Driver Drowsiness Detection

Using Neural Networks & Computer Vision

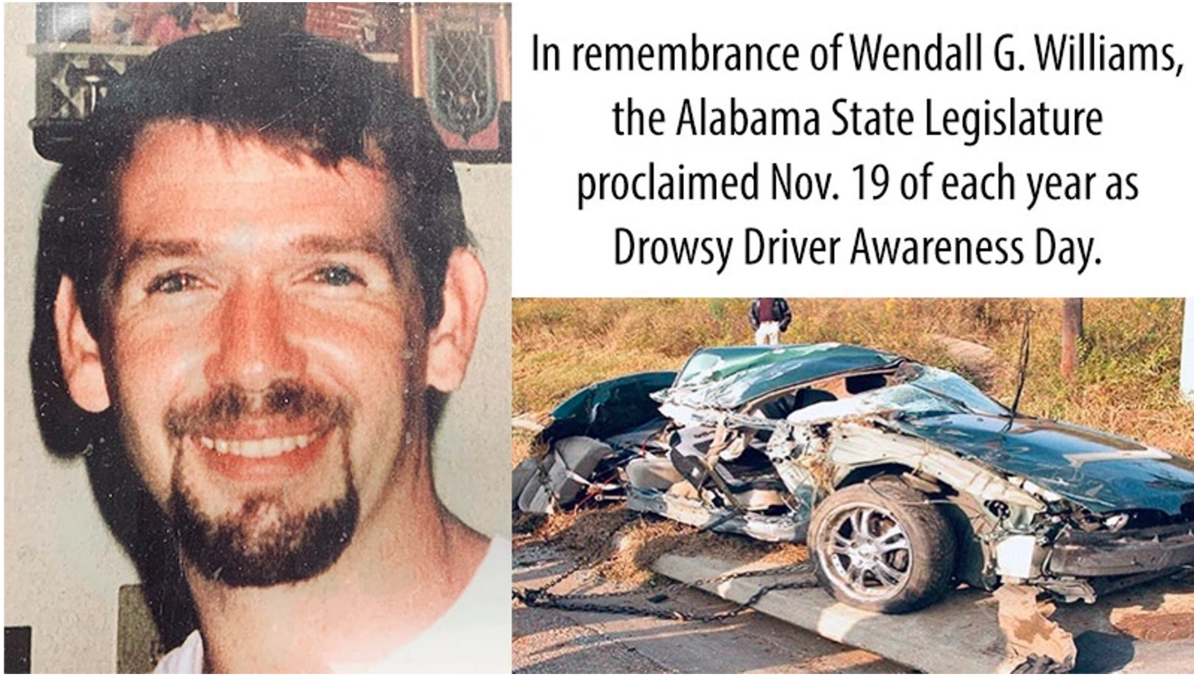
Jahnavi Bonagiri | Tobin Cherian | Evangeli Silva | Andrew Okoro

Overview

It is estimated that in 2017, 91,000 police-reported crashes involved **drowsy drivers**. These crashes led to an estimated 50,000 people injured and nearly 800 deaths. (NHTSA, 2019)

Monitoring facial expressions such as head movement and **eye blinking** can help diagnose if a driver is drowsy.

To prevent accidents, it is important to come up with **detection systems that can alert if the driver is drowsy**.

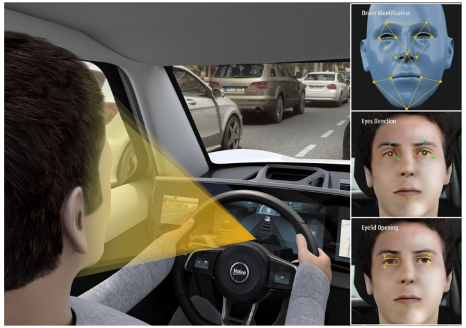


Objective

To develop a system that utilizes **non-intrusive** methods to accurately determine driver state in **real-time** and generate effective and acceptable **warnings** to increase driver alertness and safety.

Various Approaches

1. **Visual monitoring of the driver**
   Optical sensor to detect head and/or eye gestures
2. **Lane deviation monitoring**
   Sensor monitors vehicle
3. **Head nod monitoring**
   Sensor to detect head movement
4. **Activity monitoring**
   Driver wears watch-type sensor that tracks of motion
5. **Mental Reaction Time**
   Measures driver's reaction time to random warning



