Modern Education Society's

Wadia College of Engineering

Pune-01 Department of

Computer Engineering

Name of Student:	**Class:**
Semester/Year:	**Roll No:**
Date of Performance:	**Date of Submission:**
Examined By:	**Experiment No: Mini Project**

ASSIGNMENT – Mini Project

TITLE: Gender and Age Detection: predict if a person is a male or female and also their age

AIM: To develop a gender and age detection system using deep learning techniques, leveraging the UTK Dataset to train a model for accurate predictions based on facial images.

REQUIREMENTS:

- **Data Set Name**: UTK Dataset
- **Tools Used**: Python programming language, Keras (interface for TensorFlow library)
- **Data Mining Tasks Done**: Classification (Gender Prediction), Regression (Age Prediction)

THEORY

Gender and age detection from images has been a long-standing problem in computer vision. The UTK Dataset comprises age, gender, images, and pixels in .csv format. Various methodologies have been proposed over the years to tackle this issue. In this project, we aim to utilize the Python programming language, along with the Keras interface for the TensorFlow library, for age and gender detection.

Keras is chosen as it offers a user-friendly interface, ease of use, isolation, and extensibility. It supports both convolutional neural networks (CNNs) and recurrent networks, as well as combinations of the two. Additionally, it seamlessly runs on both CPU and GPU, making it suitable for rapid prototyping and experimentation.

Data Set Description:

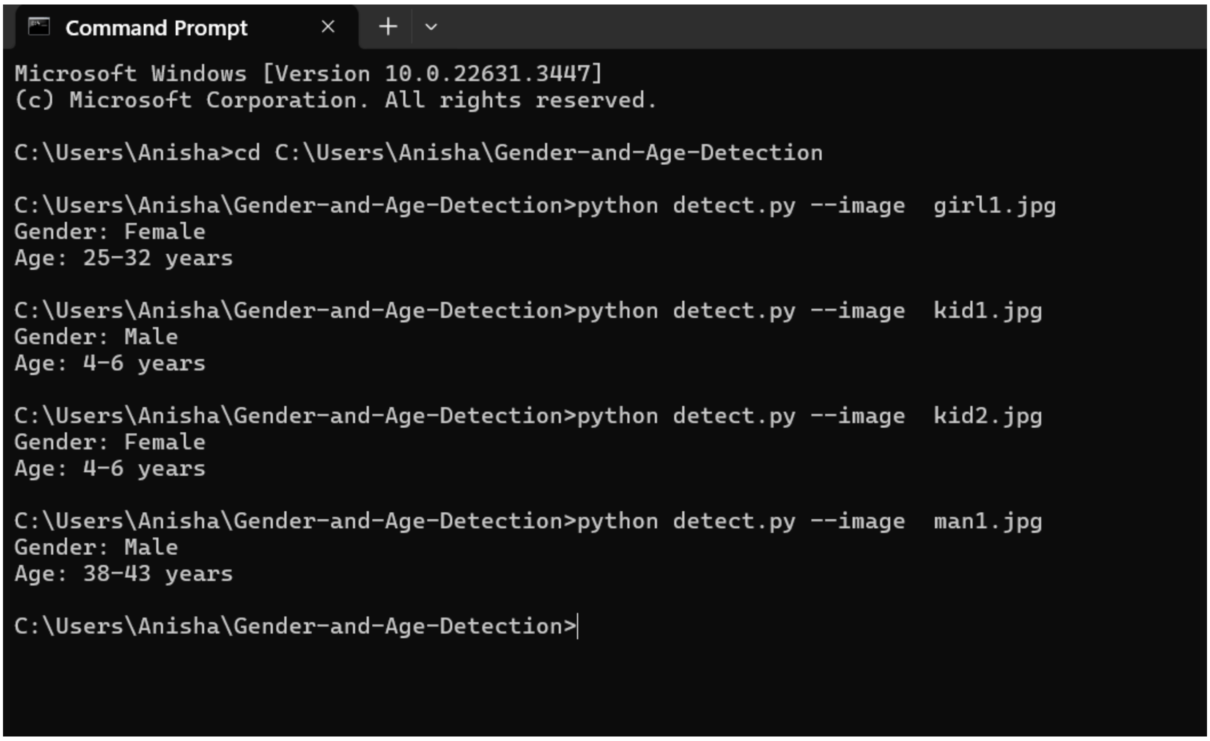
The UTK Dataset contains images of faces along with corresponding labels for age and gender. Each image is associated with metadata including age and gender labels. The dataset is provided in .csv format, where each row represents an image entry with associated age, gender, and pixel data.

