**DROWSINESS DETECTION MODEL ABOUT**

**INTRODUCTION-**

* Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving.
* The objective of this Python project is to build a Drowsiness Detection Model that will detect that a person’s eyes are closed for a few seconds.
* The implementation of this project uses a pre–built model of face landmark for easy deployment on edge or computationally less efficient devices.
* The project has a direct application in the automobile industry, makes drive safer, and reduces the death toll caused by drowsy driving.

**APPROACH-**

1. Take image as input from a camera.
2. Detect the face in the image and create a Region of Interest (ROI).
3. Detect the eyes from ROI and feed it to the classifier.
4. Classifier will categorize whether eyes are fully/partially open or completely closed.
5. Calculate score to check whether the person is drowsy.

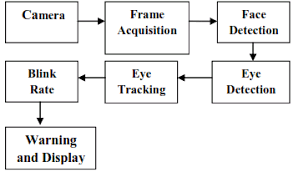
**BASIC REQUIREMENTS-**

* Python 3.X (all version except 3.10)
* OS: Windows 7/8/10, Mac, Linux
* GPU required for fast processing (Min. 512 MB)
* Web-Cam (Recommended)
* Speaker
* IDE: Visual Studio & Visual Studio Code
* Command Prompt
* Cmake (for Dlib) **(Recommended)**

**Modules used or are required to run this Project:**

* OpenCV
* Dlib
* SciPy
* Imutils
* PyGame
* Tkinters

**PRINCIPLE & WORKING-**

****

* The driver drowsiness detection is based on a Dlib, which begins recording the driver’s conscious behaviour the moment the trip begins.
* Dlib has excellent Face Detection and Face Landmark Detection algorithms built-in.
* It provides pre-trained models for face landmark detection.
* The shape\_predictor\_68\_face\_landmarks.dat file is the pre-trained Dlib model for face landmark detection.

**WORKING-**

* **Take Image as Input from a Camera**

With a webcam, we will take images as input. 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.

* **Detect Face in the Image and Create a Region of Interest (ROI)**

OpenCV algorithm for object detection takes Gray images in the input. face\_utils.FACIAL\_LANDMARKS\_68\_IDXS[“EYE\_L/R"] is used to define eyes from the Model itself.

* **Detect the eyes from ROI and feed it to the classifier**

The same procedure to detect faces is used to detect eyes. We can use an OpenCV Cascade Classifier to detect a face and eye and use it to get the face bounding box.

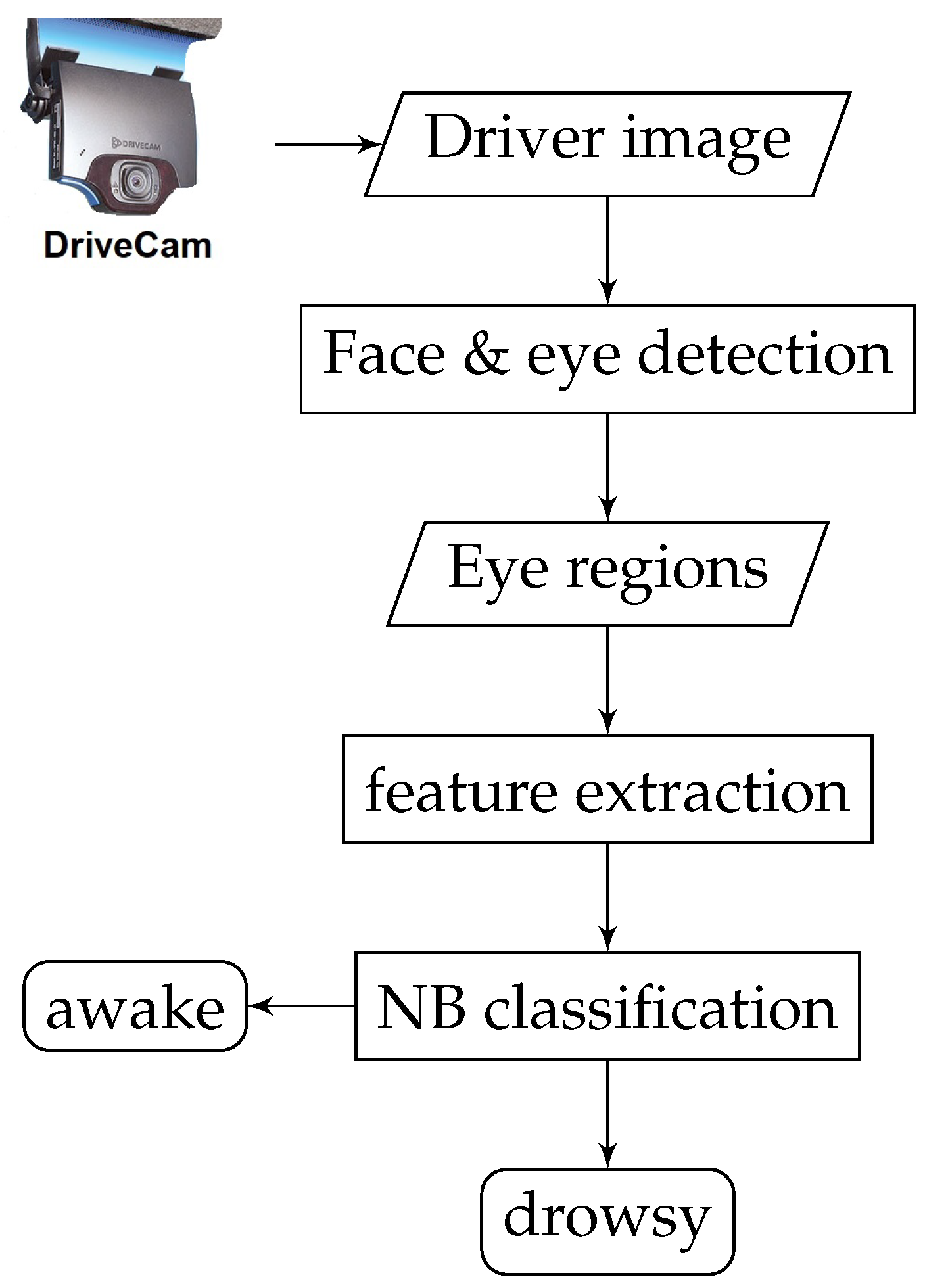
* **Classifier will Categorize whether Eyes are Open or Closed**

The Classifier will detect the eye aspect ratio to whether the eyes are open or not. This is done by a simple Euclidean formula.

* **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. We are drawing the result on the screen using **cv2.putText()** function which will display real time status of the person.

**WORKFLOW-**

****

**BIBLIOGRAPHY-**

[www.stackoverflow.com](http://www.stackoverflow.com/)

[www.dlib.net](http://www.dlib.net/)

[www.github.com](http://www.github.com/)