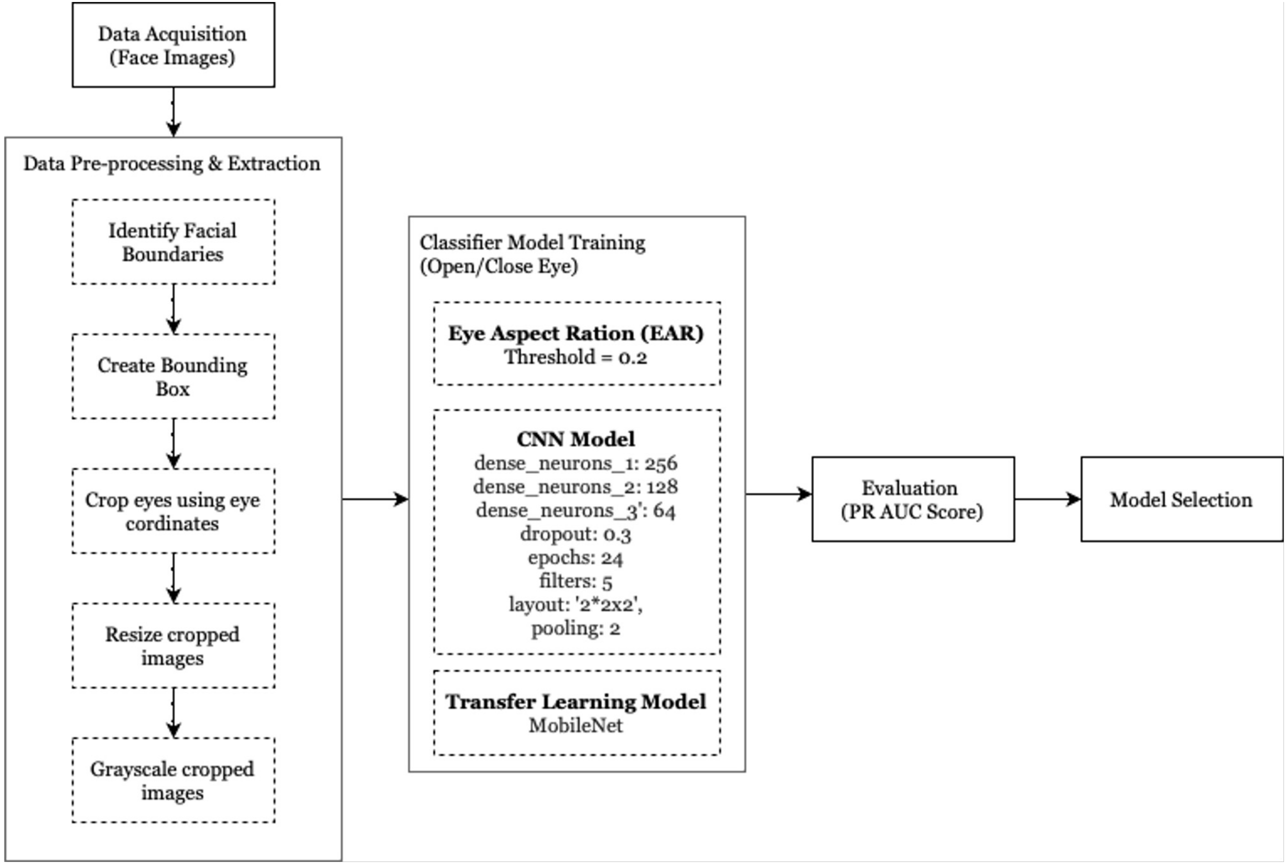
Our Approach

Monitor facial expressions, specifically eye closure time

Involves 2 phases:

1. Open/Close Eye Detection
2. Real-time Drowsiness Detection

Phase 1: Open/Close Eye Detection



Data Sources

“Labeled Faces in the Wild” (link)

Univ. of Massachusetts Amherst

18,983 photos of people with | both eyes open

“Closed Faces in the Wild” (link)

Nanjing Univ. of Aeronautics and Astronautics

1192 photos of people with | both eyes closed

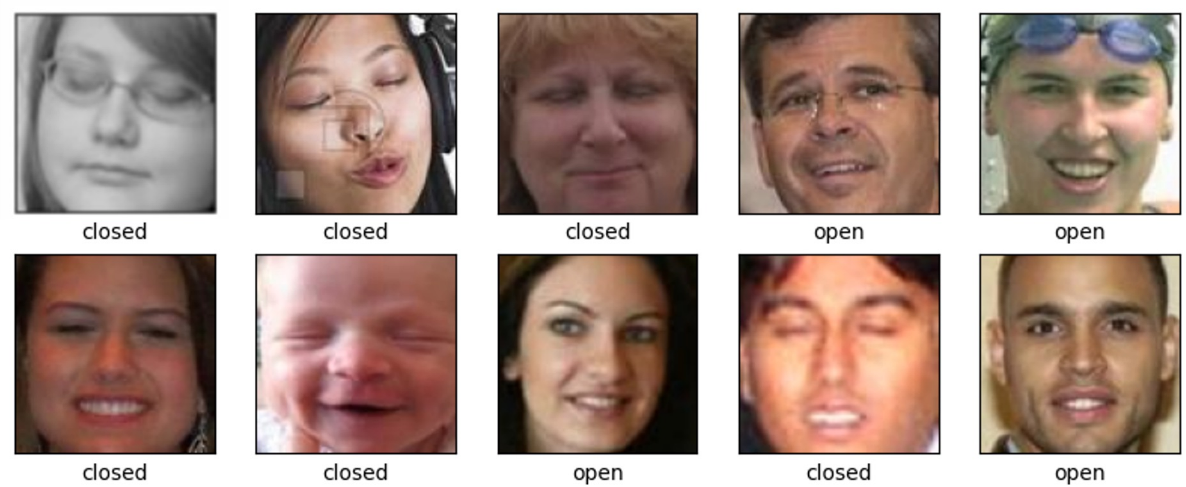
Kaggle Dataset (link)

