Team Procyon

CE 784A Project Report: Semester 2022-23 (II)

Evaluating Various Deep CNN (convolutional neural networks) Architectures for Driver Drowsiness Detection

Avadhi Jindal (19816204), P. Bhakthi Reddy (190604), Shubhi Kant (19807836)

Abstract

Driver drowsiness is one of the major causes of road accidents. In this paper we attempt to assess the performance of various deep CNN architectures for driver drowsiness detection. This paper compares VGG16, Resnet50 and our custom DDN net performing a binary classification task. The evaluation metric used is F1 score. Besides achieving high accuracy, some of the major parameters considered for its deployment are its size and amount of computing power used by the architecture. In this paper we attempt to create a CNN architecture which is computationally optimal (less prediction time over CPU and small sized so easily loadable in RAM) and hence is real-time deployable.

Keywords: Type your keywords here, separated by semicolons ; Logistic Regression, SVM (Support Vector Machine), CNN, Drowsiness Detection, VGG16, Resnet50, DDN

1. Introduction

Driver Drowsiness is one of the significant causes of road accidents. It is linked to 40% of car accidents in India (CRRI, 2019). In the USA, about 100,000 drowsiness-related crashes are related yearly, out of which 50,000 lead to injuries and 800 are fatal. A drowsiness detection system could help increase awareness among drivers and timely warn them if they show signs of fatigue.

There have been several studies on driver drowsiness detection. The most commonly used machine learning methods were logistic regression, SVM and CNNs. In this paper, we attempt to implement and compare these models.
Drowsy driving is a leading contributor to the tragic loss of life that occurs on India's roads on an annual basis, with thousands of lives being lost as a direct result of drivers falling asleep while driving. The problem is the same in different regions of the world, as evidenced by the fact that numerous accidents take place as a direct result of drivers being overly tired. A sleepiness detection system for drivers is an absolute necessity in light of the fact that automobile accidents continue to be one of the most major threats to public health on a global scale, accounting for millions of annual deaths and injuries.
Many automakers have created drowsiness detection systems that employ cameras and various sensors to inform drivers when they show signs of becoming drowsy. These safety features are largely found in premium cars; therefore, drivers hardly ever use them.
Recent developments in machine learning and deep learning have increased the possibility for numerous applications to create widely applicable, low-cost real-time driver sleepiness detection systems. This paper proposes a method that utilizes video datasets to extract essential features. The data is normalized and classified using different learning algorithms, including convolutional neural networks.
In summary, the paper proposes a system to detect driver drowsiness and prevent accidents caused by fatigue.

2. Literature Review

2.1. Shallow machine learning algorithms

S. Mittal et al. suggest use of 68 facial landmarks for derivation of mouth aspect ratio, eye aspect ratio, pupil circularity and mouth over eye ratio. The classification is then done using K-Nearest Neighbour, Naïve Bayes, Logistic Regression, Decision Trees, Random Forest, XGBoost, MLP and 1-D CNN. The highest accuracy is received by Logistic regression model with an ROC of 0.78772.

A. Kumar and R. Patra used 24 facial landmarks for feature extraction. The features computed are eye aspect ratio, mouth opening ratio and nose length ratio. After this adaptive thresholding is used to get the time period during which the driver shows symptoms. Bayesian classifier, Fisher's linear discriminant analysis (FLDA) and Support Vector Machine (SVM) with linear kernel are used for classification.

In S. Hu and G. Zheng paper, SVM is used on eyelid movement parameters like blink duration, lid closure speed, peak closing velocity peak opening velocity and delay of eyelid reopening. Using these parameters three classes were predicted: alert, sleepy and very sleepy. 100% accuracy was received for very sleepy class while 83.33% was received for alert and 86.67% for sleepy.

2.2. Deep Neural Network based approaches

B. Reddy et al. use four stream and two stream deep neural networks taking 3 channel cropped images of the both the eyes, face and mouth as inputs. Each stream of the model is similar to Alex Net architecture. The streams are connected through fully connected layers. These networks predict three classes normal, yawning and drowsy respectively. They receive a test accuracy of 93.8% with two-stream model taking right eye and left eye as input. Authors of [5] make use of a CNN on 68 facial landmarks. The model is compact as compared to traditional architectures like VGG-16 and AlexNet. They receive an accuracy of 83.3% on binary classification.

