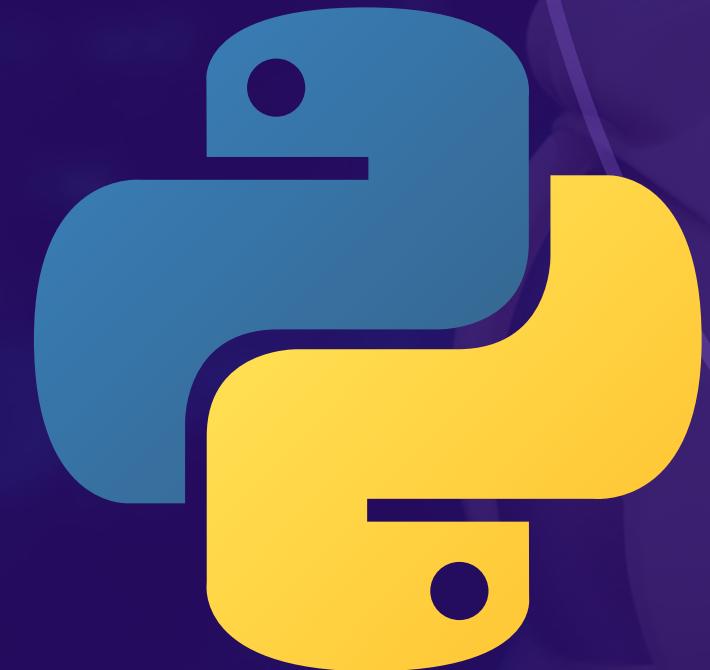


# DRIVER DROWSINESS DETECTION

BY

- APOORV MEHAR -(2021BCS091)
- RAJ CHAVHAN -(2021BCS087)
- RAHUL CHORMARE -(2021BCS071)
- VISHAL BANSODE -(2021BCS077)