Drowsiness Detection

Professionally taken close ups | images of 2,000 open and 2,000 closed eyes

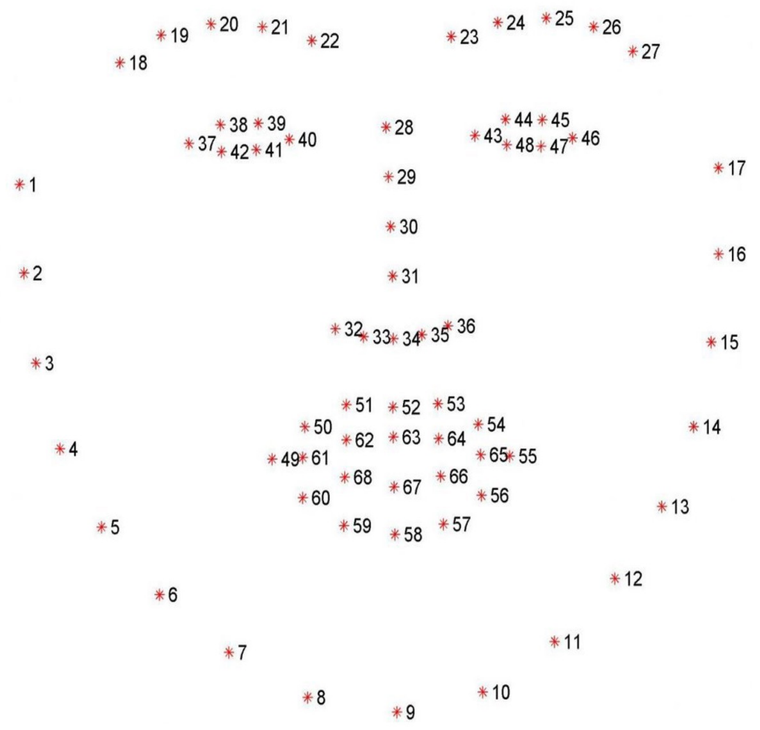
Data Forms

1. Gray scale
2. RGB
3. High/Low quality
4. Male/Female
5. With glasses
6. Diverse facial structures
7. Different angles



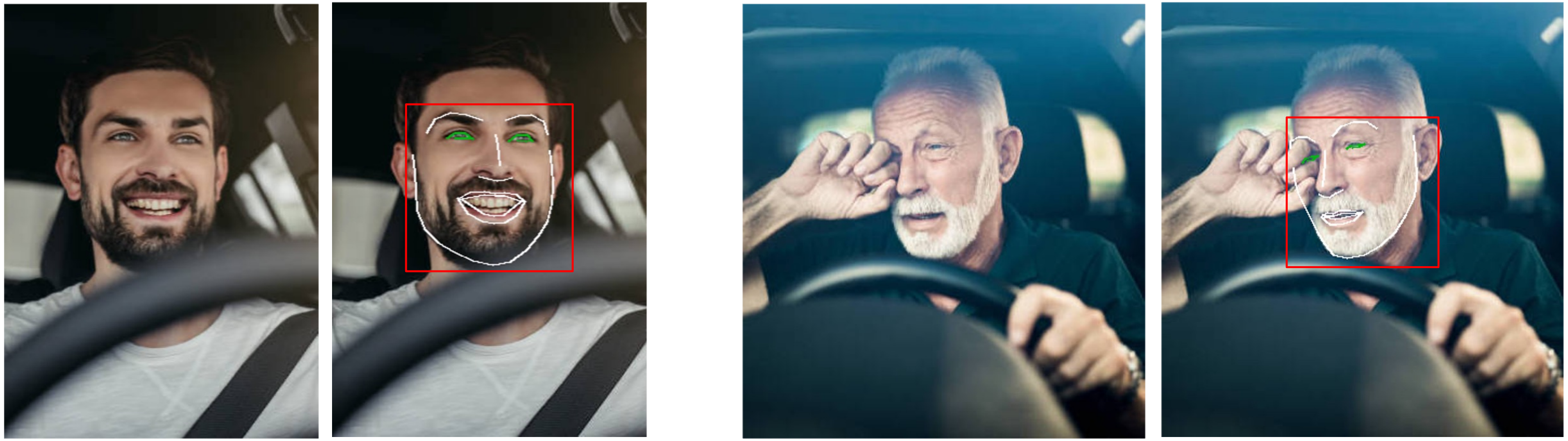
Pre-processing (Step 1: Identify Facial landmarks)

- Used **face_recognition library** to detect the key facial structures on the face ROI.
- face_recognition is built using **dlib**'s state-of-the-art face recognition and claims to have an **accuracy of 99.38%**.
- dlib's implementation of facial landmark detector uses a training set of labeled facial landmarks on an image. These images are manually labeled, specifying **68 (x, y)-coordinates** of regions surrounding each facial structure.
- Given this training data, an ensemble of **regression trees** are trained to **estimate the facial landmark positions** directly from the pixel intensities themselves.