Driver drowsiness is one of the major causes of road accidents and fatalities. Researchers have proposed various techniques for detecting driver drowsiness to prevent these accidents. In this literature review, we will discuss recent techniques for driver drowsiness detection.

One of the most common approaches for driver drowsiness detection is using image processing techniques machine learning. Mittal et al. (2021) proposed a driver drowsiness detection system that uses machine learning and image processing to analyze the driver's eye and face movements. They used a deep learning algorithm to analyze the driver's eye and face movements and detected the drowsiness based on the changes in facial features and eye closure patterns.

Kumar and Patra (2018) proposed a monitoring system for driver drowsiness that analyses visual behavior and uses machine learning techniques. They used a camera to monitor the driver's face and detect drowsiness based on the changes in facial expressions and eye movements.

Reddy et al. (2017) proposed a instantaneous driver drowsiness detection system for a fixed system using compression model of deep neural networks. They used a convolutional neural network (CNN) to analyze the driver's facial features and eye movements and compressed the model for real-time processing on an embedded system.

Hu and Zheng (2009) proposed a driver drowsiness detection system that uses eyelid-related parameters and SVM. They extracted the features from the driver's eye movements, like eye-opening time, eye closing time and eyelid movement, and used SVM to classify the drowsiness state of the driver.

Jabbar et al. (2020) proposed a driver drowsiness detection model that uses CNN techniques for an Android application. They used a CNN model to analyze the facial features and eye movements of the driver and developed an Android application to alert drivers when they become drowsy.

In conclusion, driver drowsiness detection is an emerging area of research, and various new techniques are being proposed by researchers for this purpose. Machine learning and image processing techniques like CNN and SVM are commonly used for driver drowsiness detection. These techniques have shown promising results and have the potential to prevent road accidents caused by driver drowsiness.

3. Overview of the Data

The Data used in this study is constituted by 227 x 227-pixel RGB images. There are 41,793 files in total, out of which 22,348 are samples of drowsy drivers and 19,445 are samples of non-drowsy drivers. The pictures are cropped faces of the drivers from captured frames of the Real-Life Drowsiness Detection Dataset (RLDD) maintained by University of Texas at Arlington. The

RLDD consists of more than 30 hours of video data, from which frames were extracted using VLC algorithm and then Viola-Jones algorithm was used to crop region of interest. Some of the images belonging to both categories are shown in Fig. 1.

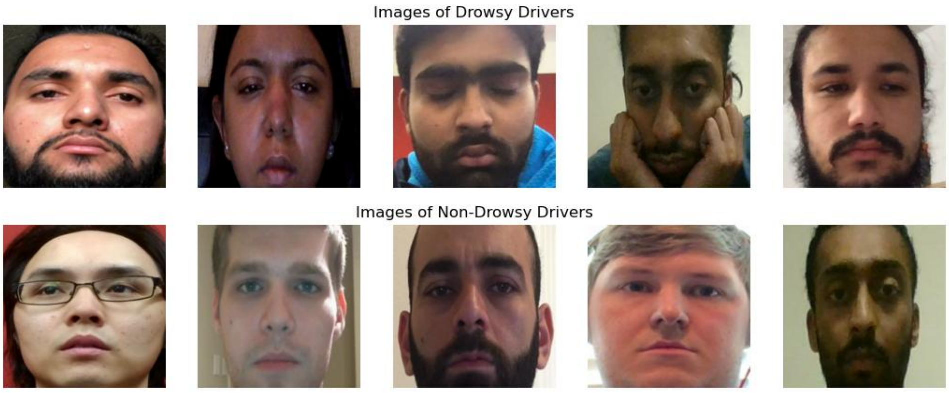


Fig. 1: Images of drowsy and non-drowsy drivers from the dataset

For training and testing purposes we first divide the dataset first into two parts in a ratio of 90:10 making a train and test set. We apply the following augmentations to the train set:

1. Random Resized Crop: to crop different parts of images and resize then to 224x224, so that they could be used as input to the models, this would allow model to learn from different parts of the face.
2. Random Rotation: This would rotate image at a randomly chosen angle between ±30°.
3. Random Horizontal Filp: This would randomly horizontally flip the images with a probability of 0.5.
4. Color Jitter: To randomly change the saturation, contrast, brightness and hue of an image.
5. Random Grayscale: to convert rando images to grayscale at a probability of 0.1.
6. Normalize: All the three input channels of image are normalized with 0.5 mean and 0.5 standard deviation following: output [channel] = (input [channel] - 0.5) / 0.5.

