Driver's Drowsiness Detection System

Article · May 2023

CITATIONS
0

READS
365

2 authors:

Aayush Bhetuwal
Yeshiva University
6 PUBLICATIONS **1** CITATION
SEE PROFILE

Siddanta K C
Yeshiva University
6 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Driver's Drowsiness Detection System

Aayush Bhetuwal
Siddanta K.C.
Katz School of Science and Health
Yeshiva University
MS in Artificial Intelligence - AIM-5007-1
abhetuwa@yu.mail.edu
skc@yu.mail.edu

Abstract

Driver sleepiness is a serious issue that contributes to innumerable traffic accidents all around the world. A reliable driver sleepiness detection technology is needed to stop these accidents. Deep learning techniques have been widely used to create these strategies in recent years. This work focuses on methods for detecting driver drowsiness that use the Image-net and YOLOv5s models. Real-time detection features using the yolov5s models are also demonstrated in this study. In one of the image-net models, the implementation of these models yields the best accuracy of 96.8%.

1. Introduction

Most drivers are aware of the risks associated with texting while driving and driving while intoxicated, but many are unaware of the risks associated with drowsy driving. Drowsiness in this context refers to the situation in which the driver falls into a microsleep, which increases the risk of unconsciousness and accidents. The National Safety Council (NSC) estimates that each year, nearly 100,000 crashes involving drowsy driving result in 71,000 injuries and 1,550 fatalities. Similar to the previous study, the AAA Foundation for Traffic Safety calculated that between 16 and 21 percent of all fatal car accidents reported to the police involve drowsy driving. As a result, a system must be created to manage and keep an eye on these problems.

Convolutional neural networks (CNNs), a type of deep learning approach, have been found to be successful in creating driver drowsiness detection systems. CNNs, which have been used to create a variety of computer vision applications, including object identification and recognition, can learn intricate patterns and features from images. Driver drowsiness detection systems have largely been developed using pre-trained CNNs on massive image datasets like Imagenet.

Identifying and localizing items in a picture is the goal of object detection, another well-liked deep-learning method used in computer vision. YOLO (You Only Look Once) is one of the most popular object recognition models. It uses a single CNN to predict the type and location of objects in a picture. The most recent and effective iteration of the YOLO concept is called YOLOv5s.

The proposed driver sleepiness detection system in this study combines a pre-trained CNN from Imagenet with an object detection model from YOLOv5. To determine the degree of tiredness, the system examines the driver's facial features, including the eyes and face. The final few layers of the CNN and YOLOv5s models are adjusted to make them more suitable for the task of detecting driver fatigue. The effectiveness of the suggested system is assessed using a dataset of photos of drivers who are active and drowsy, and it is contrasted with other cutting-edge driver drowsiness detection systems. The device is a potential option for reducing accidents brought on by drowsy driving because of its accuracy and resilience. Additionally, the suggested system has real-time sleepiness detection capabilities, increasing its effectiveness in averting accidents.

2. Related Work

Several driver sleepiness detection systems have been created during the past few years using various machine learning algorithms.

2.1. Driver drowsiness estimation using EEG signals

The proportion of eyelid CLOSure (PERCLOS) is used as the ground truth of driver tiredness in their paper, which offers a novel dynamical modeling technique to assess the immediate level of driver drowsiness using EEG signals.

Early research by S. Arefnezhad et al., [1], focused on identifying driver drowsiness using physiological signals like electroencephalography (EEG) and electrooculography (EOG), with a binary classification accuracy of 90%. The use of EEG-based measures in driver sleepiness detection systems is made possible, they claimed, by the suggested method's robustness and dependability as a means of estimating drowsiness in real-time. These systems aren't appropriate for use in the real world and need more sensors.

2.2. A Deep Neural Network for Real-Time Driver Drowsiness Detection

Recently, deep learning techniques have been applied to driver drowsiness detection using camera-based systems. These systems detect drowsiness by analyzing facial features such as the eyes and face. Many studies have used convolutional neural networks (CNNs) pre-trained on large-scale image datasets such as Imagenet to develop driver drowsiness detection systems. For example, the study by Wang et al. [3] proposed a system that uses a deep neural network to analyze the driver's facial features, including the eyes and mouth, to detect drowsiness in real-time. In their model, the driver's faces were extracted from video frames as inputs and hence, the model achieved a competitive accuracy of 84.81%.

