Project Development Phase

Sprint-3 Coding (Drowning Detection along with age prediction)

Date	10 November 2022
Team ID	PNT2022TMID51724
Project Name	VirtualEye - Life Guard For Swimming Pools To Detect Active Drowning
Maximum Marks	8 Marks

app.py :

```
# import necessary packages
import cvlib as cv
from cvlib.object_detection import draw_bbox
import cv2 import time import numpy as np import
math import argparse def getFaceBox(net,
frame,conf_threshold = 0.75):     frameOpencvDnn
= frame.copy()     frameHeight =
frameOpencvDnn.shape[0]     frameWidth =
frameOpencvDnn.shape[1]
   blob = cv2.dnn.blobFromImage(frameOpencvDnn,1.0,(300,300),[104, 117, 123], True,
False)

   net.setInput(blob)
detections = net.forward()
bboxes = []


   for i in range(detections.shape[2]):
confidence = detections[0,0,i,2]         if
confidence > conf_threshold:             x1 =
int(detections[0,0,i,3]* frameWidth)           y1 =
int(detections[0,0,i,4]* frameHeight)           x2 =
int(detections[0,0,i,5]* frameWidth)           y2 =
int(detections[0,0,i,6]* frameHeight)
bboxes.append([x1,y1,x2,y2])

cv2.rectangle(frameOpencvDnn,(x1,y1),(x2,y2),(0,255,0),int(round(frameHeight/150)),8)

   return frameOpencvDnn , bboxes

faceProto = "opencv_face_detector.pbtxt"
faceModel = "opencv_face_detector_uint8.pb"
```

```
ageProto = "age_deploy.prototxt" ageModel
= "age_net.caffemodel"

genderProto = "gender_deploy.prototxt"
genderModel = "gender_net.caffemodel"

MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)','(21-24)' ,'(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList = ['Male', 'Female']


#load the network
ageNet = cv2.dnn.readNet(ageModel,ageProto) genderNet
= cv2.dnn.readNet(genderModel, genderProto) faceNet =
cv2.dnn.readNet(faceModel, faceProto)

# open webcam
webcam = cv2.VideoCapture(0) padding
= 20

if not webcam.isOpened():
print("Could not open webcam")
exit()


t0 = time.time() #gives time in seconds after 1970
#print('t0=',t0)
#variable dcount stands for how many seconds the person has been standing still for
centre0 = np.zeros(2) isDrowning = False

#this loop happens approximately every 1 second, so if a person doesn't move,
#or moves very little for 10seconds, we can say they are drowning

# loop through frames while
webcam.isOpened():

    # read frame from webcam
    status, frame = webcam.read()

    if not status:
        break
    #small_frame = cv2.resize(frame,(0,0),fx = 0.5,fy = 0.5)
    # apply object detection
    bbox, label, conf = cv.detect_common_objects(frame, confidence=0.25,
model='yolov3-tiny')

    print(bbox, label, conf)
```

```
   if(len(bbox)>0):
bbox0 = bbox[0]
#centre = np.zeros(s)
        centre = [0,0]


         #for i in range(0, len(bbox)):
            #centre[i] =[(bbox[i][0]+bbox[i][2])/2,(bbox[i][1]+bbox[i][3])/2 ]

        centre =[(bbox0[0]+bbox0[2])/2,(bbox0[1]+bbox0[3])/2 ]

         #make vertical and horizontal movement variables
hmov = abs(centre[0]-centre0[0])
        vmov = abs(centre[1]-centre0[1])

         #there is still need to tweek the threshold
         #this threshold is for checking how much the centre has moved

        x=time.time()

        threshold = 10
#print("hmov=",hmov)
if(hmov>threshold or vmov>threshold):
            print(x-t0, 'sif')
t0 = time.time()
            isDrowning = False
         else:                  print(x-t0,
'selse')           if((time.time() - t0)
> 10):                   isDrowning =
True
               small_frame = cv2.resize(frame,(0,0),fx = 0.5,fy = 0.5)

               frameFace ,bboxes = getFaceBox(faceNet,small_frame)
if not bboxes:
                  print("No face Detected, Checking next frame")
continue                for bbox in bboxes:                    face =
small_frame[max(0,bbox[1]padding):min(bbox[3]+padding,f
rame.shape[0]-1),
                        max(0,bbox[0]-padding):min(bbox[2]+padding, frame.shape[1]-1)]
                  blob = cv2.dnn.blobFromImage(face, 1.0, (227, 227),
MODEL_MEAN_VALUES, swapRB=False)
                  genderNet.setInput(blob)
genderPreds = genderNet.forward()                    gender =
genderList[genderPreds[0].argmax()]
                  print("Gender : {}, conf = {:.3f}".format(gender, genderPreds[0].max()))
```

```
                ageNet.setInput(blob)
agePreds = ageNet.forward()
age = ageList[agePreds[0].argmax()]
print("Age Output : {}".format(agePreds))
                print("Age : {}, conf = {:.3f}".format(age, agePreds[0].max()))

                label = "{},{}".format(gender, age)
cv2.putText(frameFace, label, (bbox[0], bbox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 255), 2, cv2.LINE_AA)
cv2.imshow("Age Gender Demo", frameFace)

            print("time : {:.3f}".format(time.time() - t0))

            #print("Entered in to true")
        #print('bounding box: ', bbox, 'label: ' label ,'confidence: ' conf[0], 'centre: ', centre)
        #print(bbox,label ,conf, centre)
            print('bbox: ', bbox, 'centre:', centre, 'centre0:', centre0)
       print('Is he/she drowning: ', isDrowning)
       #print('End of the program')


       centre0 = centre
          # draw bounding box over detected objects
# draw bounding box over detected objects
       out = draw_bbox(frame, bbox, label, conf, write_conf=True)
          # display output
       cv2.imshow("Real-time object detection", out)

   # press "Q" to stop     if
cv2.waitKey(1) & 0xFF == ord('q'):
       break

# release resources webcam.release()
cv2.destroyAllWindows()
```