The train set is further divided into training and validation set during training in the ratio of 80:20. Some of the images from the train set after the augmentations are displayed in Fig. 2. The test set images are just resized to 224 x 224 from 227 x 227 and normalized with 0.5 mean and 0.5 standard deviation.

Fig. 2: Images used for training, label 0: Drowsy and label 1: non-Drowsy

4. Methodology

4.1. Architecture

The overall architecture of the Drowsiness-Detection-Net (DDN) is shown in Fig. 3. It essentially consists of three blocks consisting of a convolutional layer with ReLu (Rectified Linear Unit) activation followed by a batch normalization a max-pool

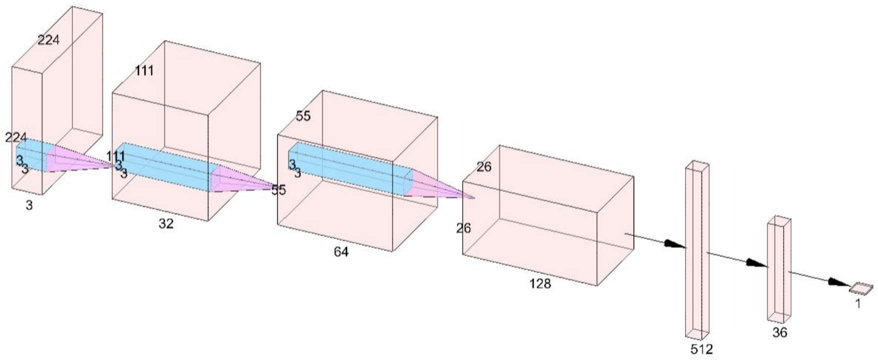


Fig. 3: Architecture of Drowsiness-Detection-Network

and a dropout layer. After three such blocks the output is flattened and fed to dense layers. Details about the layers can be seen in Fig. 4. The architecture is implemented in the file net.py.

4.2. Training

DDN was trained for 100 epochs using stochastic gradient descent with a momentum of 0.9 and learning rate of 0.001 to optimize binary cross entropy loss. The metric used for calculating accuracy was F1 score.

$$BCE\ Loss = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i .\ \log\left(p(y_i)\right) + \left(1 - y_i\right) .\ \log\left(1 - p(y_i)\right)\right)$$

Where N is the number of images in a batch, y_i is the prediction for i-th sample and $p(y_i)$ is the probability that belongs to the first class and $(1 – p(y_i))$ is the probability that y_i belongs to the other class.

F1 score can be calculated using the following equations:

$$precision = \frac{\#\,of\ true\ positives}{\#\,of\ true\ positives + \#\,of\ false\ positives}$$

$$recall = \frac{\#\,of\ true\ positives}{\#\,of\ true\ positives + \#\,of\ false\ negatives}$$

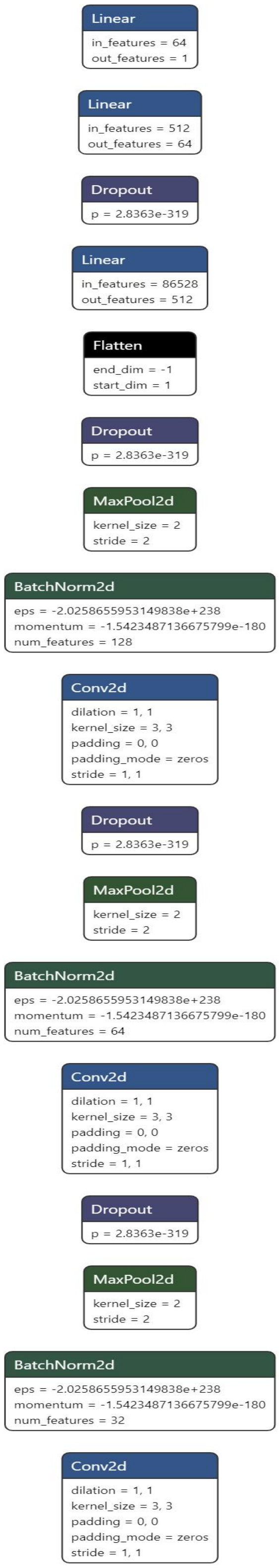
$$F1\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

The training and validation curves for the DDN can be seen in Fig. 5.