Deep learning methods have recently been used to identify driver drowsiness utilizing camera-based systems. These devices examine facial characteristics like the eyes and face to identify tiredness. Convolutional neural networks (CNNs) pre-trained on expansive picture datasets like Imagenet have been employed in numerous studies to create driver drowsiness detection systems. A system that uses a deep neural network to scan the driver's facial features, such as the eyes and lips, in order to identify tiredness in real time is one example from the paper by Wang et. al. [3]. They used the driver's faces as inputs in their model, which allowed them to reach a competitive accuracy of 84.81 percent.

2.3. Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images

The associated study discusses Elena M. et al.'s research "Driver Drowsiness Detection by Applying Deep Learning Techniques to Sequences of Images" [2]. In order to warn drivers of their tired state and stop traffic accidents, the study focuses on the creation of an ADAS (advanced driving assistance system) for detecting driver sleepiness. The suggested method makes use of 60-second image sequences that focus on the subject's face to perform non-intrusive fatigue detection in a driving context. With a focus on reducing false positives, the paper offers two alternative approaches for identifying sleepy driving. While the second solution collects numerical features from images using deep learning methods and feeds them into a fuzzy logic-based system, the first solution uses a recurrent and convolutional neural network. Similar accuracy levels of about 65% on training data and 60% on test data are attained by both strategies. The fuzzy logic-based system, on the other hand, stands out since it avoids sounding erroneous alerts and manages to attain a high specificity of 93%. The outcomes of this work are not very satisfactory, despite the fact that the suggested methods are promising and lay a strong platform for future work in driver sleepiness detection utilizing deep learning techniques.

2.4. Car Companies

Similar to this, numerous major automakers, including Mercedes-Benz and Land Rover, have built drowsiness detection technologies into their cars utilizing sensors that monitor the driver's facial expressions and movements, like head position and eye blink frequency. These devices analyze the gathered data using machine learning techniques and warn the driver if drowsiness is found.

3. Dataset

3.1. Data Collection

The dataset consists of 4560 Active images and 4560 Fatigue images which are collected from Kaggle.

3.2. Data Preparation

In the present study, the Yolov5s model was utilized as the primary model for detecting and localizing the driver's face and eyes within an image. This was done to enable further analysis and classification of the driver's level of drowsiness. To train the Yolov5s model, labeled images were required. Specifically, each image in the training dataset needed to be labeled with bounding boxes that define the locations of the objects or features of interest, such as the driver's face and eyes in the case of driver drowsiness detection. The label for each bounding box typically includes the coordinates of the box in the image, as well as the class label for the object or feature that the box encloses.

As there were only two classes of interest, i.e., active or fatigue images, a pre-trained face-net PyTorch model was used for extracting the bounding boxes. However, it was observed that some images were not suitable for training as some faces were covered, and some people's faces were less visible. As a result, the pre-trained face-net model failed to detect faces in some images, and those images were removed from the training dataset. Each image was also associated with a corresponding .txt file that contained its labels.

After preparing the dataset, the next step is to split it into training and testing data. The dataset is divided using an 80-20 ratio, meaning that 80% of the images and their corresponding labels are used for training the model,

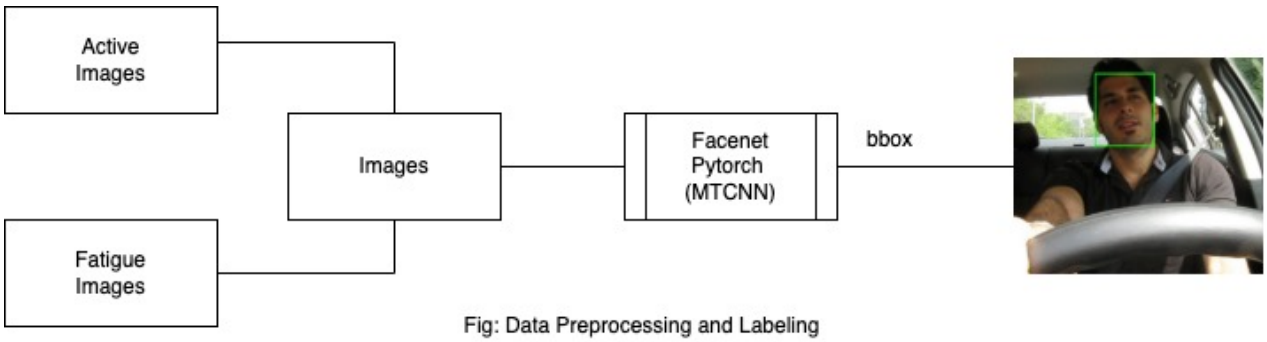

Fig: Data Preprocessing and Labeling

Table 1. Dataset

Dataset	Count
Active Subjects	4560
Active Labels	4560
Fatigue Subjects	4538
Fatigue Labels	4538

while the remaining 20% are used for testing the model's performance. The training and testing data are further organized into two separate folders, with each folder containing both active and fatigued images and their corresponding labels. By splitting the dataset into training and testing sets, the model can be trained on the training data and evaluated on the testing data. This process allows the model to learn from the training data and adjust its weights to accurately classify the testing data.

