Real-time Drowsiness Detection

Project Report

Image processing with Machine Learning(DA526)

Group Members

Mayukh Das(224101036)
Debarpan Jana(224101016)
Bitan Guha Roy(224101015)
Mahasin Hossen Munshi(224101035)
Madhurima Sen(224101034)

Follow the link for the code
https://github.com/mak109/realtime-drowsiness-detection

Problem Statement:-

- *Real time drowsiness detection*

This project aims to develop a real-time drowsiness detection system for individuals in a video feed. The system will first detect faces in the video and then classify them as drowsy,low vigilant or awake using a pretrained model trained on a dataset of alert,low vigilant and drowsy individuals. The model will output a probability score for each class, and the system will use a threshold to determine whether the individual is drowsy or awake along with localisation of the class(bounding box). This approach can have practical applications in driver safety and workplace safety, but the system's accuracy will depend on the quality of the face detection algorithm, the training data, and the threshold chosen.The drowsiness detection system consists of the following components:

a. Face Detection Algorithm: This algorithm identifies and localises faces within the video feed. Accurate face detection is crucial for the system's overall performance.

b. Pretrained Model: Trained on a dataset of alert, low vigilant, and drowsy individuals, this model classifies faces into one of the three aforementioned categories and provides probability scores for each class.

c. Threshold Determination: A threshold value is defined to categorise the individuals as either drowsy or awake based on the probability scores generated by the model.

d. Bounding Box Localization: The system identifies and localises the specific region within the video where the drowsiness is detected, providing a bounding box around the individual's face.

Dataset to be used:

The UTA-RLDD (Real Life Drowsiness Dataset) has been specifically developed to address the task of multi-stage drowsiness detection. Unlike other datasets, RLDD focuses not only on extreme and easily visible cases of drowsiness but also includes subtle instances of drowsiness. This dataset aims to provide a comprehensive representation of drowsiness across different scenarios.

Dataset Description:-

The RLDD dataset comprises approximately 30 hours of RGB videos recorded from 60 healthy participants. Each participant contributed three videos, one for each of the three classes: Alertness, Low Vigilance, and Drowsiness, resulting in a total of 180 videos.

The participants involved in the dataset were over 18 years old, consisting of 51 men and 9 women from diverse ethnic backgrounds. The ethnicities represented in the dataset include 10 Caucasian, 5 non-white Hispanic, 30 Indo-Aryan and Dravidian, 8 Middle Eastern, and 7 East Asian individuals. The age range of the participants was from 20 to 59 years, with a mean age of 25 and a standard deviation of 6. In 21 out of the 180 videos, the participants wore glasses, and in 72 videos, participants had considerable facial hair. The videos were recorded using the participants' own cell phones or web cameras, capturing them from various angles in real-life environments with diverse backgrounds.

Class Definitions:-

The three classes in the RLDD dataset were explained to the participants as follows:

Awake: Participants were instructed that being in an alert state meant they were experiencing no signs of sleepiness.

Low Vigilant: This class represents subtle cases where some signs of sleepiness may appear, or sleepiness is present but no conscious effort to remain alert is required.

Drowsy: Participants in this state were instructed that they need to actively try to prevent themselves from falling asleep.

Dataset Usage:-

For training and validation purposes, the UTA-RLDD dataset is utilised. It provides a rich and diverse set of video data, enabling the development and evaluation of robust drowsiness detection algorithms. The dataset's extensive collection of videos ensures that models trained on it are capable of detecting both extreme and subtle cases of drowsiness.

To assess the performance and generalizability of the trained models, a custom dataset will be used for testing. This dataset includes videos recorded by the group members who participated in the project. By using this custom dataset, the system can be evaluated on real-life scenarios involving the project team, allowing for a more realistic assessment of the drowsiness detection system's performance.

Related Works:-

Drowsiness detection has garnered considerable attention in various fields, such as transportation safety, workplace safety, and healthcare. Numerous methods and techniques have been proposed to effectively detect drowsiness in individuals. Here, we elaborate on some of the prominent approaches:

1.EEG-based Drowsiness Detection:

EEG measures the electrical activity of the brain and has been widely used to detect drowsiness. Studies have analysed EEG signals to identify specific patterns associated with drowsiness, such as increased theta and alpha waves and decreased beta waves. By monitoring these patterns, algorithms can determine an individual's drowsiness level.