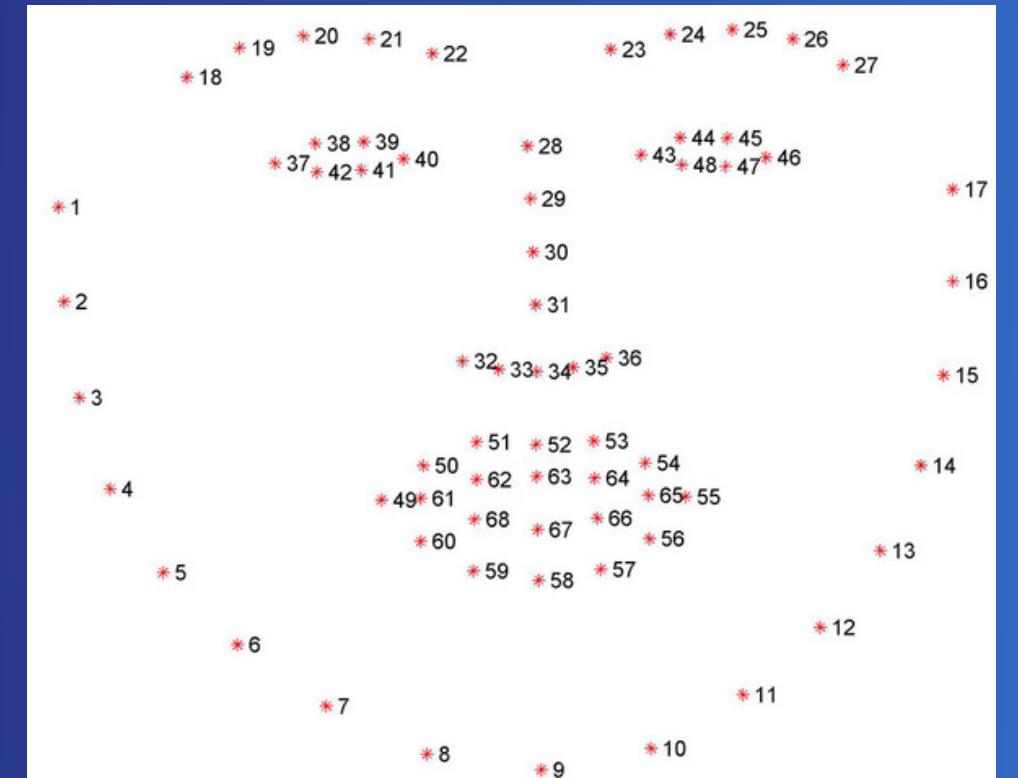


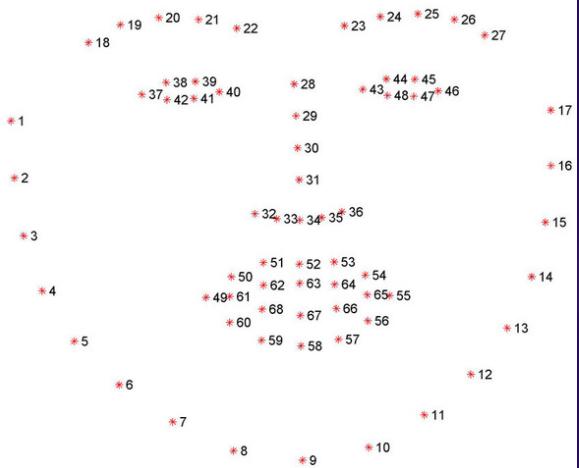
# What is Driver Drowsiness Detection ?

- Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.
- this python project will detect that a person's eyes are closed for a few seconds which will alert the driver when drowsiness is detected.
- resulting in less accident on the road due to drowsiness

# Logic of project

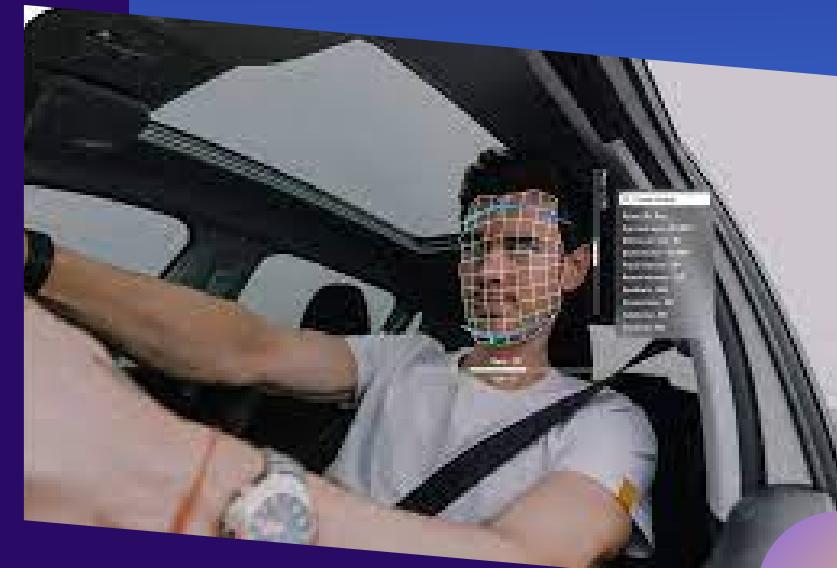
The project includes direct working with the 68 facial landmark detector and also the face detector of the Dlib library. The 68 facial landmark detector is a robustly trained efficient detector which detects the points on the human face using which we determine whether the eyes are open or they are closed.





# The working of the project

- As you can see the above screenshot where the landmarks are detected using the detector.
- Now we are taking the ratio which is described as 'Sum of distances of vertical landmarks divided by twice the distance between horizontal landmarks'.
- if the ratio is grater than 2.5 than the eye is open
- if between 2.5-2.1 Drowsiness
- less tan 2.1 sleep



“

## library using

- open cv
- dlib
- numpy
- imutils

”

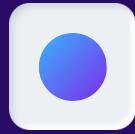


OPEN CV



# Open cv

# Open cv



## OPEN CV (OPEN SOURCE COMPUTER VISION LIBRARY)

- is a library of programming functions mainly for real-time computer vision.  
Originally developed by Intel



## ABOUT OPEN CV

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.



## APPLICATIONS

- 2D and 3D feature toolkits
- Facial recognition system
- Augmented reality
- Gesture recognition





**DLIB**



# DLIB

# DLIB



## DLIB

- Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems.