CODE:

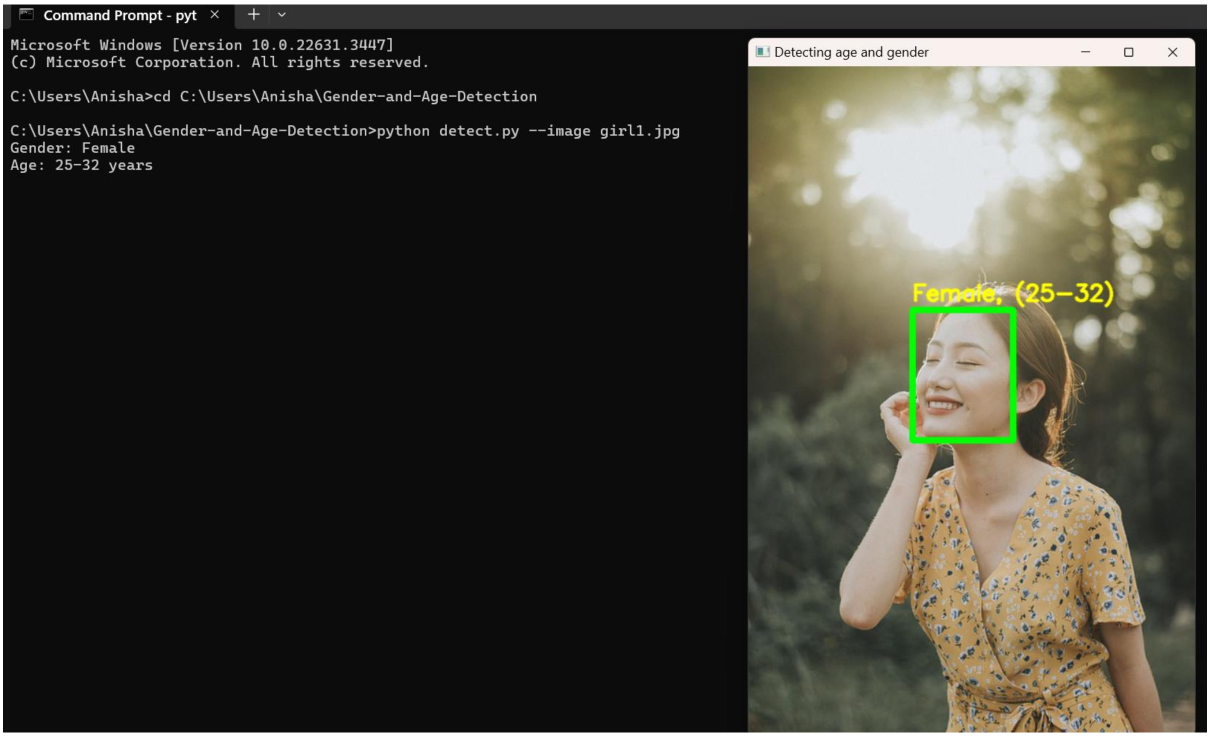
```
import cv2
import math
import argparse

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


parser=argparse.ArgumentParser()
parser.add_argument('--image')

args=parser.parse_args()

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)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList=['Male','Female']