4.3. Training of VGG16 and Resnet50

Pre-trained VGG16 and Resnet50 are used for transfer learning, only the linear fully connected layers of the models are trained. A Fully connected layer is added at end of Resnet50 to get 1 output neuron from 1000 neurons. Similarly, for VGG16 one linear layer converts 4096 output neurons to 1 output neuron. The architectures of Resnet50 and VGG16 are shown in Fig. 6 and Fig. 7. Both VGG16 and Resnet50 were trained over 50 epochs using stochastic gradient descent with momentum of 0.9 and learning rate of 0.001 to optimize binary cross entropy loss. The training and validation curves for VGG16 and Resnet50 are shown in Fig. 8 and Fig. 9.

Fig. 4: Layers of the Drowsiness Detection Network



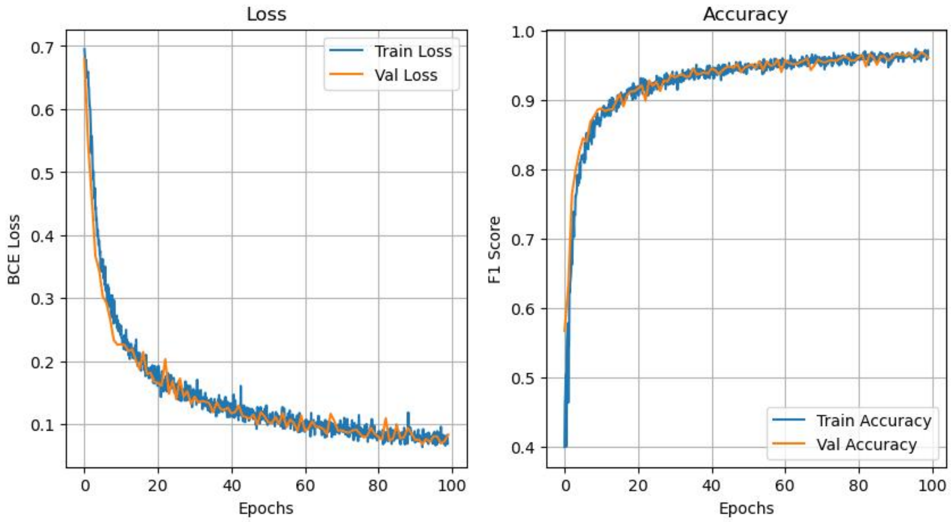


Fig. 5: Training and validation curves for drowsiness detection network

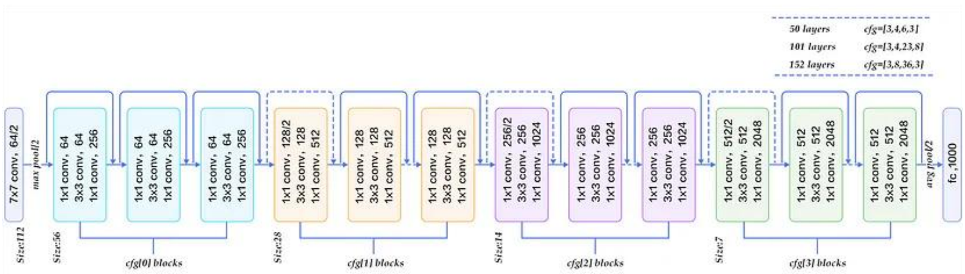


Fig. 6: Resnet50 architecture, source: https://blog.devgenius.io/resnet50-6b42934db431