Table 2. Training

Training Dataset	Count
Images (Active and Fatigue)	6653
Labels (Active and Fatigue)	6653

Table 3. Testing

Testing Dataset Count	
Images (Active and Fatigue)	1792
Labels (Active and Fatigue)	1792

4. Method

A pre-trained face-net PyTorch model was utilized to extract the bounding boxes since there were only two classes of interest, active or fatigue photos. It was noted, nonetheless, that some photos were unsuitable for training since some features were obscured and others' faces were less obvious. As a result, some of the photographs were excluded from the training dataset since the pre-trained facenet model was unable to recognize faces in them. Additionally, each image had a corresponding.txt file that provided the labels for each image.

4.1. Motivation

The goal of this study is to use several models trained on the ImageNet dataset with YOLOv5s to increase the precision of driver sleepiness detection. The research aims to surpass 90% accuracy, which would be a major advance over earlier attempts. The authors have built and improved many deep learning models, such as VGG-16, ResNet50, and GoogleNet, to detect and categorize the degree of tiredness in drivers using sequences of photos with this goal in mind. They then employed YOLOv5s to locate the driver's face and eyes within the image, which was then used for additional analysis and tiredness degree assessment. The authors' goal is to identify which model produces the best results by contrasting which model works best for this specific application and improving the accuracy of driver drowsiness detection, thereby enhancing road safety.

4.2. Implementation Details

The model was trained and evaluated over the course of a few steps. First, the model was trained using the training dataset, and its performance was assessed using the validation dataset. A GPU was used to accelerate the training process, and the PyTorch CUDA library was used to enable GPU acceleration. An optimizer, a scheduler for the learning rate, and the number of epochs were all defined during the training process.

With a learning rate of 1e-4 and a weight decay of 5e-4, the Adam optimizer was employed. The learning rate scheduler was intended to reduce the learning rate by a factor of 0.50 if the validation loss did not decrease after 5 epochs. Each of the models had a different number of epochs, which were chosen depending on how well they performed on the validation dataset.

In conclusion, the definition of the optimization algorithm, the learning rate scheduler, and the number of epochs, which varied across different models, were all part of the training process. For GPU acceleration, the PyTorch CUDA library was used, and the Adam optimizer was used

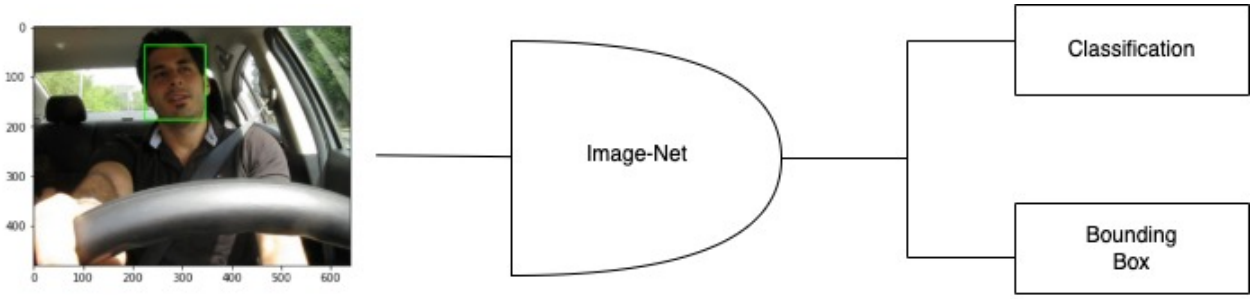

Figure 1. Finetuned ImageNet Models

with a particular learning rate and weight decay. The learning rate was modified if, after a predetermined number of epochs, the validation loss did not decrease. The model's performance during training was assessed using the validation dataset.

To avoid overfitting, the model's performance during the training phase was assessed using the validation dataset. The difference between the predicted bounding box and the ground truth bounding box was calculated using the smooth-L1 loss function. The difference between the expected class probabilities and the actual class probabilities was calculated using the cross-entropy loss function. The total loss, which was utilized to update the model parameters during training, was created by combining these two losses. The gradients of the total loss with respect to the model parameters were calculated using the backward() method, and the optimizer was then used to update the model parameters.

