iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** X **Month of publication:** October 2022

DOI: https://doi.org/10.22214/ijraset.2022.47254

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue X Oct 2022- Available at www.ijraset.com

Safety for Drivers using OpenCV in Python

Mattupalli Nikitha[1], Adhina Roy[2], Shreya Agrawal[3]
[1, 2, 3]*Electronics and Communications department, Vellore Institute of Technology*

Abstract: ***Majority of accidents happening in the country is mainly due to the usage of cell-phones or drowsiness. Travelling long distance or driving trucks and taxis for a long time every day and night can cause the driver to be sleep deprived and drowsy which can lead to accidents. With this project, we will be building a system using python and OpenCV that will detect the age and gender of the person and detect if there is any mobile phone in the vicinity. It also detects if the person is sleeping or not and will alarm the driver accordingly. With this project, we would like to reduce the number of accidents happening around us.***
Keywords: OpenCV, Keras, NumPy, Pygame, Face recognition, Object detection, CNN

I. INTRODUCTION

The goal of this paper is to develop a python code for safety of drivers, that detects the person's age and gender. It detects if the person is sleeping by checking if the eyes are closed for a few seconds or using a mobile phone so as to ensure the safety of the driver by alarming the person.

Terms used in the project:

1) *OpenCV:* It is an open-source Computer Vision and Machine Learning library. This library is capable of processing real-time image and video. It supports the Deep Learning frameworks TensorFlow, Caffe, and PyTorch.
2) *Face Recognition and object detection with OpenCV:* A computer vision technology is face recognition. We discover and show human faces in any digital image using face recognition and detection. It is a subdomain of Object Detection, where we try to observe objects. These objects are of particular class such as humans, vehicles, animals etc.
3) *CNN:* A Convolutional Neural Network is a deep neural network widely used for the purposes of image recognition, image processing and NLP.
4) *Gender and Age Detection:* We will use Deep Learning to accurately identify the gender and age of a person from a single image of a face. We will use the models trained by Tal Hassner and Gil Levi. The predicted gender may be one of 'Male' and 'Female', and the predicted age may be one of the following ranges- (0 – 2), (4 – 6), (8 – 12), (15 – 20), (25 – 32), (38 – 43), (48 – 53), (60 – 100).
5) *Numpy:* Large, multi-dimensional arrays and matrices are supported in scientific computing via the Python software library known as NumPy. Numerous open-source software interfaces and contributors are present in NumPy.

Additionally, it includes the following:

a) A robust array object with N dimensions;
b) Broadcasting capabilities;
c) The ability to use tools to merge C/C++ and FORTRAN code.

For mathematical computations like linear algebra, the Fourier transform, etc., NumPy is helpful. Data size and type can be properly defined in numPy. NumPy allows provide quick integration with a variety of databases. Additionally, it is BSD-licensed, although with some limitations.

II. PROCEDURE

A. *Gender and Age Detection*

Two custom CNN layers are used for age group and gender estimation. The age group classification and gender classification trains the CNN layer over many images. These CNN layers can be implemented over OpenCV and can detect age and gender on live camera. The CNN using a Caffe deep learning framework

Using Caffe, there are 4 steps to training a CNN:

1) *Step 1:* The first step is data preparation, where we clean the photos and store them in a Caffe-compatible format. We'll create a Python script to take care of the pre-processing and storage of the images.

2) *Step 2:* A CNN architecture is selected in this stage, and its parameters are defined in a configuration file with the.prototxt extension.
3) *Step 3:* Model optimization is the responsibility of the solver. The solver parameters are specified in a configuration file with the.prototxt extension.
4) *Step4:* Training the model entails running a single Caffe command from the terminal. We will receive the trained model in a file with the extension when the model has been trained .caffemodel.

