



Real Time Face and Drowsiness Detection Using OpenCV

Arpit Nigam, Kapil Semwal, Vivek, Ashwini Kumar Chauhan

³U.G. Student, Department of Computer Science and Engineering, IIMT College of Engineering, Greater NOIDA, Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh, India

Abstract:

Humans have always invented mechanisms and devised techniques to ease and safeguard their lives, for mundane activities like traveling to their job, or for more captivating purposes like air travel. As technology kept on improving, modes of transportation kept on advancing and our dependency on it started escalating exponentially. In present times, almost everyone in this world uses some sort of conveyance daily. Some are wealthy enough to have their own wheels while others use public transportation. However, there are some rules for those who drive irrespective of their social status. One of them is staying alert while driving.

Forgetting our responsibilities towards secure travel has enabled hundreds of thousands of mishaps on roads. According to official enumeration 469,417 mangled in road traffic crashes and 1,51,416 persons were killed in 2018, 275,000 in 2017 and 148,708 in 2015 passed away due to car accidents in India. Out of these, at least 25% were caused due to fatigue or drowsiness or sleepiness, causing drivers to make mistakes. Drowsiness, in general, is very arduous to measure or observe unlike alcohol and drugs, as now they have clear key indicators and tests that are effortlessly available.

Keywords: Face Recognition, Drowsiness Detection, Python, OpenCV, Numpy, Harr Like Features.

Introduction:

Face recognition technology is used to recognize a person's face using a photo or video. It is a computer technique that discovers the locations and measurements of a person's face in arbitrary (digital) pictures. It recognizes facial characteristics and pays no attention to everything else, such as trees and bodies. Computer Vision (OpenCV) is a rapidly expanding field for inspecting, adjusting, and high-level insights of images. OpenCV is among the most stretched Python libraries that are used in computer vision programs. It is an open-source library containing a large number of C / C++ functions and classes, that can be used to achieve strong image processing, and to generate real-time applications system.



OpenCV Features:

- ➔ Photo data manipulation,
- ➔ Photo and video input or output,
- ➔ Image handling (image filtering, corners and edges identification, sampling and attaching images, colour change),
- ➔ Camera calibration,
- ➔ Movement inspection (optical flow, movement segmentation, tracing),
- ➔ Object identification,
- ➔ Image labelling.

In order to recognize a person's face in an image, the system needs to take a photo using a camera. Then search for the key features in the image and then use these key features to describe the position of the face. For detecting the face in an image there are several methods and algorithms including Haar like features, skin tone based, Adaboost and cascade classifier. The process of face detection involves two terms, they are explained as follows;

Face Detection is, where a photo is explored to find a face. Then the image is cropped, so that, the person's face can be drawn out. OpenCV has face detector known as "Haar Cascade classifier". When we give an input as an image or video using a camera or a webcam, the face detector detects the image location and classifies it as face or not. The classifier makes use of a xml file (haarcascade_frontalface_default.xml) to decide how to analyze each image's location.

Face Recognition is another phase, where the distinguished face image is compared with images in the database of face images. The OpenCV framework comes with an inbuilt face detector that can work 92-95% on the clear pictures. It is a little tough to distinguish a face if a person is wearing glasses or an image is foggy.

Requirement Analysis:

Python: OpenCV, that we will use for this detector is a library of python. Many other python libraries will be used.

Libraries:

Numpy: Pre-requisite for Dlib library,

Scipy: Used to find Euclidean distance of the eyes,

Pygame: Used to raise an alarm when the person feels drowsy,

Dlib: This is used to detect the frontal human face and evaluate its pose using 68 face landmarks.dat,

Imutils: Advantageous functions written for OpenCV,

Opencv: Used to get the video stream from the camera, analyze it, etc,

Time: Used to calculate the time for which the eyes are closed.

XML Files: haarcascade_eye, haarcascade_frontalface_default.