The intersection over union (IOU) between the predicted bounding boxes and the ground truth bounding boxes for each object instance was assessed to determine the performance of the trained model. The IOU is calculated as the ratio of the areas of intersection and union of the projected and actual bounding boxes. In addition, cross-entropy was used to evaluate the dissimilarity between the predicted probability distribution and the actual probability distribution of the target classes. These evaluation criteria helped to assess the precision of the model and its ability to accurately detect driver drowsiness.

$$\mathcal{L}_{total} = \mathcal{L}_{class} + \lambda \mathcal{L}_{reg} \tag{1}$$

Where:

$$\mathcal{L}_{class} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{C} y_{i,j} \log(p_{i,j}) \tag{2}$$

$$\mathcal{L}_{reg} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{4} \text{smooth}_{L1}(t_{i,j} - \hat{t}_{i,j}) \tag{3}$$

The intersection over union (IoU) metric was used to assess the effectiveness of the segmentation model by computing the amount of overlap between the predicted and target segmentation masks. It measures the similarity between the expected and actual masks.

$$IOU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{4}$$

5. Models

This paper includes a total of four models, three image-net, and yolov5s. The three image-net models were finetuned by replacing the last layers with the classification and regression to obtain the label and bounding box.

5.1. Yolov5s

The YOLOv5s model is a well-liked object detection technique that combines object localisation and classification using a single convolutional neural network (CNN). It is intended to be quick and effective, making it suitable for real-time applications like the detection of driver intoxication. The YOLOv5s model is utilized in this context to locate the driver's face and eyes inside an image, which is a crucial step in figuring out how sleepy the driver is. A dataset of photos with varied degrees of tiredness was utilized to train the model. The locations of the driver's face and eyes were indicated on the photos by bounding boxes. During training, a loss function that included both the cross-entropy loss and the smooth L1 loss was minimized. The YOLOv5s.pt file's pre-trained weights were used to establish the model's weights.

The batch size and picture size for the training were set to 64 and 400, respectively, on a single GPU. The implementation settings were chosen after testing to achieve a compromise between accuracy and training duration. The training procedure was run for 10 epochs. The YOLOv5s model, which achieved great accuracy in localizing the driver's face and eyes inside a picture, has generally shown to be a valuable tool for detecting driver drowsiness.

The detect.py script is used to assess the trained model's performance on fresh photos or videos during testing using YOLOv5s. The model is applied to the test data by the