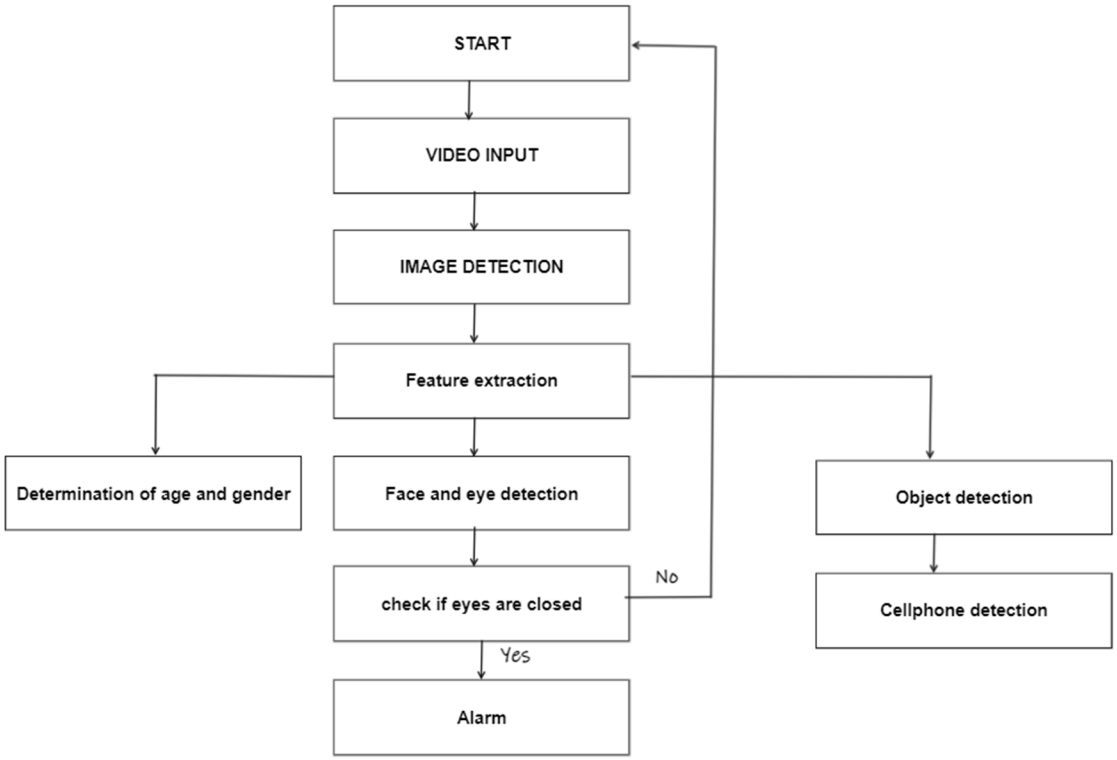
gender_net.caffemodel: It is the pre-trained model weights for gender detection.
deploy_gender.prototxt: is the model architecture for the gender detection model (a plain text file with a JSON-like structure containing all the neural network layer's definitions).
res10_300x300_ssd_iter_140000_fp16.caffemodel: The pre-trained model weights for face detection.
deploy.prototxt.txt: This is the model architecture for the face detection model.

B. Cellphone Detection

cv::dnn::DetectionModel Class is a Deep Neural Network module which we use to detect the cellphone after integrating it with OpenCV.

C. Face and Drowsiness Detection using Haar Classifier

A face detector known as a Haar Cascade classifier is used by OpenCV to identify drowsiness. The algorithm first requires an image with and without faces. The face detector looks at each picture point and labels it as "Face" or "Not Face," which is used to extract features. Two or three adjacent rectangles with varying contrast values combine to form the haar-like features. The camera placed in front of the driver begins to recognise the face and subsequently the eyes. The software will then take data from the webcam and determine whether the eyes are open or closed in order to detect sleepiness. The technology will play a loud alarm sound to rouse the driver up if their eyes are closed. The system will continue repeating the programme if the eyes are opened. Since the landmarks of the eyes are numbers 37 to 48, we used them for eye detection according to the Haar cascade 68 Landmarks pointers. Haar-like features have the potential for great accuracy and low expenditure. Hence with the help of this program, we can recognise whether the driver is drowsy or not and also provide a blaring alarm sound to warn the driver if sleeping.

