

PROJECT: Driver Drowsiness Detection System using deep learning.

In this Python project, we will be using OpenCV for gathering the images from webcam and feed them into a Deep learning model which will classify whether the person's eyes are 'Open' or 'Closed'.

Steps for model:

Step 1 – Take image as input from a camera.

Step 2 – Detect the face in the image and create a Region of Interest (ROI).

Step 3 – Detect the eyes from ROI and feed it to the classifier.

Step 4 – Classifier will categorize whether eyes are open or closed.

Step 5 – Calculate score to check whether the person is drowsy.

- The “haar cascade files” folder consists of the xml files that are needed to detect objects from the image. In our case, we are detecting the face and eyes of the person.
-

Step 1 – Take Image as Input from a Camera

With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV,

cv2.VideoCapture(0) to access the camera and set the capture object (cap). **cap.read()** will read each frame and we store the image in a frame variable.

Step 2 – Detect Face in the Image and Create a Region of Interest (ROI)

To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier **face = cv2.CascadeClassifier(' path to our haar cascade xml file')**. Then we perform the detection using **faces = face.detectMultiScale(gray)**. It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face.

for (x,y,w,h)in faces:
cv2.rectangle(frame, (x,y), (x+w, y+h), (100,100,100), 1)

Step 3 – Detect the eyes from ROI and feed it to the classifier

The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in **leye** and **reye** respectively then detect the eyes using **left_eye = leye.detectMultiScale(gray)**. Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code.

```
l_eye = frame[ y : y+h, x : x+w ]
```

l_eye only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into **r_eye**.

Step 4 – Classifier will categorize whether Eyes are open or Closed

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the color image into grayscale using **r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY)**. Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images **cv2.resize(r_eye, (24,24))**. We normalize our data for better convergence **r_eye = r_eye/255** (All values will be between 0-1). Expand the dimensions to feed into our classifier. We loaded our model using **model = load_model('models/cnnCat2.h5')** . Now we predict each eye with our model **lpred = model.predict_classes(l_eye)**. If the value of **lpred[0] = 1**, it states that eyes are open, if value of **lpred[0] = 0** then, it states that eyes are closed.

Step 5 – Calculate Score to Check whether Person is Drowsy

The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using **cv2.putText ()** function which will display real time status of the person.

Python code:

Importing libraries

```

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

```

```

import cv2
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')
face = 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')
lbl=['Close','Open']
model = load_model('models/cnn-cat2.h5')
path = os.getcwd()
cap = cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_COMPLEX_SMALL
count=0
score=0
thicc=2
rpred=[99]
lpred=[99]
while(True):
    ret, frame = cap.read()
    height,width = frame.shape[:2]
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face.detectMultiScale(gray,minNeighbors=5,scaleFactor=1.1,minSize=(25,25))
    left_eye = leye.detectMultiScale(gray)
    right_eye = reye.detectMultiScale(gray)
    cv2.rectangle(frame, (0,height-50) , (200,height) , (0,0) , thickness=cv2.FILLED )
    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)
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)
  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)
  # 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)
  if(score<0):
    score=0
    cv2.putText(frame,'Score:'+str(score),(100,height-20), font, 1,(255,255,255),1,cv2.LINE_AA)
  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  
cap.release()  
cv2.destroyAllWindows()
```