## APPLICATIONS

- Machine Learning Algorithm
- Numerical Algorithms
- Graphical Model Inference Algorithms
- Image Processing



# code

```
#Importing OpenCV Library for basic image processing functions
import cv2
# Numpy for array related functions
import numpy as np
# Dlib for deep learning based Modules and face landmark detection
import dlib
#face_utils for basic operations of conversion
from imutils import face_utils

#Initializing the camera and taking the instance
cap = cv2.VideoCapture(0)

#Initializing the face detector and landmark detector
detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")

#status marking for current state
sleep = 0
drowsy = 0
active = 0
status=""
color=(0,0,0)
```



function of dlib used to detect the face

```

def compute(ptA,ptB):
    dist = np.linalg.norm(ptA - ptB)
    return dist

```

```

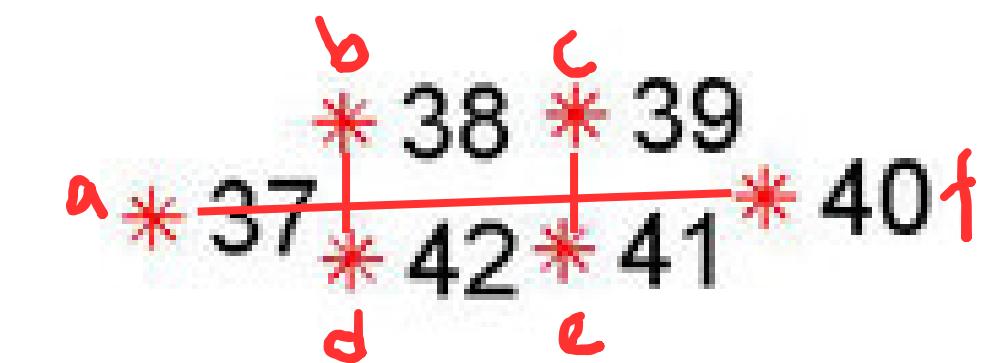
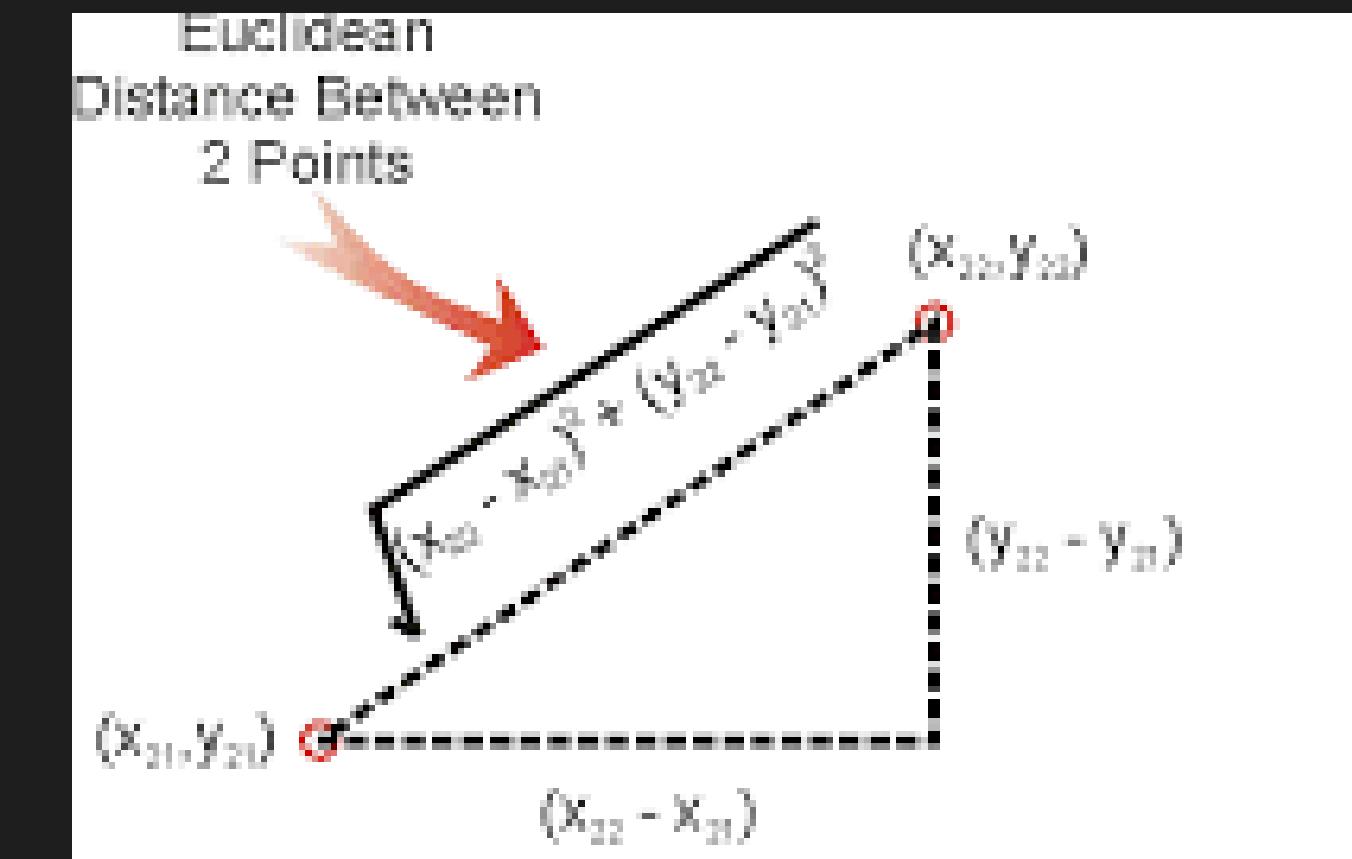
def blinked(a,b,c,d,e,f):
    up = compute(b,d) + compute(c,e)
    down = compute(a,f)
    ratio = up/(2.0*down)