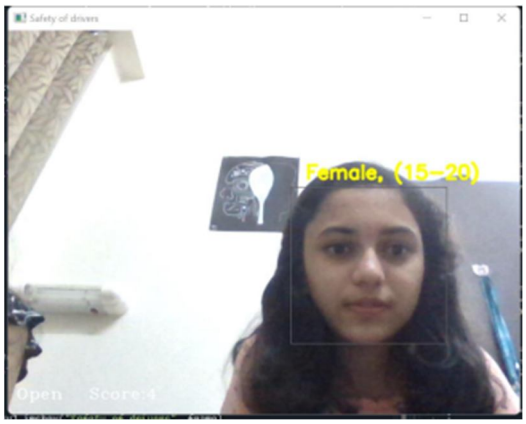


III.CODE

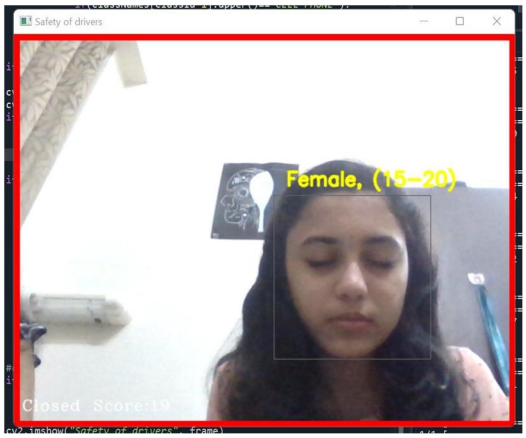
```
import cv2
import argparse
import os
from keras.models import load_model
import numpy as np
from pygame import mixer
import time

mixer.init()
sound = mixer.Sound('alarm.wav')
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']
lbl=['Close','Open']

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

faced = cv2.CascadeClassifier('haar cascade
files\haarcascade_frontalface_alt.xml')
leye = cv2.CascadeClassifier('haar cascade
files\haarcascade_lefteye_2splits.xml') reye =
cv2.CascadeClassifier('haar cascade
files\haarcascade_righteye_2splits.xml')
padding=20

model = load_model('models/cnncat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
score=0
thicc=2
rpred=[99]
lpred=[99]
classNames = []

classFile = 'coco.names'
with open(classFile, 'rt') as f:
  classNames = f.read().rstrip('\n').split('\n')
configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
weightPath = 'frozen_inference_graph.pb'
net = cv2.dnn_DetectionModel(weightPath, configPath)
net.setInputSize(320, 320)
net.setInputScale(1.0/ 127.5)
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)

while cv2.waitKey(1)<0 :
    hasFrame,frame=cap.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)]
        #ret, frame = cap.read()
        height,width = frame.shape[:2]

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
      faces =
faced.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,m
inSize=(25,25))
        left_eye = leye.detectMultiScale(gray)
        right_eye =  reye.detectMultiScale(gray)

        #cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0,0) ,
thickness=cv2.FILLED )

        classIds, confs, bbox = net.detect(frame,
confThreshold=0.5)
        print(classIds, bbox)
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')
        for (x,y,w,h) in faces:
            cv2.rectangle(frame, (x,y) , (x+w,y+h) , (100,100,100) ,
1 )
        for (x,y,w,h) in right_eye:
            r_eye=frame[y:y+h,x:x+w]
            count=count+1
            r_eye = cv2.cvtColor(r_eye,cv2.COLOR_BGR2GRAY)
            r_eye = cv2.resize(r_eye,(24,24))
            r_eye= r_eye/255
            r_eye=  r_eye.reshape(24,24,-1)
            r_eye = np.expand_dims(r_eye,axis=0)
            #rpred = model.predict_classes(r_eye)
            predict_x=model.predict(r_eye)
            rpred=np.argmax(predict_x,axis=1)
            if(rpred[0]==1):
                lbl='Open'
            if(rpred[0]==0):
                lbl='Closed'
            break
for (x,y,w,h) in left_eye:
            l_eye=frame[y:y+h,x:x+w]
            count=count+1
            l_eye = cv2.cvtColor(l_eye,cv2.COLOR_BGR2GRAY)
            l_eye = cv2.resize(l_eye,(24,24))
            l_eye= l_eye/255
            l_eye=l_eye.reshape(24,24,-1)
            l_eye = np.expand_dims(l_eye,axis=0)
            #lpred = model.predict_classes(l_eye)
            predict_x=model.predict(l_eye)
            lpred=np.argmax(predict_x,axis=1)
if(lpred[0]==1):
                lbl='Open'