Visualization of the 68 facial landmark coordinates from the iBUG 300-W dataset

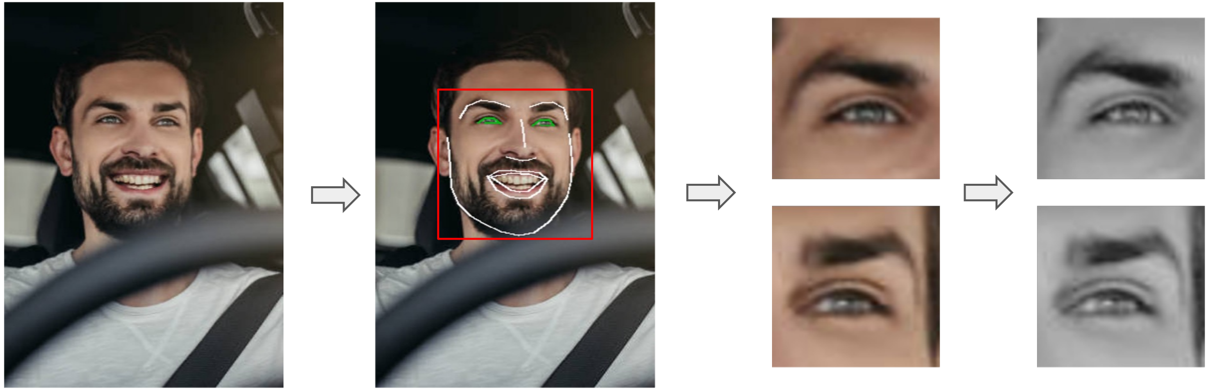
Pre-processing (Step 1: Identify Facial landmarks)



Facial structure coordinates represented as a line

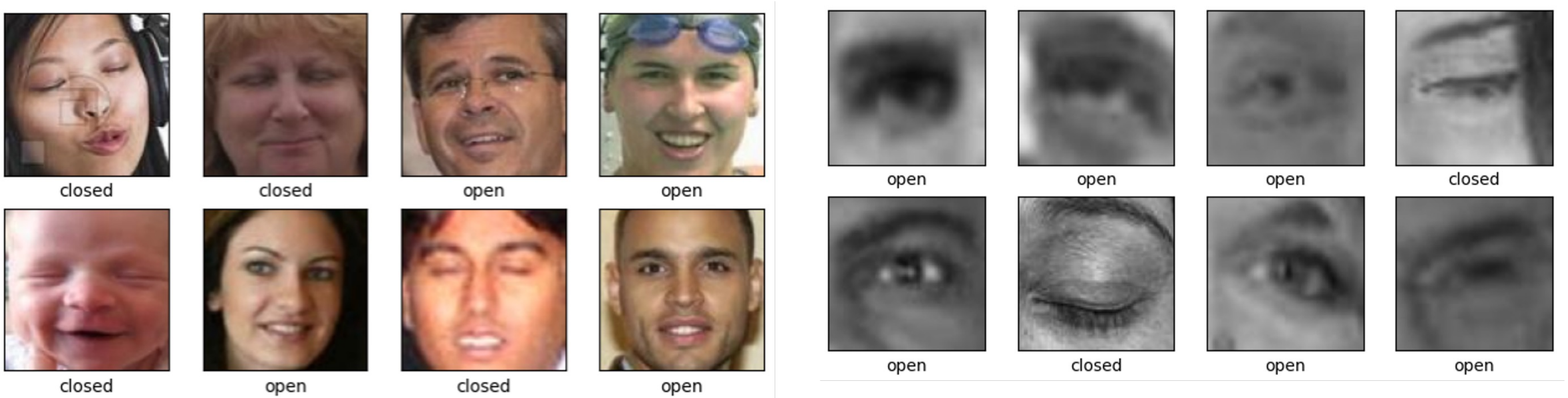
Pre-processing (Step 2: Crop Eye & Refine Further)

- Cropped images, based on the (x,y)-coordinates of the eye **(right eye points: 37–42, left eye points: 43–47)**.
- To make sure the full eye is captured, the coordinates of a square that has 50% cushion were added to the axis
- Finally, resized & gray scaled cropped images



A sample image after pre-processing

Pre-processing (Step 2: Crop Eye & Refine Further)



Sample images before & after pre-processing

Model Training (Open/Closed Eye Detection)