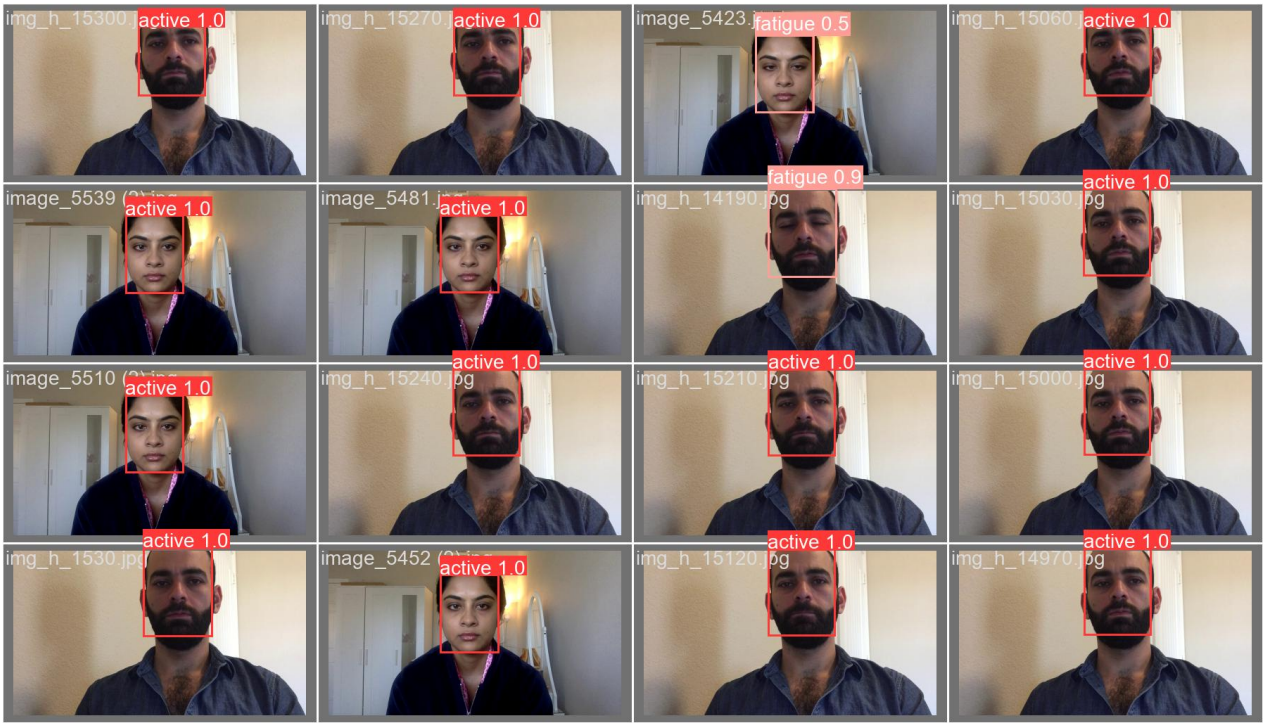

Figure 2. Validation Batch1 Predicted images with labels

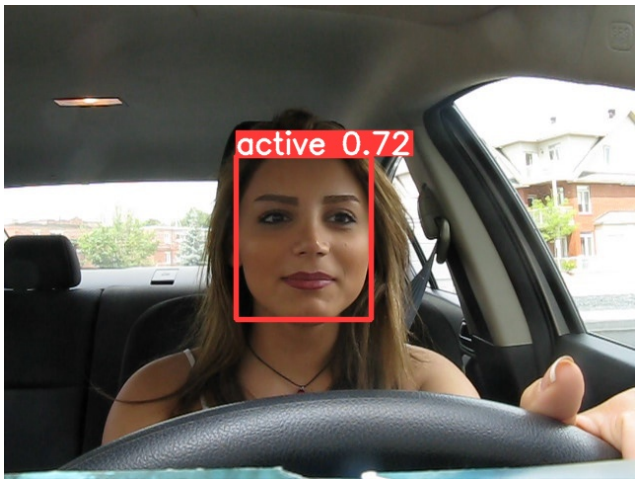

Figure 3. Predicted Test Images

script using the trained model weights as input. In order to eliminate repeated detections and filter out detections with a low confidence level, the script uses non-maximum suppression (NMS). NMS is a post-processing procedure used to cut down on the quantity of bounding boxes the model generates. It operates by deleting any bounding boxes that have a significant amount of overlap with the bounding boxes with the highest confidence scores. Only the most certain and non-overlapping detections are retained as a result.

The detect.py script offers a number of parameters for managing the testing procedure, including the ability to define the input image size, the confidence threshold, the IoU (Intersection over Union) threshold for NMS, and the output format for the detection results. The bounding box coordinates, confidence scores, and, if applicable, class labels for each identified object can all be included in the output format. The script may also be used to compute measures like precision, recall, and F1 score, which are frequently used in object detection tasks, to assess the performance of the model. These measurements offer a factual assessment of the model's performance on the test data and can be applied

to the comparison of several models or parameter settings.

5.2. Resnet

The ResNet50 model was adjusted for this research in order to get classification and regression results for driver sleepiness detection. The ImageNet dataset, which has millions of tagged images, served as the pre-training ground for the ResNet50 model used in this design. The architecture may take advantage of the expertise amassed by the ResNet50 model on a huge dataset and apply it to a new task, such as object identification and classification, by using a pre-trained model.

The final fully connected layer used for classification on the ImageNet dataset was removed in order to adjust the pre-trained ResNet50 model for object identification and classification. Two completely connected layers that were created expressly for the identification and classification of objects were placed in its place. The regression layer produces projected bounding box coordinates for each object in the image, while the classification layer produces predicted class scores for each class in the dataset.

The newly added layers were trained using a dataset of images related to the detection of driver drowsiness in order to fine-tune the ResNet50 model. The pre-trained model's weights were modified during training to accommodate the new goal of detecting driver tiredness. In the context of detecting driver drowsiness, the resulting fine-tuned model was capable of accurately detecting and classifying items.

5.3. VGG

A convolutional neural network with a widely used architecture is the VGG-16 architecture. It has 3 fully linked layers and 13 convolutional layers. The convolutional layers are organized in a way that gradually expands the receptive field size, enabling the network to catch increasingly intricate aspects of the input image. The VGG-16 architecture has been successfully used for a number of computer vision tasks, including object and picture detection.

Transfer learning is implemented in VGG-16 for the purpose of detecting driver drowsiness in a manner similar to the ResNet50 architecture. To do this, two more layers are added and the network's final completely connected layer is eliminated. A classifier layer and a regressor layer make up these new layers. The classifier layer is in charge of forecasting the driver's level of intoxication, while the regressor layer is in charge of forecasting the bounding box locations of the eyes.

The regressor layer has four output nodes that represent the four parameters of the bounding box (the x and y coordinates of the top-left corner, width, and height), whereas the classifier layer has two output nodes that represent the two classes (drowsy and alert). With this method, the VGG-16 architecture is able to utilize the information gained from a big dataset and apply it to a new task, such driver sleepiness detection.