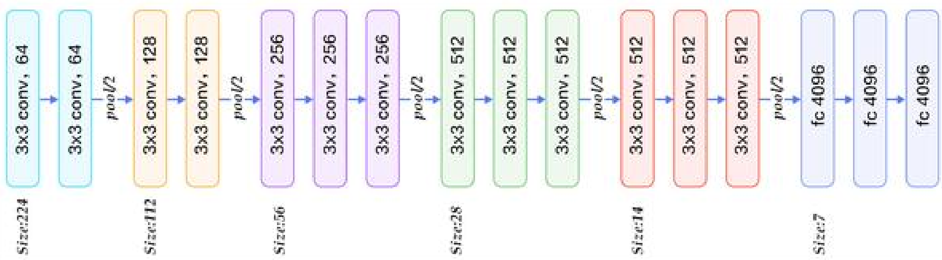


***Fig. 7: VGG16 architecture, source:* https://iq.opengenus.org/vgg16/**

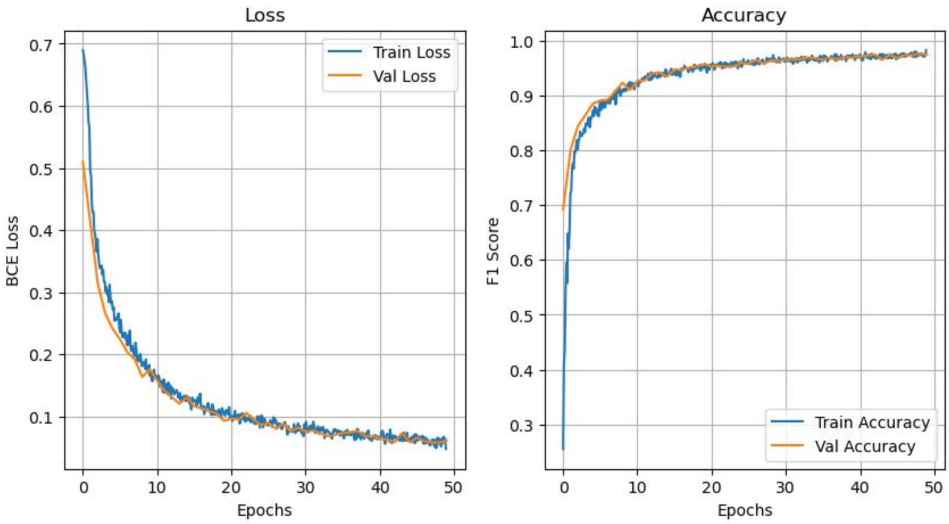


Fig. 8: Training and validation curves for VGG16

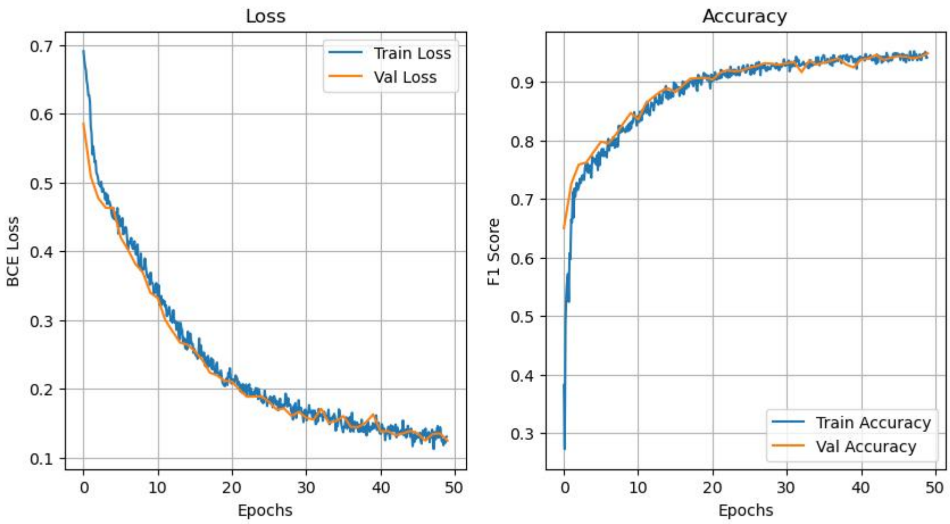


Fig. 9: Training and validation curves for Resnet50

5. Results

The results of training the three models and some of their characteristics are summarized in Table 1. As can be seen the prediction time over GPU for DDN net comes out to be 22.12 seconds for 4179 images. Thus, the average prediction time per image is 0.005 seconds.