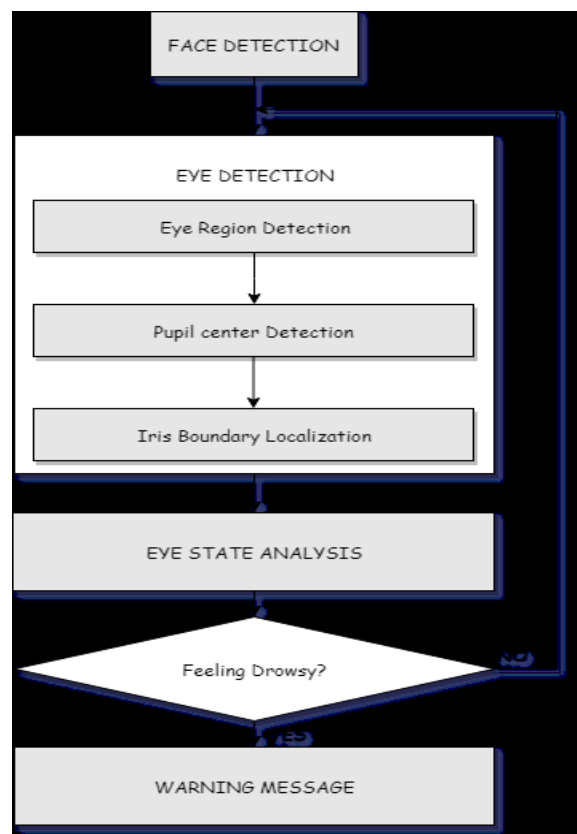
.dat File: shape_predictor_68_face_landmarks.dat

Laptop: Used to operate our code.

Webcam: Used to get the video feed or image.

OS: Preferably Windows 10.

Data-Flow Structure:



Implementation:

- ➔ In our program, we will use **Dlib**, a pre-trained program on the HELEN dataset to detect a person's face using the pre-defined 68 landmarks.
- ➔ After passing our videofeed or image to the **Dlib**, frame by frame, we will be able to detect right eye and left eye.
- ➔ Next, we draw a box around it using **OpenCV** so that, we can identify the eyes' movements and monitor its opening and closing patterns.
- ➔ Using **Scipy's** Euclidean function, we will calculate sum of both eyes' aspect ratio which is the sum of 2 distinct vertical distances between the eyelids divided by its horizontal distance (making a '+' sign on the eye).

→ Next, we check if the value of aspect ratio is less than 0.25(0.25 was chosen as a base case after some tests). If it is less, an alarm will be sounded using **Pygame** and user will be warned.

Screenshots of Project:

Non-drowsy Person(Normal):



Drowsy Person:



Conclusion:

In this paper, we have developed a system for face recognition and drowsiness detection using OpenCV. Haar cascade algorithm is used for detection. It totally meets the objectives of the system. It takes care of the issue of stressing out for individuals having fatigue-related issues by informing them about their drowsiness while driving. OpenCV is mainly used for computer vision application development. For embedded systems, where processing speed is of supreme significance, OpenCV is the fastest and thoroughgoing gizmo for computer vision applications. Furthermore, thanks to the fact that OpenCV keeps expanding is an added guarantee



that it will advance research in vision and encourage the development of rich, vision-based CPU intensive applications.

References:

- ➔ Krupali Mistry, AvneetSaluja-An Introduction to OpenCV using Python with Ubuntu, International Journal of Scientific Research in Computer Science, Engineering and Information Technology.
- ➔ Drakshayani Desai, Dr. Ramakrishna Seemakurti- Face Detection Using Open CV in Real Time.
- ➔ Mrs. Madhuram.M, B. Prithvi Kumar, Lakshman Sridhar, Nishanth Prem, Venkatesh Prasad- Face Detection and Recognition Using OpenCV, International Research Journal of Engineering and Technology (IRJET)
- ➔ M. Naveenkumar-OpenCV for Computer Vision Applications, Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 20, 2015.
- ➔ Prof. P Y Kumbhar, Mohammad Attaullah, Shubham Dhere, ShivkumarHipparagi-Real Time Face Detection and Tracking Using OpenCV, INTERNATIONAL JOURNAL FOR RESEARCH IN EMERGING SCIENCE AND TECHNOLOGY, VOLUME-4, ISSUE-4, APR-2017.
- ➔ <http://www.ee.ryerson.ca/~phiscock/thesis/drowsy-detector/drowsy-detector.pdf>
- ➔ http://opencv.willowgarage.com/documentation/reading_and_writing_images_and_video.html
- ➔ GuoboXie and Wen Lu-Image Edge Detection Based On Opencv, International Journal of Electronics and Electrical Engineering Vol. 1, No. 2, June 2013
- ➔ https://www.youtube.com/watch?list=PLQVvva0QuDfKTOs3Keq_kaG2P55YRn5v&v=OGxgnH8y2NM