            if(lpred[0]==0):
                lbl='Closed'
            break
if(rpred[0]==0 and lpred[0]==0):
            score=score+1
            cv2.putText(frame,"Closed",(10,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
            cv2.putText(frame, f'{gender}, {age}', (faceBox[0],
faceBox[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.8,
(0,255,255), 2, cv2.LINE_AA)
            if len(classIds) != 0:   #
                for classId, confidence, box in zip(classIds.flatten(),
confs.flatten(), bbox):
                    if(classNames[classId-1].upper()=="CELL
PHONE"):
                        cv2.rectangle(frame, box, color=(0, 255, 0),
thickness=2)
                        cv2.putText(frame, classNames[classId-
1].upper(), (box[0]+10, box[1]+30),
cv2.FONT_HERSHEY_COMPLEX, 1, (0,255,0), 2)
        # if(rpred[0]==1 or lpred[0]==1):
            else:
                score=score-1
                cv2.putText(frame,"Open",(10,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
                cv2.putText(frame, f'{gender}, {age}', (faceBox[0],
faceBox[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.8,
(0,255,255), 2, cv2.LINE_AA)
                if len(classIds) != 0:   #
                    for classId, confidence, box in zip(classIds.flatten(),
confs.flatten(), bbox):
                        if(classNames[classId-1].upper()=="CELL
PHONE"):
                            cv2.rectangle(frame, box, color=(0, 255, 0),
thickness=2)
                            cv2.putText(frame, classNames[classId-
1].upper(), (box[0]+10, box[1]+30),
cv2.FONT_HERSHEY_COMPLEX, 1, (0,255,0), 2)
          if(score<0):
                score=0
            cv2.putText(frame,'Score:'+str(score),(100,height-20), font,
1,(255,255,255),1,cv2.LINE_AA)
            cv2.putText(frame, f'{gender}, {age}', (faceBox[0],
faceBox[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 0.8,
(0,255,255), 2, cv2.LINE_AA)
            if len(classIds) != 0:   #
                for classId, confidence, box in zip(classIds.flatten(),
confs.flatten(), bbox):
                    if(classNames[classId-1].upper()=="CELL PHONE"):
                        cv2.rectangle(frame, box, color=(0, 255, 0),
thickness=2)
                        cv2.putText(frame, classNames[classId-1].upper(),
(box[0]+10, box[1]+30), cv2.FONT_HERSHEY_COMPLEX,
1, (0,255,0), 2)
            if(score>15):
                #person is feeling sleepy so we beep the alarm
                cv2.imwrite(os.path.join(path,'image.jpg'),frame)
try:
                    sound.play()

                except:  # isplaying = False
                    pass
                if(thicc<16):
                    thicc= thicc+2
                else:
                    thicc=thicc-2
                    if(thicc<2):
                        thicc=2
                cv2.rectangle(frame,(0,0),(width,height),(0,0,255),thicc)
            #cv2.imshow('frame',frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break


            cv2.imshow("Detecting age and gender", frame)
```