We explode the following methods:

a. Mathematical approach using **eye aspect ratio (EAR)**

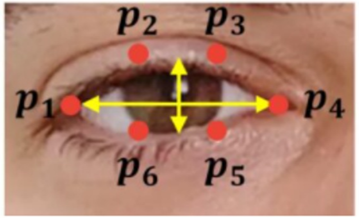
b. Building a **CNN model** from scratch for Image Classification

c. Transfer Learning (Image classification with transfer learning by fine-tuning a pretrained CNN model)

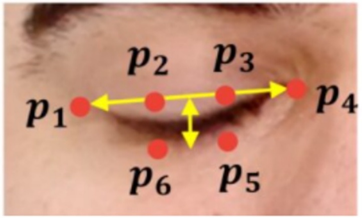
Approach 1: Eye Aspect Ratio (EAR)

Computed based on the (x,y)-coordinates of the eye

P2, P3, P5, and P6 key points are the value of longitudinal coordinate, P1 and P4 key points are the value of horizontal coordinate.



Open eye will have more EAR



Closed eye will have less EAR

EAR is calculated using the <u>euclidean distance</u> between key points around the eye.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

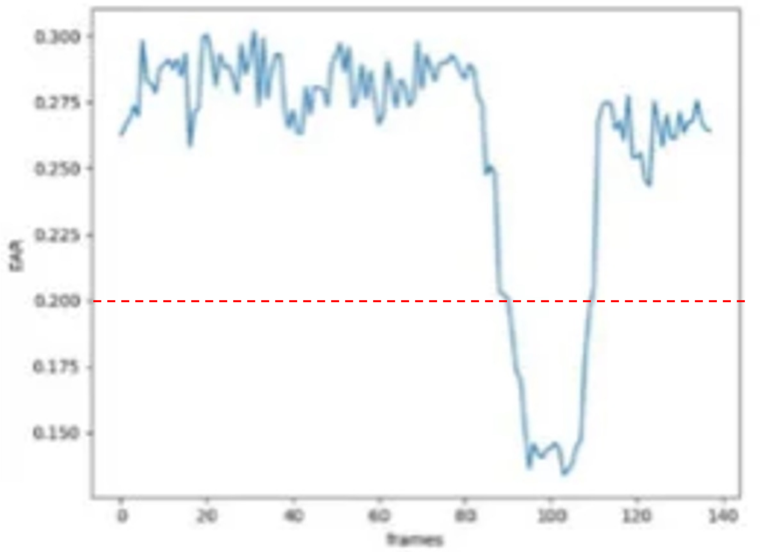
Approach 1: Eye Aspect Ratio (EAR)

According to T Zhu et al (2022), when **EAR > 0.2**, it is identified as an **open eye**, else a closed eye.

This was adopted to detect open/closed eye.

Observation:

- EAR fluctuates greatly when given an image of an eye with glasses



Approach 2: Convolutional Neural Network (CNN)

Hyper-parameter Tuning

Around 100 models were trained & evaluated varying the parameters

Based on Grid Searching using k-fold cross validation (k = 3)

1. Convolutional Layers (Number of layers: 1, 2, 3, 4, **5**)
2. Filters (Number of filters: 8, 16, **32** | Size of filters: **3x3**, 5x5)
3. Pooling (Pooling of features 2x2: **Yes**, No)
4. Dense Layers (Number of Layers: 1, 2, 3 | Neurons: **64**, **128**, **256**, 512, 1024)
5. Dropout: (Percentage: 10%, 20%, **30%**, 40%, 50%)

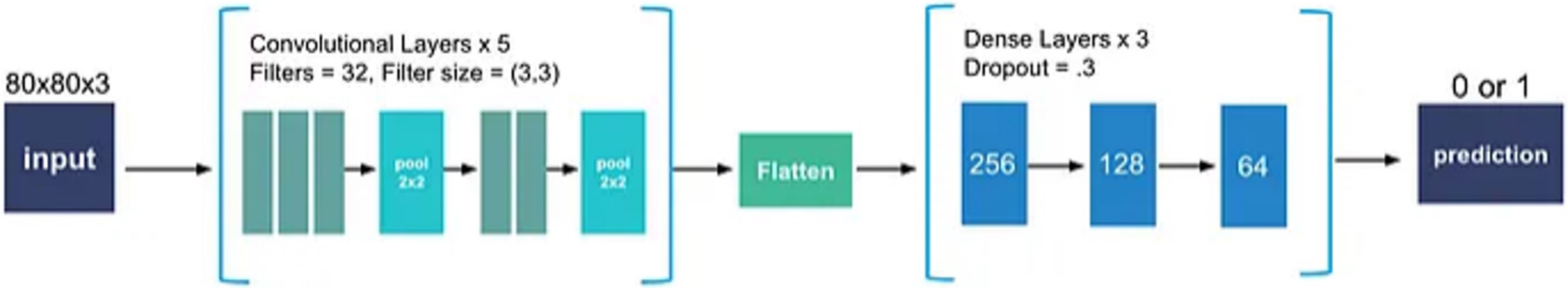
Approach 2: Convolutional Neural Network (CNN)