2.Eye-tracking-based Drowsiness Detection:

Eye-tracking techniques track eye movements, including blink rate, eye closure duration, and eye gaze direction, to assess drowsiness. The analysis of eye-related parameters can indicate the level of alertness. For example, increased blink rate or prolonged eye closure duration may suggest drowsiness.

3.Heart Rate Variability (HRV)-based Drowsiness Detection:

HRV measures the variation in the time interval between successive heartbeats. Reduced HRV has been associated with drowsiness and decreased autonomic nervous system activity. By analysing HRV signals, algorithms can infer an individual's drowsiness state.

Facial Expression-based Drowsiness Detection:

Facial expressions can provide insights into an individual's alertness level. Computer vision techniques are employed to analyse facial features and expressions, such as eye closure, yawning, and changes in muscle tension. These visual cues are then used to determine the presence of drowsiness.

Machine Learning-based Drowsiness Detection:

Machine learning, particularly deep learning methods, have been increasingly utilised for drowsiness detection. These techniques involve training models on large datasets that include instances of drowsy and alert individuals. The models learn to extract relevant features and classify new instances into drowsiness or alertness categories.

Methodology:-

- **Data Set preparation:**
  - Portion of the UTA- RLDD dataset is used. From video images are captured automatically and labelled manually suitable for YOLOv5 format using a open source tool labelImg.
  - Whole UTA-RLDD dataset(images) is used in baseline Model as well as fine tuned model
- **Training and Validation:**
  - Training and validation is done on the preprocessed dataset using various hyperparameter configurations.Wandb is for this purpose . Wandb is a good collaborative model tracking platform.
- **Testing:**
  - The best model(Having highest prediction score m_AP/0.5) which is found from hyperparameter tuning is used for testing purposes for YOLOv5.
  - Baseline Model and Fine Tuned model uses val_accuracy metric

BaseLine Model:-

- **Data Set :**
  - Approx 10k images dataset is almost balanced.following is the dataset structure Labels(y) and image input is inferred from the directory structure flow_from_directory() is used
  - *UTA-RLDD*
    - Fold1
      - awake
      - drowsy
      - low vigilant
    - Fold2
    - Fold3

        - ……..
        - Fold5

- **Model structure :**
    - CNN is used .The hyperparameters (number of conv layers,number of dense layers,number of filters in each conv layers ,filter sizes in each layer etc) are all flexible and tunable
- **Challenges Faced and Remedy:**
    - Huge amount of overfitting (in first few epochs train accuracy hits >90 % but val accuracy is around 40%)
    - We tried a lot of different techniques and hyper parameter setting like we used batch normalization layer, increased the dropout ,decreasing the number of convolution layers ,using 2 dense layers etc.
    - We also used two version of datasets structure One that is show in previous page and another is not in folders just dump sub folder containing awake,drowsy,low vigilant images under a train folder and in code we used train val split of 0.1.
    - The fold wise manner of dataset structure gives better result as the validation dataset is balanced

YOLO Model:-

Why YOLO???

- YOLO is extremely fast and reasons globally about the image when making predictions, so it implicitly encodes contextual information about classes as well as their appearance.
- YOLO In our application we wanted to detect multiple objects classes(drowsy,awake,low vigilant) in real time and bounding box method of localising the class and predicting their class prediction scores.YOLO is a good choice
- Yolo divides the image in SxS grid cells
    - For each grid cell,it predicts B(Generally valu = 2) boundary boxes and each box has

one box confidence score,
  - it detects one object only regardless of the number of boxes B,
  - it predicts C(C=3) conditional class probabilities (one per class for the likeliness of the object class).
- **Data Set :**
  - Around 330 labelled images(in yolo format) .Dataset is balanced.
- **Model structure :**
  - Ultralytics provides very efficient implementation of YOLO So we have cloned their repo and tuned several of their model checkpoints yolov5s, yolov5m, yolov5x ,epochs ,patience,image size,batch size etc.
- **Loss (sum-squared error between the predictions and the ground truth to calculate loss)**