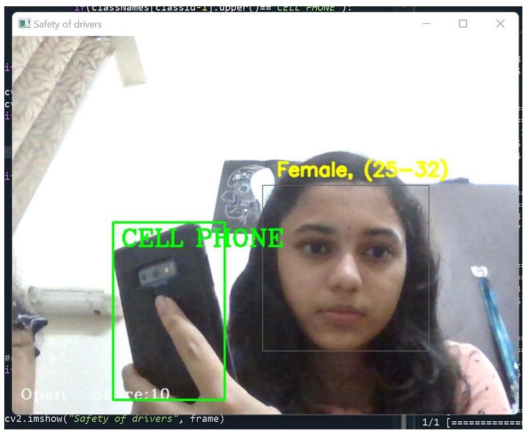
IV. RESULTS



Gender And Age Detection



Alarm Sound Upon Drowsiness Detection



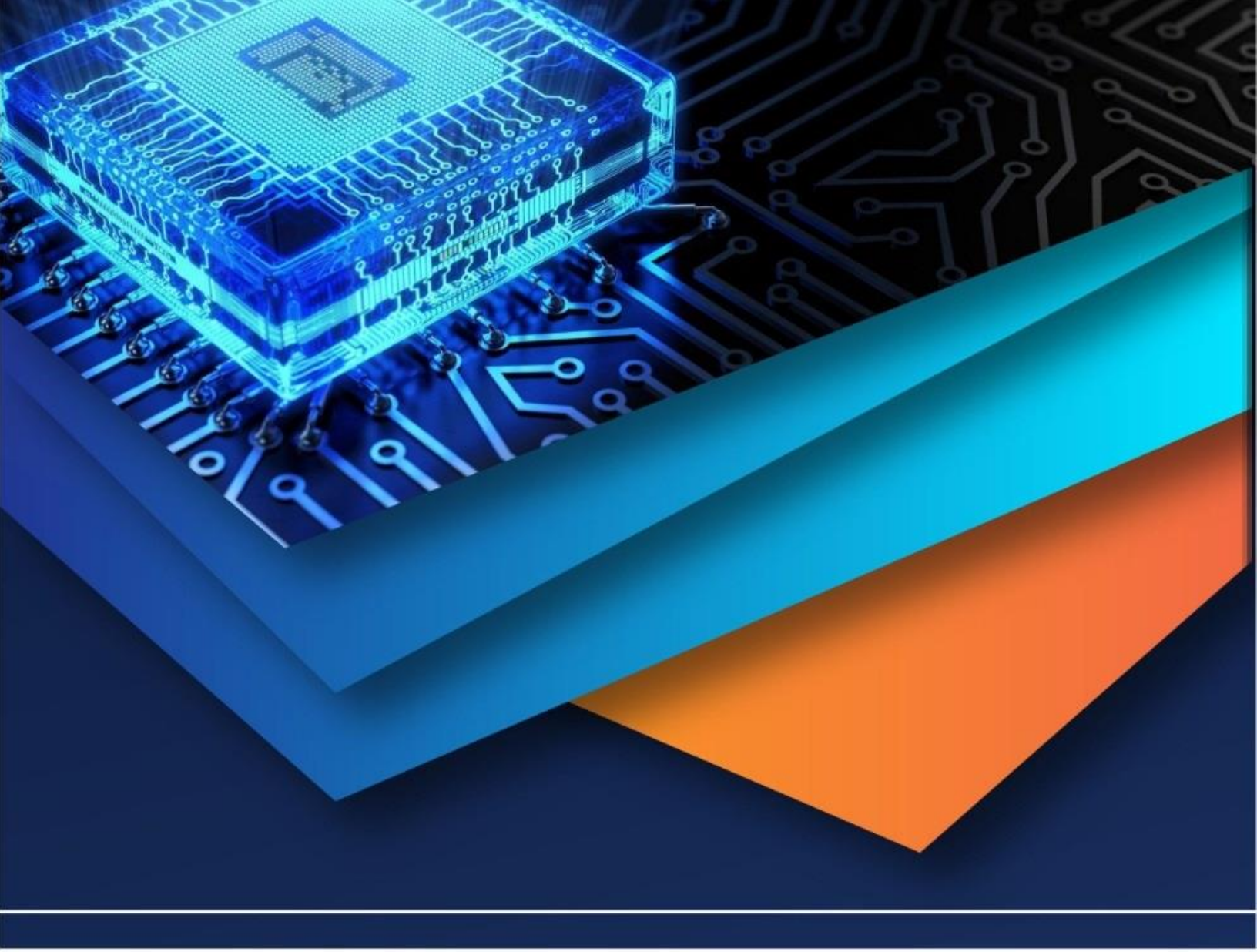
Cell Phone Detection

V. CONCLUSION

We have developed a python code that runs successfully which ensures the safety of the drivers. It alarms when the person is sleeping or if it detects a mobile phone. It also displays the gender and the age of the person with about 80% accuracy. Haar-like features have the potential for great accuracy and low expenditure. In the future years, OpenCV will become quite well-known among python programmers in the IT industry.

REFERENCES

[1] Chandan, G., Ayush Jain, and Harsh Jain. "Real time object detection and tracking using Deep Learning and OpenCV." 2018 International Conference on inventive research in computing applications (ICIRCA). IEEE, 2018.
[2] Rajput, Bhumika. "DRIVER DROWSINESS DETECTION USING PYTHON."
[3] Emami, Shervin, and Valentin Petrut Suciu. "Facial recognition using OpenCV." Journal of Mobile, Embedded and Distributed Systems 4.1 (2012): 38-43..
[4] P. Reshvanth et al., "Age Detection from Facial Images Using Python," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2021, pp. 1316-1321, doi: 10.1109/I-SMAC52330.2021.9641056.
[5] *Saxena, Meghna Raj, et al. "Real-time object detection using machine learning and opencv." Int J Inform Sci Appl (IJISA) 11.1 (2019): 0974-225.*
[6] Khunpisuth, Oraan, et al. "Driver drowsiness detection using eye-closeness detection." 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS). IEEE, 2016.
[7] Salihbašić, Alen, and Tihomir Orehovački. "Development of android application for gender, age and face recognition using opencv." 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY