**Practical Assignment**

**Objective: - Gender and Age Detection**

You might have seen many smartphone cameras are now equipped with AI. They can even predict if a person is a male or female and their age. This can be done with deep learning but we will need a good amount of data to make this model.

**Dataset Link: -**

[**https://drive.google.com/file/d/1yy\_poZSFAPKi0y2e2yj9XDe1N8xXYuKB/view**](https://drive.google.com/file/d/1yy_poZSFAPKi0y2e2yj9XDe1N8xXYuKB/view)

**Task: -** Create a Web Application using Flask. Use the end user should be able to upload an image and get results with the prediction score.

**Deployment: -** Any Free Platform(Try to look out for free options.)

**Assignment Submission: -** Only submit the hosted app link. OR GitHub Link

import cv2

import dlib

import numpy as np

img = cv2.imread('assets/kid1.jpg')

img = cv2.resize(img, (720, 640))

frame = img.copy()

# ------------ Model for Age detection --------#

age\_weights = "Models/age\_deploy.prototxt"

age\_config = "Models/age\_net.caffemodel"

age\_Net = cv2.dnn.readNet(age\_config, age\_weights)

# Model requirements for image

ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)',

           '(25-32)', '(38-43)', '(48-53)', '(60-100)']

model\_mean = (78.4263377603, 87.7689143744, 114.895847746)

# storing the image dimensions

fH = img.shape[0]

fW = img.shape[1]

Boxes = []  # to store the face co-ordinates

mssg = 'Face Detected'  # to display on image

# ------------- Model for face detection---------#

face\_detector = dlib.get\_frontal\_face\_detector()

# converting to grayscale

img\_gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

# -------------detecting the faces--------------#

faces = face\_detector(img\_gray)

# If no faces our detected

if not faces:

    mssg = 'No face detected'

    cv2.putText(img, f'{mssg}', (40, 40),

                cv2.FONT\_HERSHEY\_SIMPLEX, 2, (200), 2)

    cv2.imshow('Age detected', img)

    cv2.waitKey(0)

else:

    # --------- Bounding Face ---------#

    for face in faces:

        x = face.left()  # extracting the face coordinates

        y = face.top()

        x2 = face.right()

        y2 = face.bottom()

        # rescaling those coordinates for our image

        box = [x, y, x2, y2]

        Boxes.append(box)

        cv2.rectangle(frame, (x, y), (x2, y2),

                      (00, 200, 200), 2)

    for box in Boxes:

        face = frame[box[1]:box[3], box[0]:box[2]]

        # ----- Image preprocessing --------#

        blob = cv2.dnn.blobFromImage(

            face, 1.0, (227, 227), model\_mean, swapRB=False)

        # -------Age Prediction---------#

        age\_Net.setInput(blob)

        age\_preds = age\_Net.forward()

        age = ageList[age\_preds[0].argmax()]

        cv2.putText(frame, f'{mssg}:{age}', (box[0],

                                             box[1] - 10),

                    cv2.FONT\_HERSHEY\_SIMPLEX, 0.8,

                    (0, 255, 255), 2, cv2.LINE\_AA)

        cv2.imshow("Detecting Age", frame)

        cv2.waitKey(0)