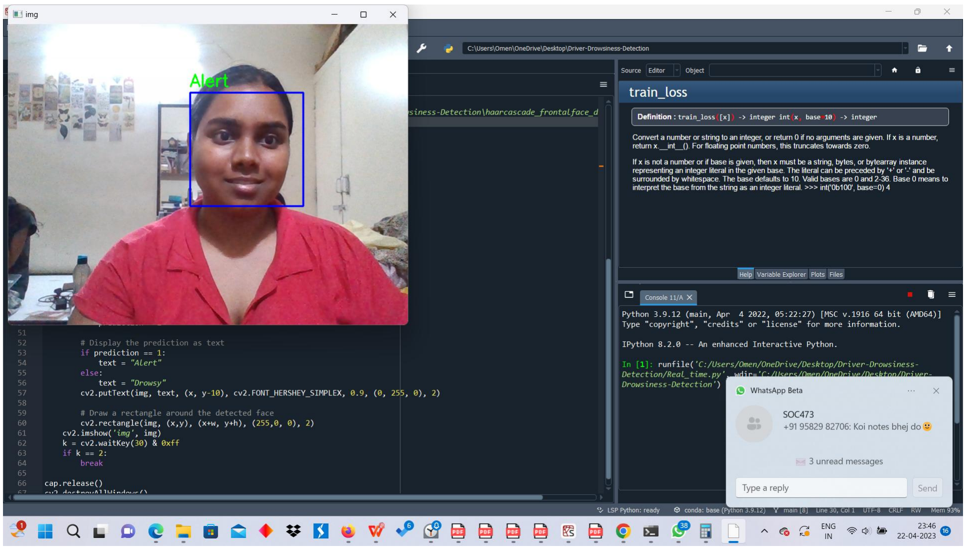
Model	F1 Score			# of trainable parameters	Prediction time over GPU P100 (For 4179 images)
	Training Set (All transforms applied)	Validation Set (All transforms applied)	Test Set (Only resized and normalized images)		
DDN	0.969	0.966	0.998	44,429,441	22.12
VGG16	0.974	0.976	0.996	119,549,953	28.87
ResNet50	0.941	0.943	0.965	2,050,001	22.08

Table 1: Results of training the models

6. Deployment

We deploy our model using OpenCV. The results for real-time predictions are shown in Fig. 10. OpenCV is used to capture frames from images from which face is extracted.

Pre-trained Haar Cascades model is used for detecting face in the images. The cropped areas are resized to 224 x 224. Pre-trained Harr Cascades is available in OpenCV, which is popular machine learning library. The inputs are then normalized and fed to Drowsiness-Detection-Network.