5.4. Googlenet

There are 22 layers in the GoogLeNet convolutional neural network design, including 9 inception modules. The inception modules' ability to conduct various convolutional sizes in the same layer allows for the efficient use of parameters. In order to limit the number of parameters and computations needed while still collecting complicated aspects of the input image, the network can combine various sizes of filters and poolings.

The final layer of the network was swapped out for a classifier and a regressor in order to modify Google Net for the goal of drowsiness detection. The regressor predicts the bounding box coordinates of the driver's face, while the classifier predicts whether the driver is awake or asleep. A dataset of images that were classified as alert or sleepy coupled with the matching bounding box coordinates served as the basis for training the refined model. The pre-trained GoogLeNet model's weights were employed as a starting point for training, enabling the network to take advantage of the information discovered from a sizable dataset like ImageNet. By training the model on the particular goal of sleepiness detection, and optimizing for both classification accuracy and bounding box regression accuracy, the model was refined.

6. Results

The results obtained from the different models show that all models perform well in the task of driver drowsiness detection. The YOLOv5s model achieved an accuracy of 93.01% and an IOU of 95.51%, which is quite impressive considering it was trained for only 10 epochs. The ResNet50 model achieved a higher accuracy of 95.65% but had a lower IOU of 62%. The VGG16 model performedd the best, with an accuracy of 96.87% and an IOU of 62%. Finally, the GoogLeNet model achieved an accuracy of 92.80% and an IOU of 78%, indicating that it performed well in detecting the driver's face.

Table 4. Testing Results

Model	Epochs	Accuracy	IOU
Yolov5s	10	93.012	62.03
Resnet-50	20	96.87	61.95
VGG 16	20	93.012	95.50
Google Net	10	92.80	78.47

7. Conclusion

Based on the results obtained from the four models tested for driver drowsiness detection, it can be concluded that all

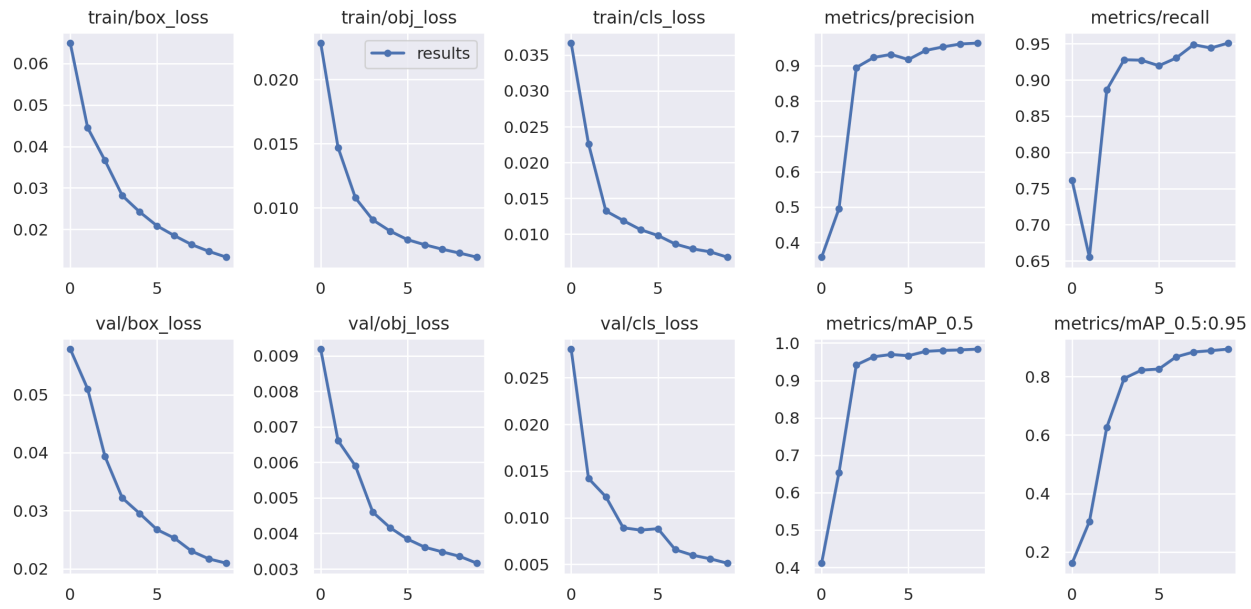

Figure 4. Yolov5s Training results

models achieved high accuracy in detecting driver drowsiness. The highest accuracy was obtained by VGG-16 with 20 epochs, achieving an accuracy of 96.875%, followed by ResNet50 with 20 epochs, achieving an accuracy of 95.647%. YOLOv5s achieved an accuracy of 93.012% with 10 epochs, while GoogleNet achieved an accuracy of 92.801% with 10 epochs.