  The loss function comprises of:-
    - the classification loss.
    - the localization loss (errors between the predicted boundary box and the ground truth).
    - the confidence loss (the objectness of the box).
- **Metric used:**
  - m_AP/0.5 which is found from hyperparameter tuning is used for testing purposes for YOLOv5.
- **Challenges Faced & Remedy:**
  - Large number of True Negatives (Ground truth detected in background) and model detects several bounding boxes for the same object (Unstable).
  - There was overfitting so we used yolov5s (small model) and cleaned the data a little bit.

Experiments and Results :-

To view the detailed results of all our experiments please follow the given links

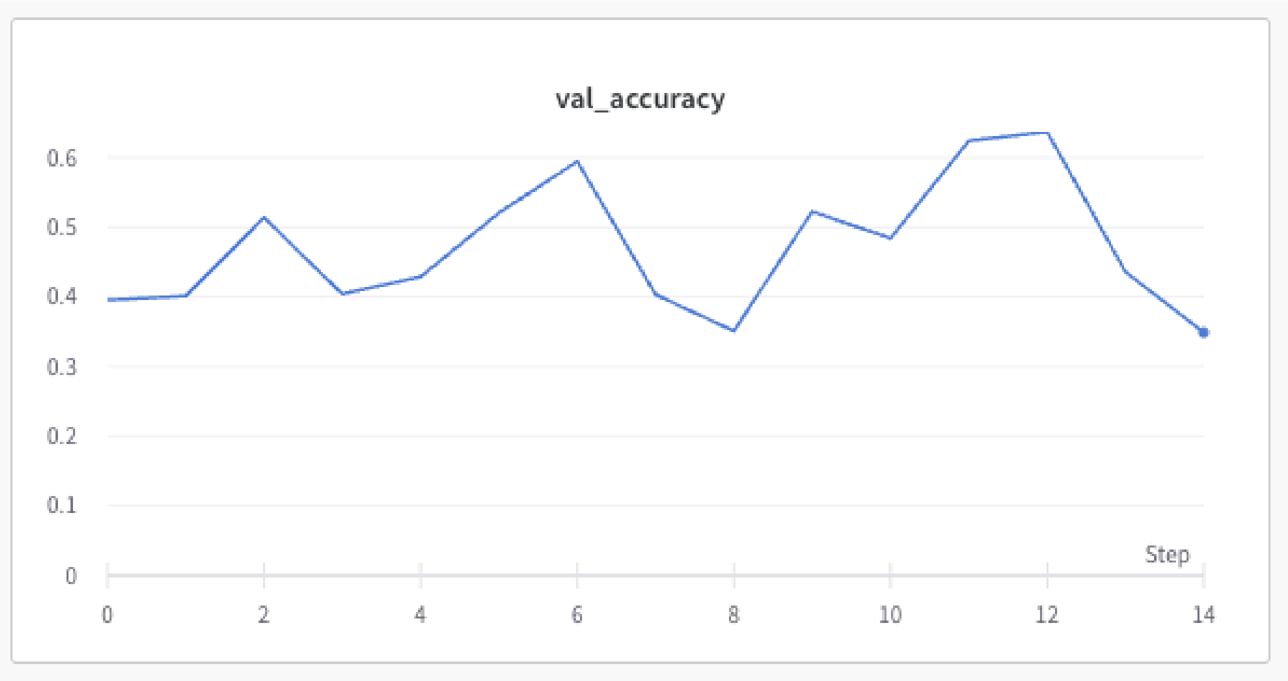
- https://wandb.ai/ipda526/baseline-drowsiness-detection baseline experiments
- https://wandb.ai/ipda526/baseline-drowsininess-detection baseline experiments
- https://wandb.ai/ipda526/finetune-drowsiness-detection Finetune experiments
- https://wandb.ai/ipda526/yolov5-drowsiness-detection YOLOv5 experiments