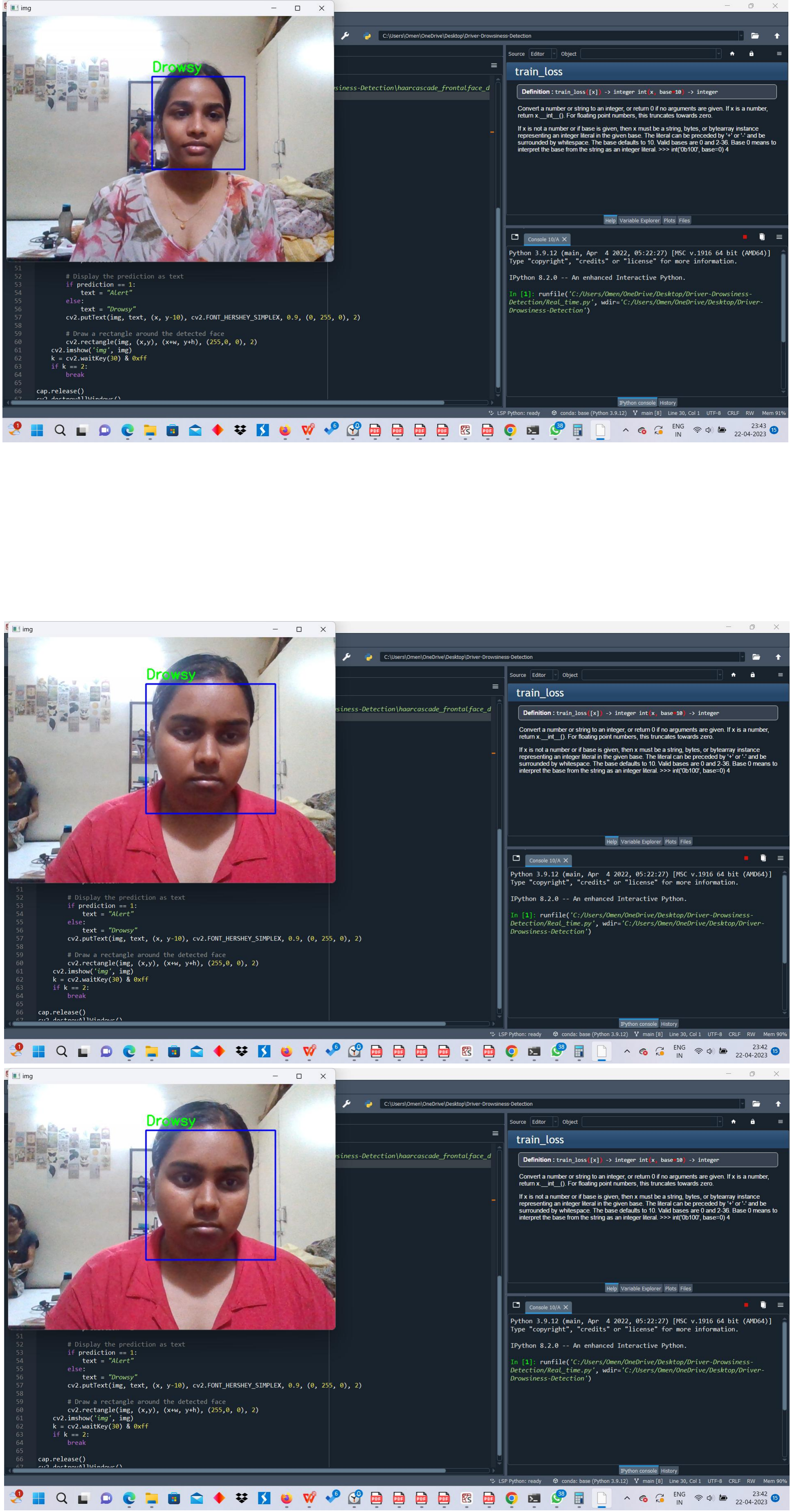


Fig. 10: Results of real-time predictions from the model

Work Contribution

This report was written by the group with equal efforts, hence the boundaries for sections are vague.

Avadhi Jindal: Methodology, Data overview, Literature review

Bhakthi Reddy: Introduction, Literature Review, Reference, Resnet and VGG16 training

Shubhi Kant: Literature Review, Methodology, DDN architecture and training, real-time implementation

References

[1] S. Mittal, S. Gupta, Sagar, A. Shamma, I. Sahni and N. Thakur, "Driver Drowsiness Detection Using Machine Learning and Image Processing," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2021, pp. 1-8, doi: 10.1109/ICRITO51393.2021.9596358.

[2] A. Kumar and R. Patra, "Driver drowsiness monitoring system using visual behavior and machine learning," 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 2018, pp. 339-344, doi: 10.1109/ISCAIE.2018.8405495.

[3] B. Reddy, Y. -H. Kim, S. Yun, C. Seo and J. Jang, "Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 2017, pp. 438-445, doi: 10.1109/CVPRW.2017.59.

[4] S. Hu and G. Zheng, “Driver drowsiness detection with eyelid related parameters by Support Vector Machine,” Expert Systems with Applications, vol. 36, no. 4. Elsevier BV, pp. 7651–7658, May 2009. doi: 10.1016/j.eswa.2008.09.030.

[5] R. Jabbar, M. Shinoy, M. Kharbeche, K. Al-Khalifa, M. Kirchen and K. Barkaoui, "Driver Drowsiness Detection Model Using Convolutional Neural Networks Techniques for Android Application," 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2020, pp. 237-242, doi: 10.1109/ICIoT48696.2020.9089484.