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

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

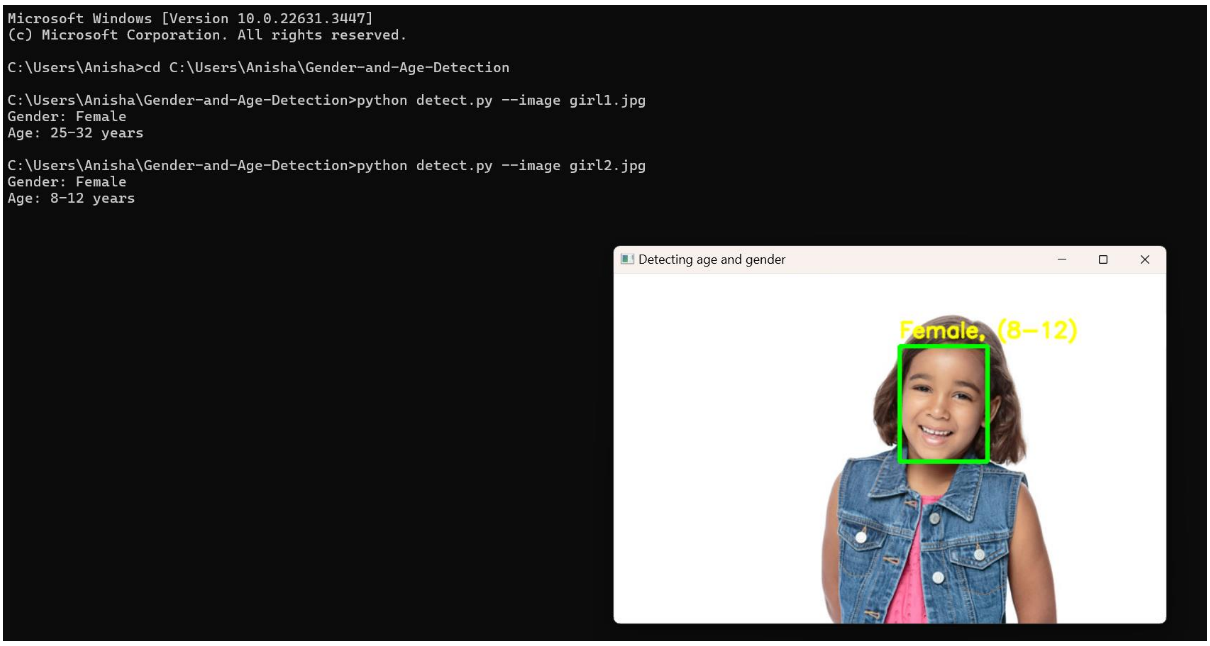
OUTPUT (Screenshots of Implementation):



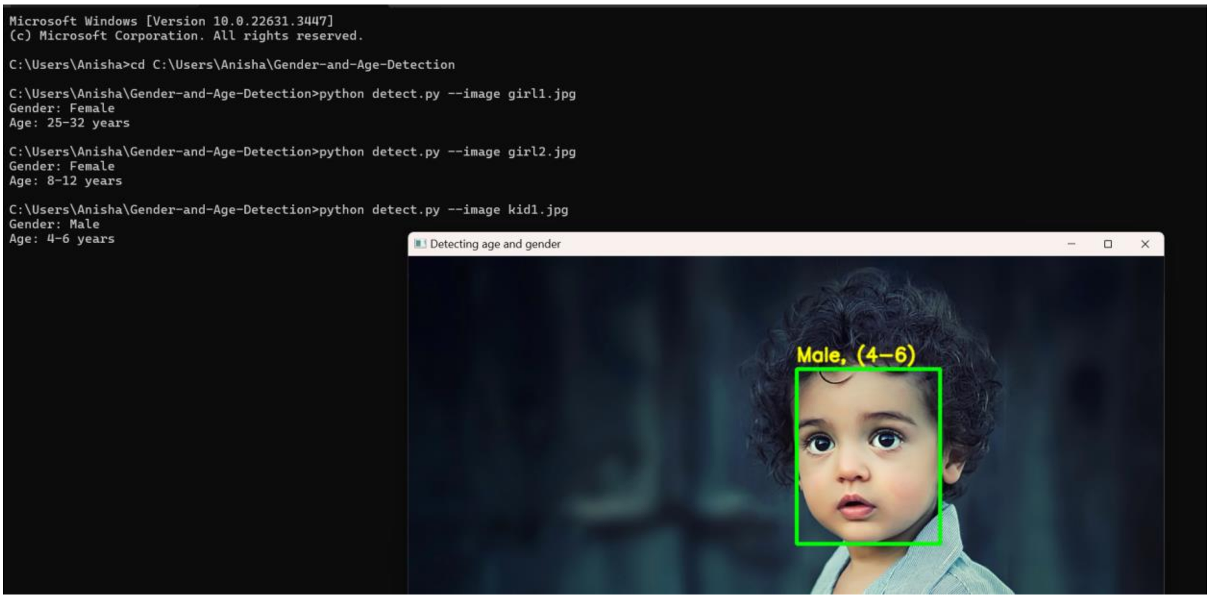
```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Anisha>cd C:\Users\Anisha\Gender-and-Age-Detection