In terms of Intersection Over Union (IOU), GoogleNet performed the best, achieving an IOU of 0.7847. ResNet50 had an IOU of 62 followed by VGG-16 with an IOU of 61.96%. YOLOv5s had an IOU of 51%.

Overall, all four models performed well in detecting driver drowsiness, but the VGG-16 and ResNet50 models with 20 epochs achieved slightly better results than YOLOv5s and GoogleNet with 10 epochs. The choice of the best model for a specific application will depend on factors such as the size and complexity of the dataset, available computing resources, and the desired balance between accuracy and speed.

Also, the YOLOv5s model has demonstrated its effectiveness in detecting driver drowsiness in real time. However, there is still potential for further improvements to the system, and future work could focus on incorporating more sophisticated techniques and additional indicators of drowsiness to further enhance the accuracy and reliability of the system.

8. Future Work

The accuracy of the detection models can be further improved in future work for this driver drowsiness detection system. Exploring more sophisticated models, like YOLOv8, which incorporates cutting-edge features and methods including attention mechanisms, multi-scale processing, and increased feature extraction, can help improve accuracy.

To further increase the detection system's accuracy, it may be possible to investigate the use of other sensor data, such as heart rate or eye movements. Combining data from many sources can increase the system's overall accuracy and make it easier to detect the onset of sleepiness.

The robustness of the detection models to changes in lighting conditions, head positions, and other elements that may affect the performance of the system in real-world scenarios can also be improved by using data augmentation approaches.

Further testing and validation of the system in real-world scenarios, such as in-vehicle conditions, can help to assess the effectiveness of the system and identify any limitations or areas for improvement. This is in addition to these technological advancements. This may entail working with suppliers and automotive manufacturers to incorporate the technology into cars and assess how well it performs in various driving situations.

Generally speaking, these future directions can serve to increase the precision and efficacy of the driver drowsiness detection system and aid to lower the risk of accidents brought on by driver fatigue.