Scoring the Model (Top 5 scores)

Dense Layer 1 Neurons	Dense Layer 2 Neurons	Dense Layer 3 Neurons	Dropout Rate	Number of Iterations	Input Filters	Shape of Filters	Pooling Layer	Average Test Score
256	128	64	0.3	24	32	3x3	Yes	0.9631
256	512	1028	0.4	30	32	3x3	Yes	0.9614
256	512	1028	0.3	30	64	3x3	Yes	0.9610
128	128	128	0.1	10	32	5x5	No	0.9123
128	128	128	0.1	10	16	3x3	No	0.9028

Approach 2: Convolutional Neural Network (CNN)

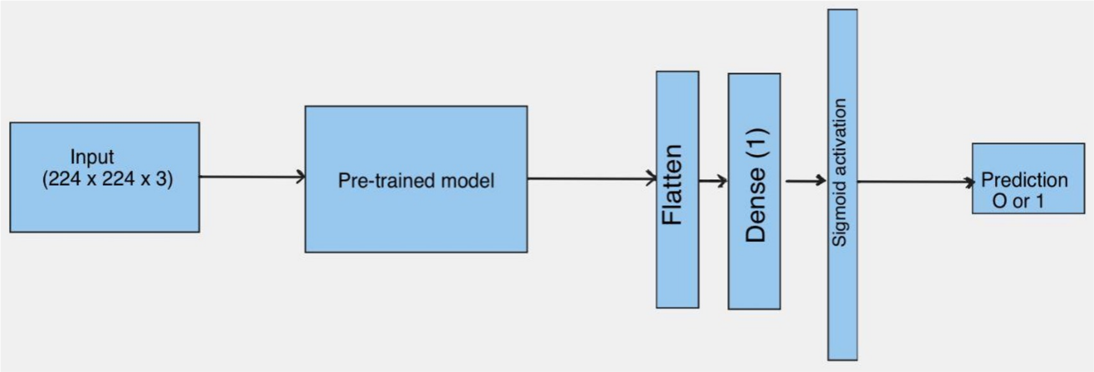
Final Network Architecture



- Convolutional Layers: **'ReLU' activation function** (helps to prevent the exponential growth in the computation required to operate the neural network)
- Output Layer: **'Sigmoid' activation function** (maps input values to a value between 0 and 1)
- Model Compilation using **'Adam' optimizer** (have faster computation time, and require fewer parameters for tuning)
- **24 Epochs** (diminishing returns after 20 epochs)

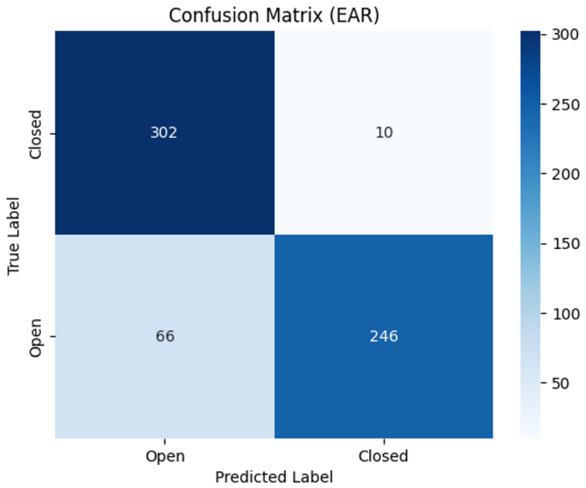
Approach 3: Transfer Learning

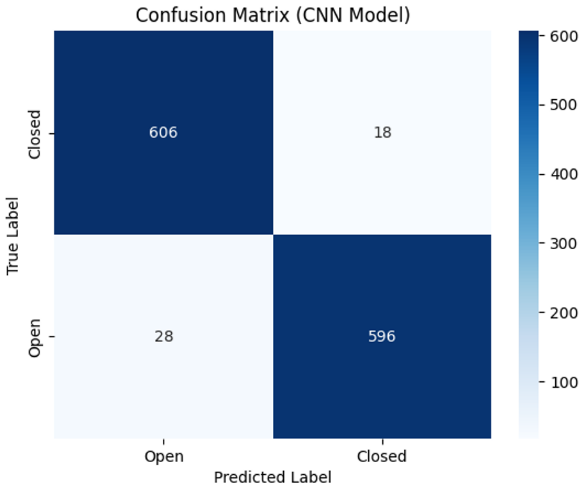
Image classification with transfer learning by fine-tuning the pretrained MobileNet CNN model

