytics

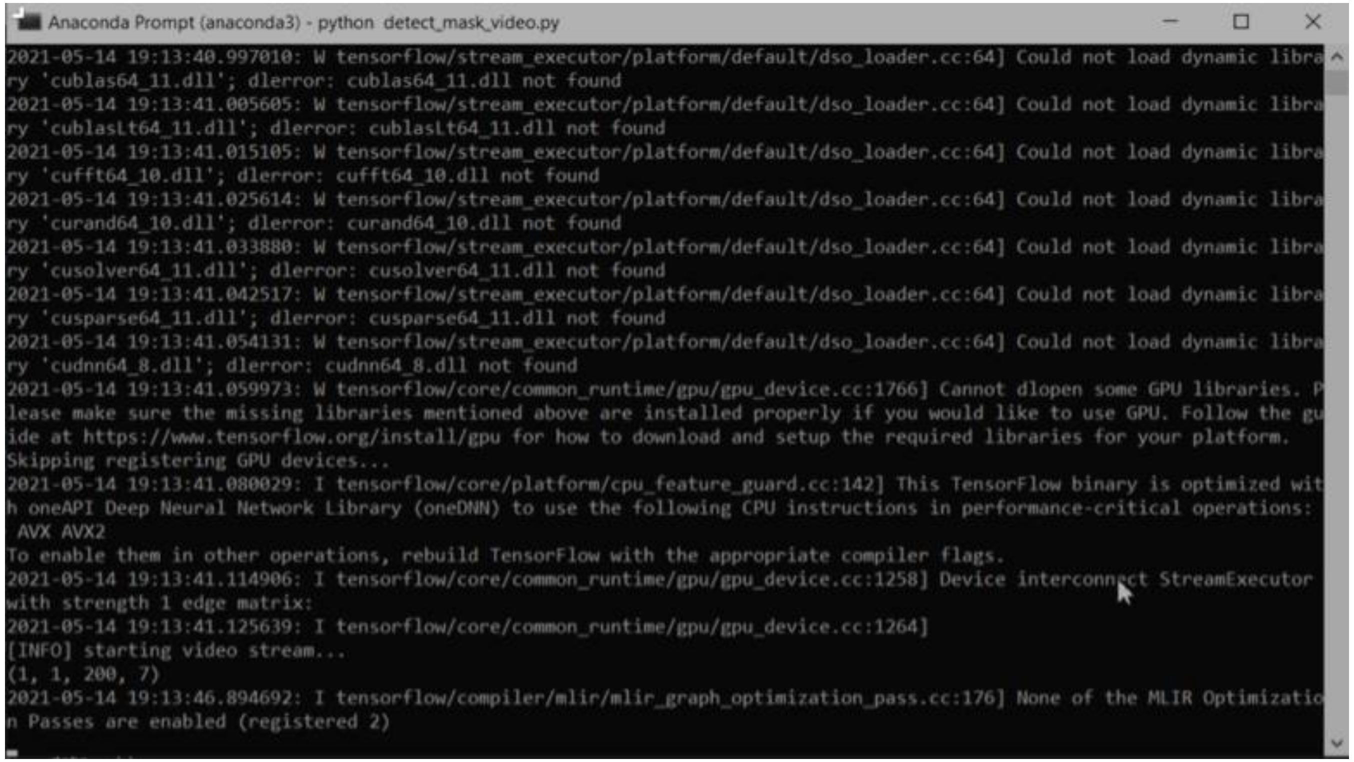
5. watchlist-as-a-sevice

Snapshots


C:\Face-Mask-Detection-master\Face-Mask-Detection-master\train_mask_detector.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
train_mask_detector.py
detect_mask_video.py
Line 34, Column 34
Tab Size: 4
Python
Type here to search
ENG
19:13
14-05-2021

```python
# import the necessary packages
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications import MobileNetV2
from tensorflow.keras.layers import AveragePooling2D
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Input
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.utils import to_categorical
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from imutils import paths
import matplotlib.pyplot as plt
import numpy as np
import os

# initialize the initial learning rate, number of epochs to train for,
# and batch size
INIT_LR = 1e-4
EPOCHS = 20
BS = 32

DIRECTORY = r"C:\Face-Mask-Detection-master\Face-Mask-Detection-master\dataset"
CATEGORIES = ["with_mask", "without_mask"]

# grab the list of images in our dataset directory, then initialize
# the list of data (i.e., images) and class images
print("[INFO] loading images...")

data = []
labels = []

for category in CATEGORIES:
    path = os.path.join(DIRECTORY, category)
    for img in os.listdir(path):
        img_path = os.path.join(path, img)
        image = load_img(img_path, target_size=(224, 224))
        image = img_to_array(image)
        image = preprocess_input(image)

        data.append(image)
        labels.append(category)

# perform one-hot encoding on the labels
lb = LabelBinarizer()
labels = lb.fit_transform(labels)
labels = to_categorical(labels)

data = np.array(data, dtype="float32")
labels = np.array(labels)
```

