Python code is designed to detect drowsiness in a driver using computer vision techniques. It uses a webcam feed to monitor the driver's eyes and mouth, and it triggers an alarm if it detects signs of drowsiness, such as closed eyes or yawning.

Here's a breakdown of the code:

Imports:

cv2: OpenCV library for computer vision tasks.

os: Module for interacting with the operating system.

keras.models.load\_model: Loads a pre-trained Keras model.

numpy as np: Numerical computing library.

pygame.mixer: Library for handling sound.

time: Module for time-related functions.

dlib: Library for machine learning, computer vision, and image processing.

imutils.face\_utils: Utility functions for working with facial landmarks in Dlib.

Constants:

YAWN\_THRESH: Threshold for detecting a yawn.

Functions:

lip\_distance(shape): Calculates the distance between the lips using facial landmarks.

Initialization:

Initializing mixer for sound.

Loading pre-trained cascade classifiers for face, left eye, and right eye detection.

Loading pre-trained models for face detection and facial landmark prediction.

Model Loading:

Loading a pre-trained CNN model for eye state classification.

Video Capture:

Capturing video feed from the webcam.

Main Loop:

Continuously capturing frames from the video feed.

Detecting faces, left eyes, and right eyes in the frame.

Preprocessing eye images for classification.

Predicting eye states (open/closed) using the CNN model.

Calculating lip distance to detect yawning.

Updating the drowsiness score based on eye states and yawning status.

Displaying the drowsiness score and eye states on the frame.

Triggering an alarm and drawing a red rectangle around the frame if the drowsiness score exceeds a threshold or yawning is detected.

Breaking the loop if 'q' is pressed.

Release Resources:

Releasing the video capture and closing all OpenCV windows.

Algorithm:

Initialize necessary components.

Start video capture.

Continuously read frames from the video feed.

Detect faces, left eyes, and right eyes in each frame.

Preprocess eye images and predict eye states using the CNN model.

Calculate lip distance to detect yawning.

Update drowsiness score based on eye states and yawning status.

Display drowsiness score and eye states on the frame.

If the drowsiness score exceeds a threshold or yawning is detected, trigger an alarm and draw a red rectangle around the frame.

Continue until the user presses 'q' to quit.

Release video capture and close all windows.

Flowchart:

mathematica

Copy code

+----------------+

| Start Program |

+----------------+

|

V

Initialize

|

V

Capture Video

|

V

Read Frame

|

V

Detect Face, Left Eye, Right Eye

|

V

Preprocess Eye Images

|

V

Predict Eye State

|

V

Calculate Lip Distance

|

V

Update Drowsiness Score

|

V

Display Info

|

V

Check Drowsiness Threshold

|

V

Trigger Alarm

|

V

Draw Red Rectangle

|

V

Continue Loop

|

V

Check for Quit Command

|

V

Release Resources

|

V

+----------------+

| End Program |

+----------------+