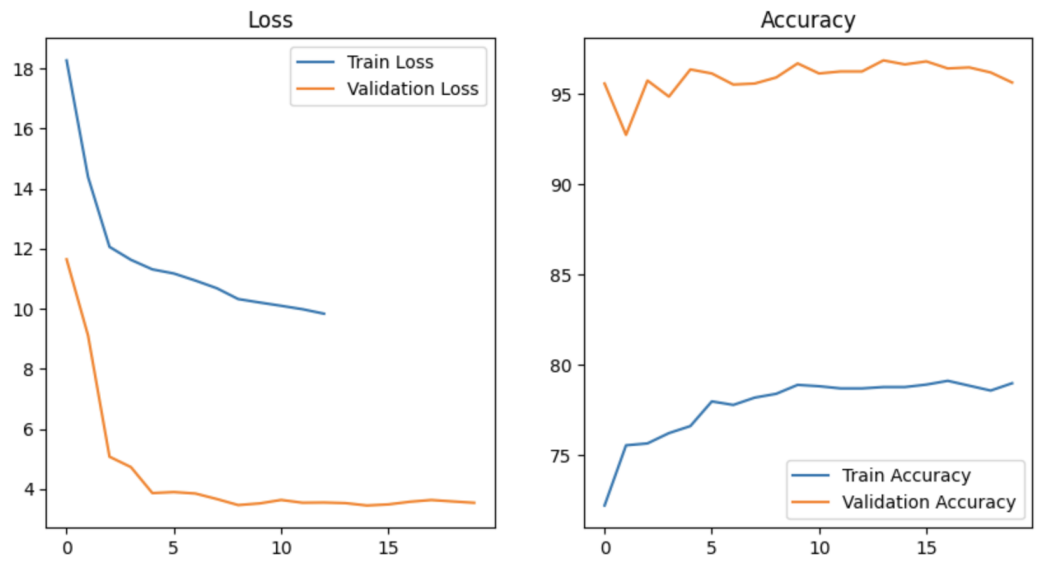

Figure 5. Resnet50 train and valid, accuracy and loss

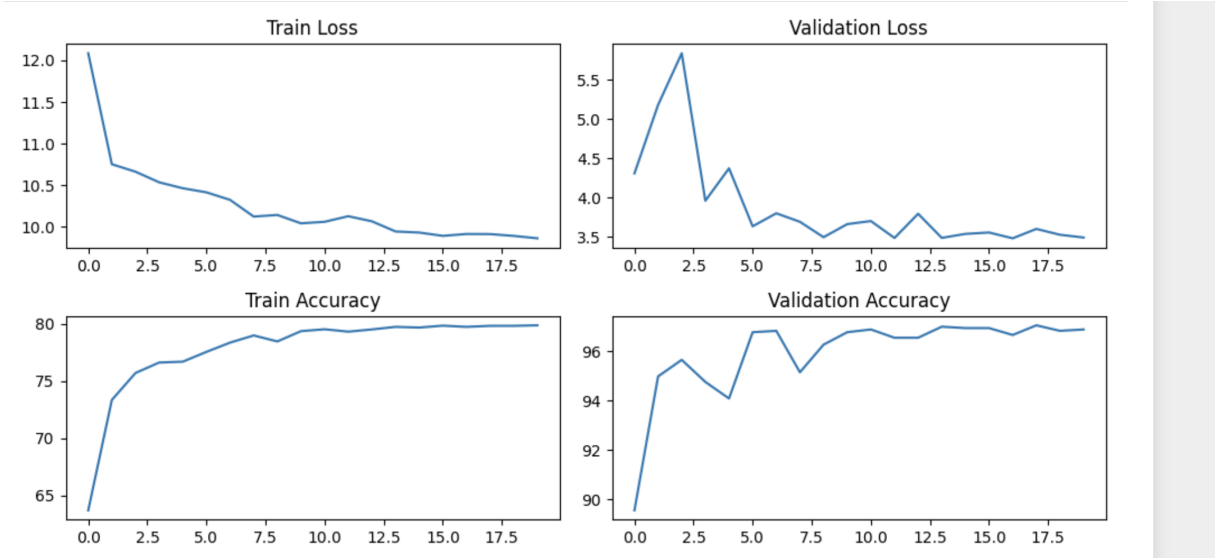

Figure 6. VGG16 train and valid, accuracy and loss

References

[1] S. Arefnezhad, J. Hamet, A. Eichberger, et al. Driver drowsiness estimation using eeg signals with a dynamical encoder–decoder modeling framework. *Scientific Reports*, 12:2650, 2022. 2

[2] Elena Magán, M. Paz Sesmero, Juan Manuel Alonso-Weber, and Araceli Sanchis. Driver drowsiness detection by applying deep learning techniques to sequences of images. *Applied Sciences*, 12(3), 2022. 2

[3] Toan Vu, An Dang, and Jia-Ching WANG. A deep neural network for real-time driver drowsiness detection. *IEICE Transactions on Information and Systems*, E102.D:2637–2641, 12 2019. 2

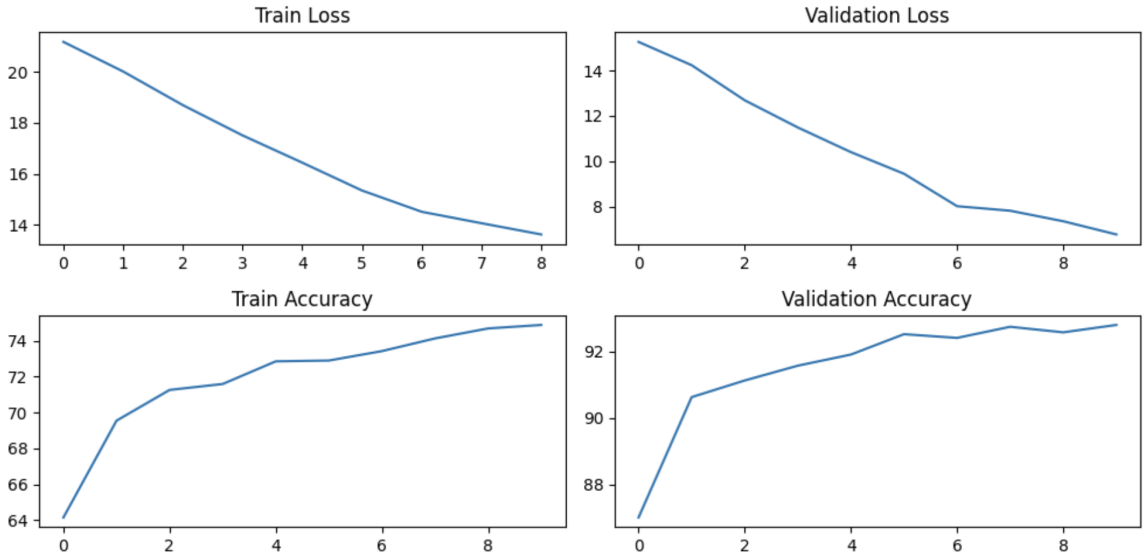

Figure 7. Googlenet train and valid, accuracy and loss