```

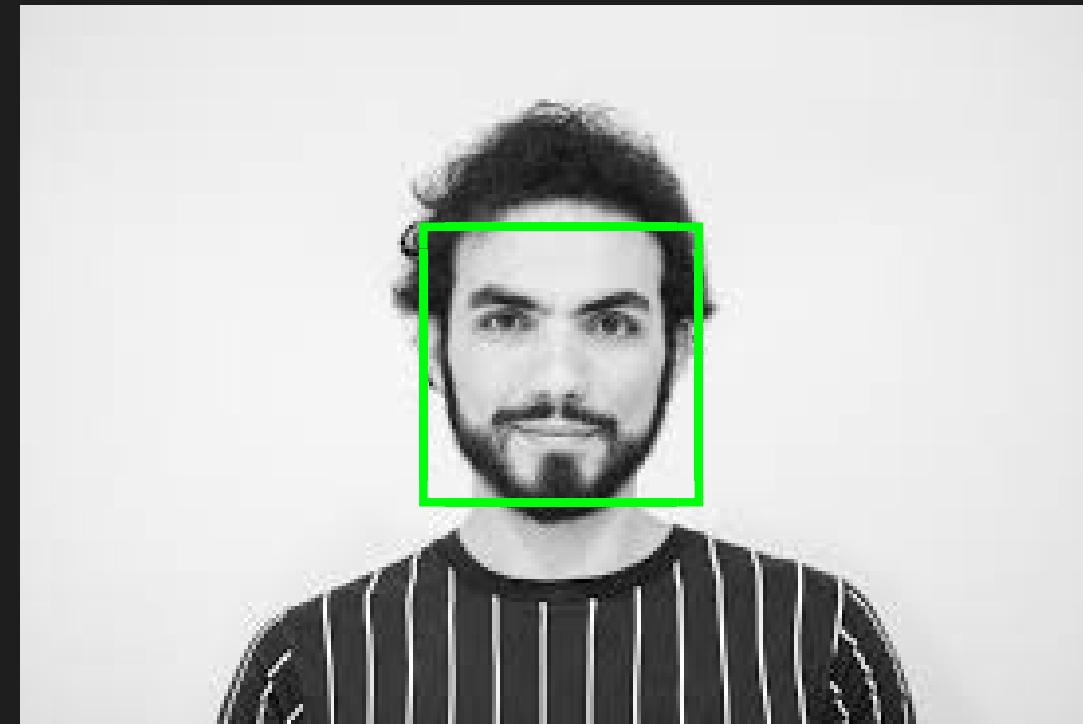
```

#Checking if it is blinked
if(ratio>0.25):
    return 2
elif(ratio>0.21 and ratio<=0.25):
    return 1
else:
    return 0