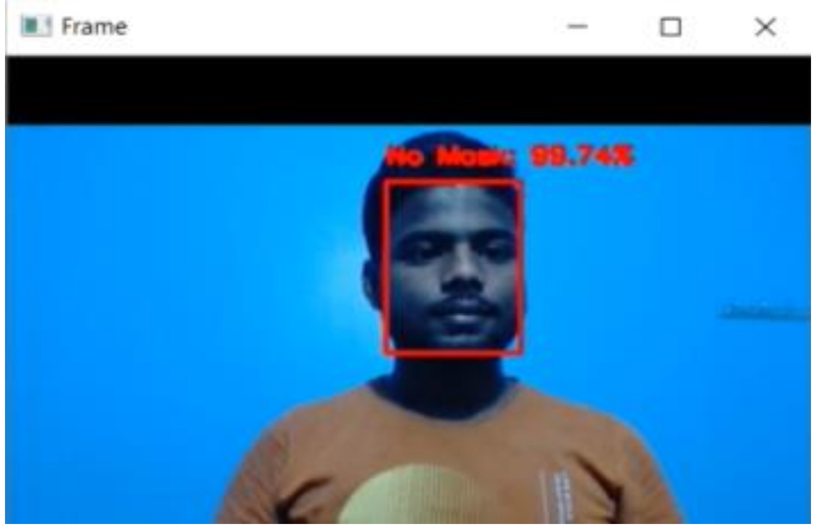
Codeusing sublime text

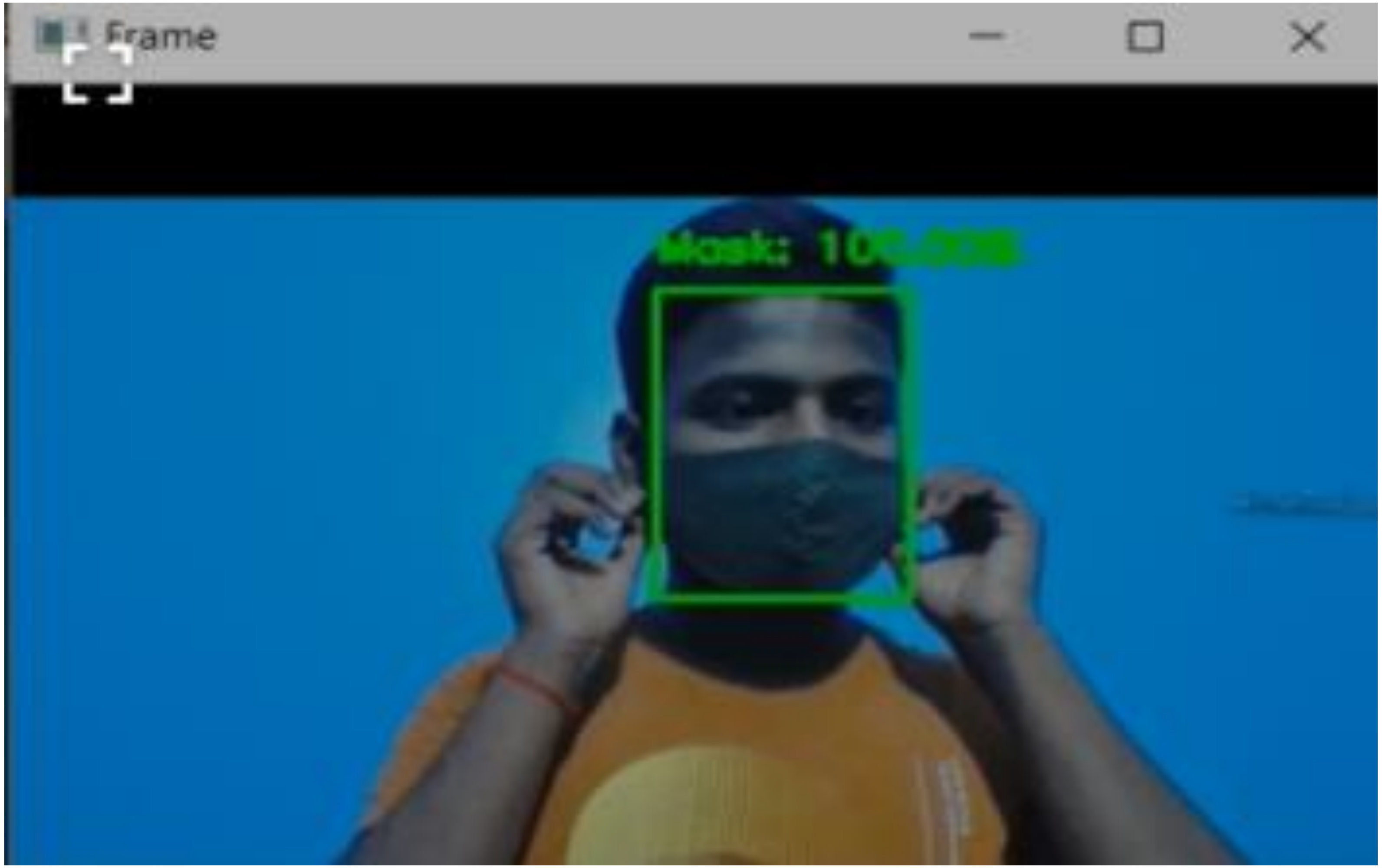


```
2021-05-14 19:13:40.997010: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cublas64_11.dll'; dlerror: cublas64_11.dll not found
2021-05-14 19:13:41.005605: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cublasLt64_11.dll'; dlerror: cublasLt64_11.dll not found
2021-05-14 19:13:41.015105: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cufft64_10.dll'; dlerror: cufft64_10.dll not found
2021-05-14 19:13:41.025614: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'curand64_10.dll'; dlerror: curand64_10.dll not found
2021-05-14 19:13:41.033880: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cusolver64_11.dll'; dlerror: cusolver64_11.dll not found
2021-05-14 19:13:41.042517: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cusparse64_11.dll'; dlerror: cusparse64_11.dll not found
2021-05-14 19:13:41.054131: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic libra
ry 'cudnn64_8.dll'; dlerror: cudnn64_8.dll not found
2021-05-14 19:13:41.059973: W tensorflow/core/common_runtime/gpu/gpu_device.cc:1766] Cannot dlopen some GPU libraries. P
lease make sure the missing libraries mentioned above are installed properly if you would like to use GPU. Follow the gu
ide at https://www.tensorflow.org/install/gpu for how to download and setup the required libraries for your platform.
Skipping registering GPU devices...
2021-05-14 19:13:41.080029: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized wit
h oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:
 AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2021-05-14 19:13:41.114906: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1258] Device interconnect StreamExecutor
with strength 1 edge matrix:
2021-05-14 19:13:41.125639: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1264]
[INFO] starting video stream...
(1, 1, 200, 7)
2021-05-14 19:13:46.894692: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimizatio
n Passes are enabled (registered 2)
```

Debugging

Output as no mask





Output with mask

Source Code

```
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):
    (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)

prototxtPath = r"face_detector\deploy.prototxt"
weightsPath = r"face_detector\res10_300x300_ssd_iter_140000.caffemodel"
faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)
```

```
maskNet = load_model("mask_detector.model")
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=400)
    (locs, preds) = detect_and_predict_mask(frame, faceNet, maskNet)
    for (box, pred) in zip(locs, preds):

        (startX, startY, endX, endY) = box
        (mask, withoutMask) = pred
        label = "Mask" if mask > withoutMask else "No Mask"
        color = (0, 255, 0) if label == "Mask" else (0, 0, 255)
        label = "{}: {:.2f}%".format(label, max(mask, withoutMask) * 100)
        cv2.putText(frame, label, (startX, startY - 10),
            cv2.FONT_HERSHEY_SIMPLEX, 0.45, color, 2)
        cv2.rectangle(frame, (startX, startY), (endX, endY), color, 2)
    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
cv2.destroyAllWindows()
vs.stop()
```

Future And Scope

- Global Facial Recognition Scenario
- Key player in the industry
- Gadget Security
- Distinguishing Genetic disorders

THANK YOU