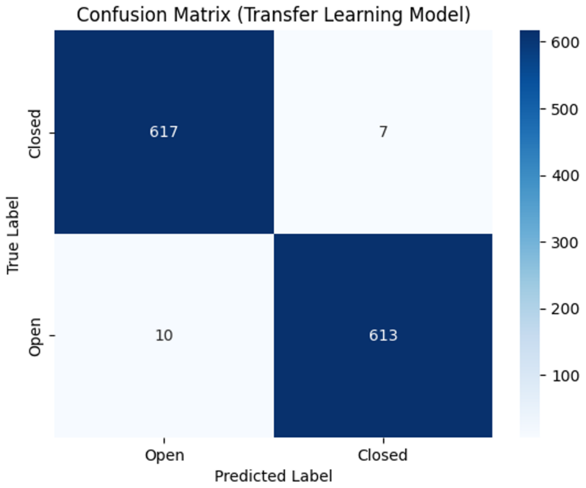


- MobileNet is pre-trained on MobileNet dataset
- Removed the output layers of the model
- Add layers: flatten and dense
- Sigmoid Activation function for the binary classification
- Binary cross entropy loss function
- Optimizer is Adam
- Early_stop and callbacks to monitor when the val_loss is not improving as the number of epochs increased.

Model Evaluation: Confusion Matrix







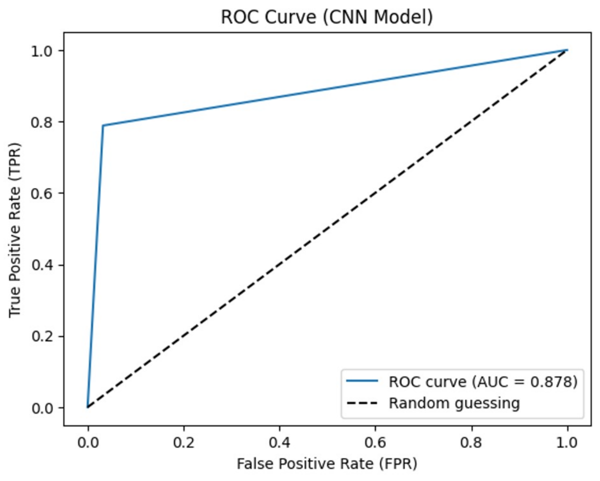
Evaluation Matrix

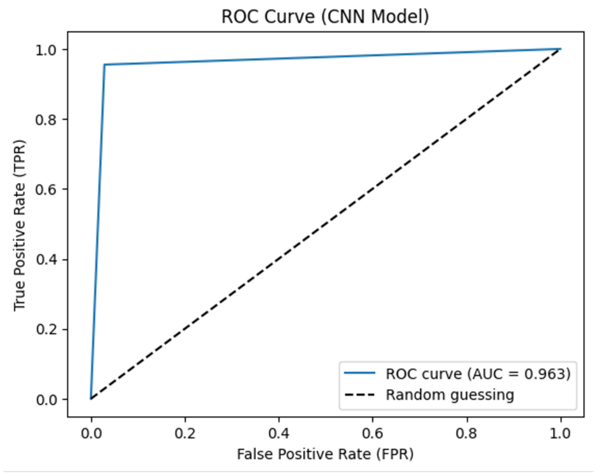
Metrics	EAR Based Classification (Threshold = 0.2)	CNN Model	Transfer Learning
Accuracy	87.82%	96.31%	98.6%
Precision	96.1%	97.07%	98.87%
Recall	78.84%	95.51%	98.39%
F-Measure	86.62%	96.28%	98.63%

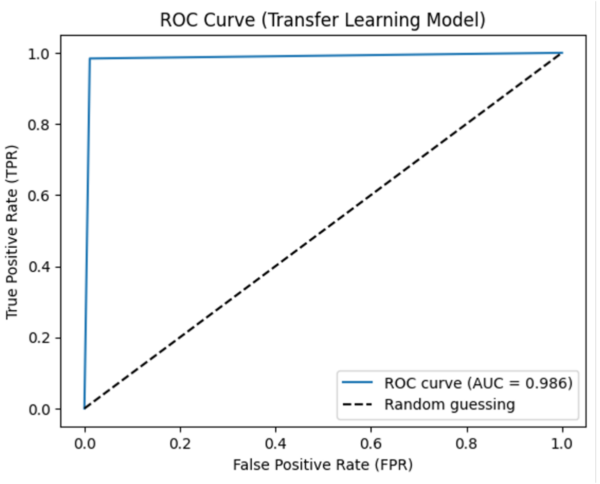
Model Evaluation: PR AUC Score

- Our primary metric was **the Precision-Recall Area under the Curve score (PR AUC)**. The “baseline” score for this metric is 0.5 for both precision and recall, and in a perfect classifier it is 1.
- The PR AUC score places the **most importance on how well we predict positives (a driver is asleep)**, which is more important to us than predicting negatives (a driver is awake).
- The area under the curve represents the proportion of true positives to false positives and false negatives.