```



```
while True:  
    _, frame = cap.read()  
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)  
  
    faces = detector(gray)  
    #detected face in faces array  
    for face in faces:  
        x1 = face.left()  
        y1 = face.top()  
        x2 = face.right()  
        y2 = face.bottom()  
  
        face_frame = frame.copy()  
        cv2.rectangle(face_frame, (x1, y1), (x2, y2), (0, 255, 0), 2)  
  
    landmarks = predictor(gray, face)  
    landmarks = face_utils.shape_to_np(landmarks)  
  
    #The numbers are actually the landmarks which will show eye  
    left_blink = blinked(landmarks[36],landmarks[37],  
                         landmarks[38], landmarks[41], landmarks[40], landmarks[39])  
    right_blink = blinked(landmarks[42],landmarks[43],  
                         landmarks[44], landmarks[47], landmarks[46], landmarks[45])
```



```
#Now judge what to do for the eye blinks
if(left_blink==0 or right_blink==0):
    sleep+=1
    drowsy=0
    active=0
    if(sleep>6):
        status="SLEEPING !!!"
        olor = (255,0,0)

elif(left_blink==1 or right_blink==1):
    sleep=0
    drowsy+=1
    active=0
    if(sleep>6):
        status="Drowsy !"
        olor = (0,0,225)

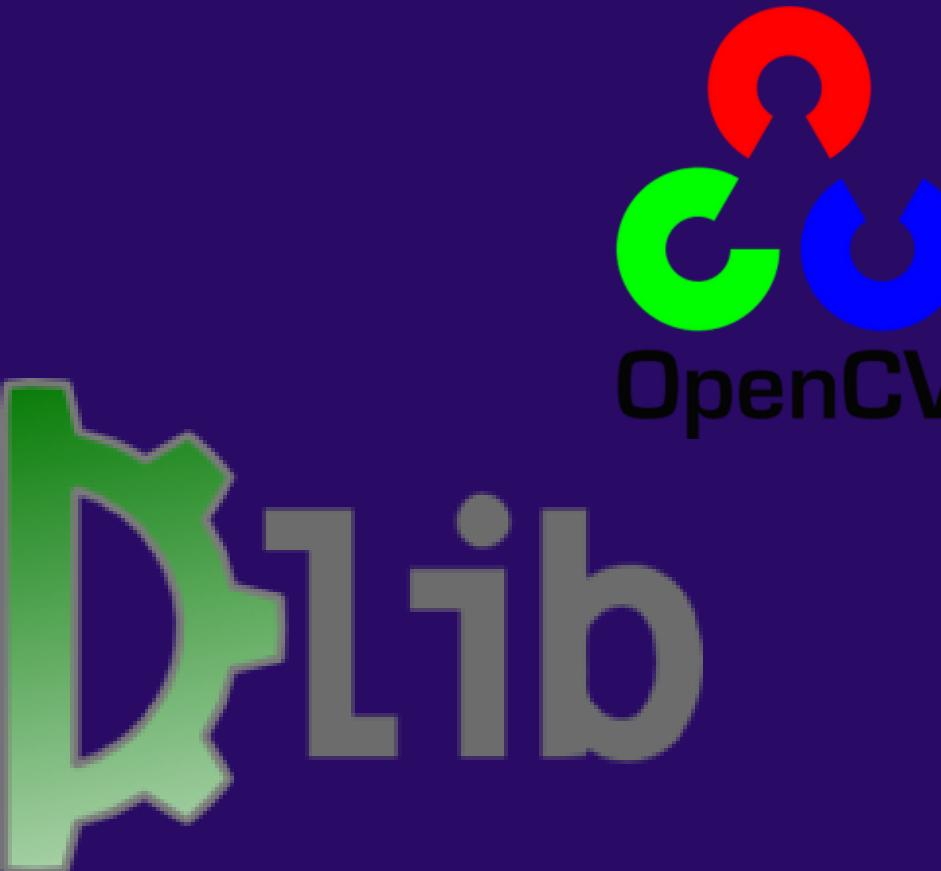
else:
    sleep=0
    drowsy=0
    active+=1
    if(sleep>6):
        status="Active :)"
        olor = (0,255,0)
```

```
cv2.putText(frame, status, (100,100), cv2.FONT_HERSHEY_SIMPLEX, 1.2, color,3)

for n in range(0, 68):
    (x,y) = landmarks[n]
    cv2.circle(face_frame, (x, y), 1, (255, 255, 255), -1)

cv2.imshow("Frame", frame)
cv2.imshow("Result of detector", face_frame)
key = cv2.waitKey(1)
if key == 27:
    break
```

# DRIVER DROWSINESS DETECTION ADVANTAGES



- Driver drowsiness detection is a car safety technology which helps prevent accidents caused by the driver getting drowsy
- Various studies have suggested that around 20% of all road accidents are drowsiness-related can prevented
- better road presence and safer road



Active :)



Drowsy !



SLEEPING !!!

thank you