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image  girl1.jpg
Gender: Female
Age: 25-32 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image  kid1.jpg
Gender: Male
Age: 4-6 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image  kid2.jpg
Gender: Female
Age: 4-6 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image  man1.jpg
Gender: Male
Age: 38-43 years

C:\Users\Anisha\Gender-and-Age-Detection>
```



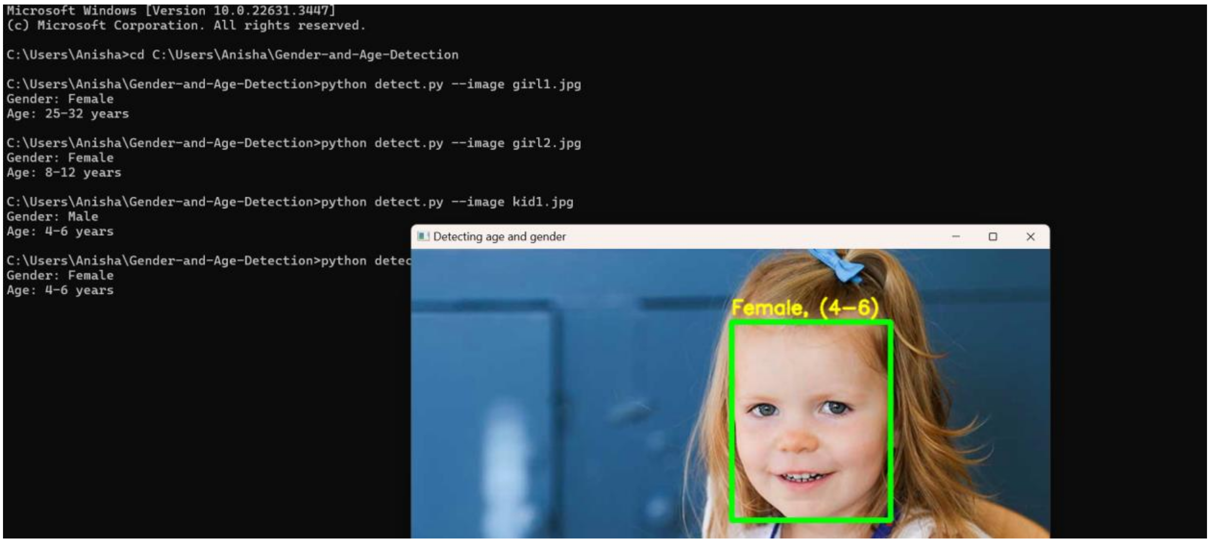
```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Anisha>cd C:\Users\Anisha\Gender-and-Age-Detection

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image girl1.jpg
Gender: Female
Age: 25-32 years
```



```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Anisha>cd C:\Users\Anisha\Gender-and-Age-Detection

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image girl1.jpg
Gender: Female
Age: 25-32 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image girl2.jpg
Gender: Female
Age: 8-12 years
```



```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Anisha>cd C:\Users\Anisha\Gender-and-Age-Detection

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image girl1.jpg
Gender: Female
Age: 25-32 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image girl2.jpg
Gender: Female
Age: 8-12 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image kid1.jpg
Gender: Male
Age: 4-6 years
```



```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Anisha>cd C:\Users\Anisha\Gender-and-Age-Detection

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image girl1.jpg
Gender: Female
Age: 25-32 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image girl2.jpg
Gender: Female
Age: 8-12 years

C:\Users\Anisha\Gender-and-Age-Detection>python detect.py --image kid1.jpg
Gender: Male
Age: 4-6 years

C:\Users\Anisha\Gender-and-Age-Detection>python detec
Gender: Female
Age: 4-6 years
```

CONCLUSION:

In this mini project, we successfully implemented gender and age detection using Python and Keras. By utilizing the UTK Dataset and training a deep learning model, we were able to predict the gender and age of individuals from facial images. Keras proved to be a powerful tool for prototyping and deploying deep learning models, offering ease of use and flexibility. This project demonstrates the effectiveness of deep learning techniques in solving real-world problems such as gender and age detection from images.