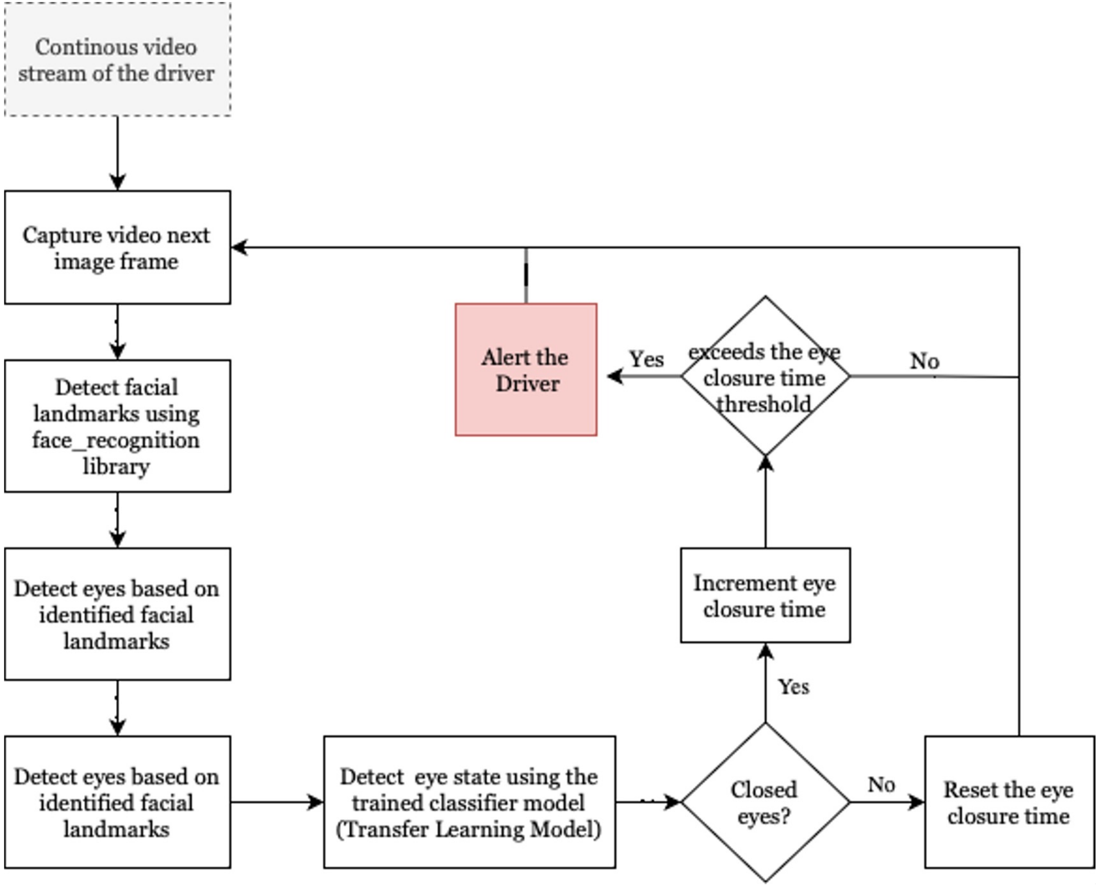
Model Evaluation: PR AUC Score







Phase 2: Real-time Drowsiness Detection



Phase 2: Real-time Drowsiness Detection

Eye Closure Time

- According to the Journal of Safety Research in 2009, reported that the mean duration of eyelid closure prior to a simulated crash was 3.79 seconds, and that eyelid closure durations of 2 seconds or longer were highly indicative of drowsiness.
- Also, the Journal of Sleep Research in 2018 found that a duration of eyelid closure of 2 seconds or more was associated with increased levels of drowsiness and impaired driving performance
- Hence, the **closure time of 2 seconds** was adopted to detect drowsiness

Limitations and Improvements

- Proximity to camera: Our system may not work properly when the camera is far from the person.
- The system is sensitive to the lightening condition in the environment.
- Apply data augmentation by applying transformations like flip, crop, rotations, to the data set, this will increase the size of the dataset and make the model more robust to variations in the input.

Demo

References

1. https://towardsdatascience.com/drowsiness-detection-with-machine-learning-765a16ca208a
2. A. Mittal, K. Kumar, S. Dhamija, and M. Kaur, "Head movement-based driver drowsiness detection: A review of state-of-art techniques," In 2016 IEEE International Conference on Engineering and Technology (ICETECH), Coimbatore, India, 2016, pp. 903-908.
3. Q. Ji and X. Yang, "Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance," Real-Time Imaging, vol. 8, no. 4, pp. 357–377, 2002.
4. https://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_de tection.html
5. A Robust Algorithm for Eye Detection on Grey Intensity Face without Spectacles- JCS&T Vol. 5 No. 3
6. Karamjeet Singh,Rupinder Kaur,"Physical and Physiological Drowsiness Detection Methods", IJIEASR, pp.35-43,vol.2,2013.
7. Jarek Krajewski,David Sommer,"Steering Wheel Behavior Based Estimation Of Fatigue", Fifth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design, Germany.