**Age and Gender Recognition**

**1.Importing the necessary libraries.**

1. import cv2
2. import math
3. import argparse

**2.In the below function we are highlighting the face using rectangle.**

def highlightFace(net, frame, conf\_threshold=0.7):

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()

faceBoxes=[]

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)

faceBoxes.append([x1,y1,x2,y2])

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

return frameOpencvDnn,faceBoxes

**3. ArgumentParser**

The argparse module makes it easy to write user-friendly command-line interfaces. The program defines what arguments it requires, and [argparse](https://docs.python.org/3/library/argparse.html#module-argparse) will figure out how to parse those out of [sys.argv](https://docs.python.org/3/library/sys.html#sys.argv). The argparse module also automatically generates help and usage messages and issues errors when users give the program invalid arguments.

parser=argparse.ArgumentParser()

parser.add\_argument('--image')

args=parser.parse\_args()

**4.Now using the faceProto and faceModel for detecting the face and ageProto and ageModel for age recognition and the genderProto and genderModel for gender recognition.**

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"

**5.** **MODEL\_MEAN\_VALUES**

**Mean subtraction is used to help combat illumination changes in the input images in our dataset. We can therefore view mean subtraction as a technique used to aid our Convolutional Neural Networks**.

MODEL\_MEAN\_VALUES=(78.4263377603, 87.7689143744, 114.895847746)

**6.ageList and genderList are used for ages and genders respectively these are used in caffemodels.**

ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']

genderList=['Male','Female']

**7.Now the variables defined above in step 4 are being read in this step.**

**Actually, we are loading the files that are stored in the variables (see step 4) for detecting the face, age and gender.**

faceNet=cv2.dnn.readNet(faceModel,faceProto)

ageNet=cv2.dnn.readNet(ageModel,ageProto)

genderNet=cv2.dnn.readNet(genderModel,genderProto)

**8.In this step we are capturing the video and process that video for age and gender recognition.**

video=cv2.VideoCapture(args.image if args.image else 0)

**9.Now from here while the loop is true we process the video for age and gender recognition**

padding=20

while cv2.waitKey(1)<0:

hasFrame,frame=video.read()

if not hasFrame:

cv2.waitKey()

break

resultImg,faceBoxes=highlightFace(faceNet,frame)

if not faceBoxes:

print("No face detected")

for faceBox in faceBoxes:

face=frame[max(0,faceBox[1]-padding):

min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)

:min(faceBox[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(f'Gender: {gender}')

ageNet.setInput(blob)

agePreds=ageNet.forward()

age=ageList[agePreds[0].argmax()]

print(f'Age: {age[1:-1]} years')

cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE\_AA)

cv2.imshow("Detecting age and gender", resultImg)