Baseline Model best results

https://wandb.ai/ipda526/baseline-drowsiness-detection/runs/xvsgsoxm Follow to view all the results for this experiment

View the model here

Validation accuracy around 63%



Test results

Other satisfactory results

https://wandb.ai/ipda526/baseline-drowsiness-detection/runs/xc3l8g28 baseline model around 55%

Fine-tuned doesn’t give that much good results surprisingly

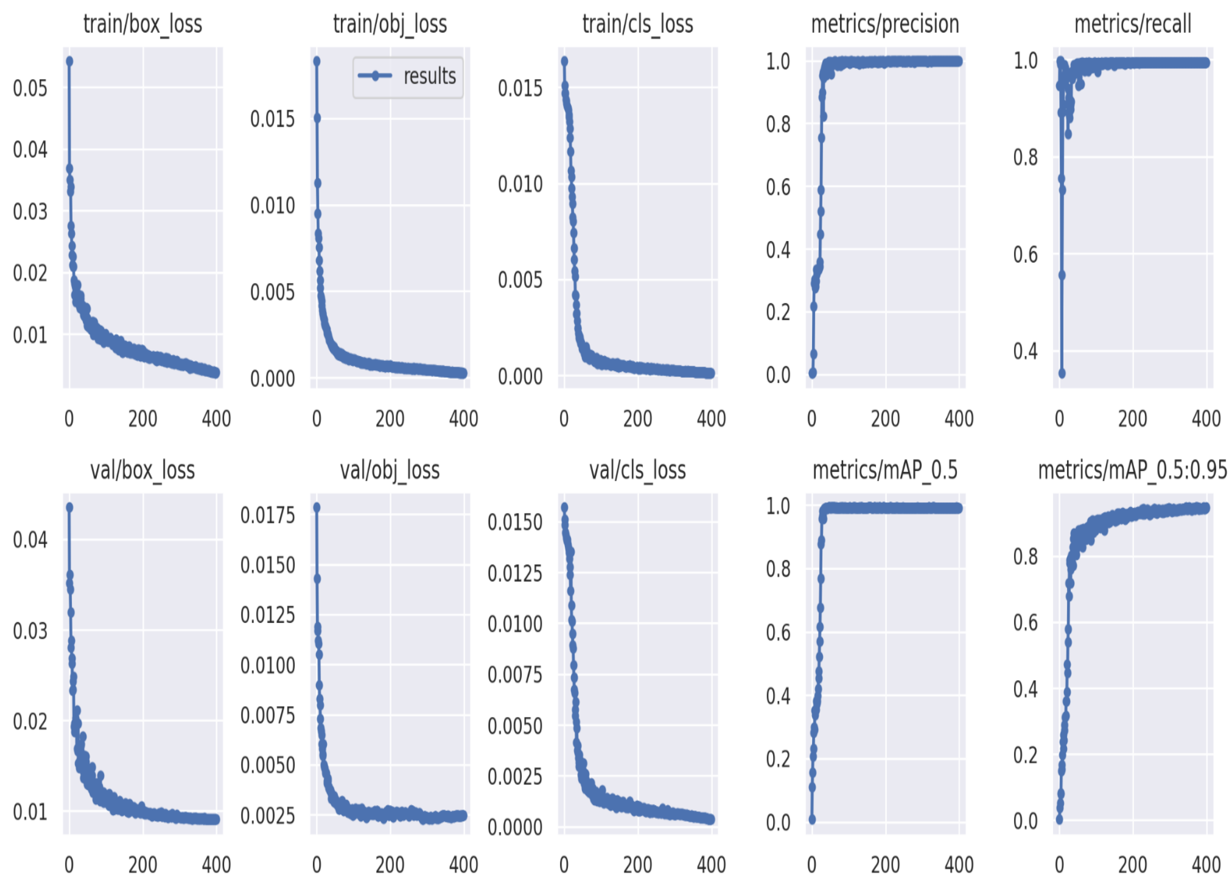
https://wandb.ai/ipda526/finetune-drowsiness-detection/rlyu87v1

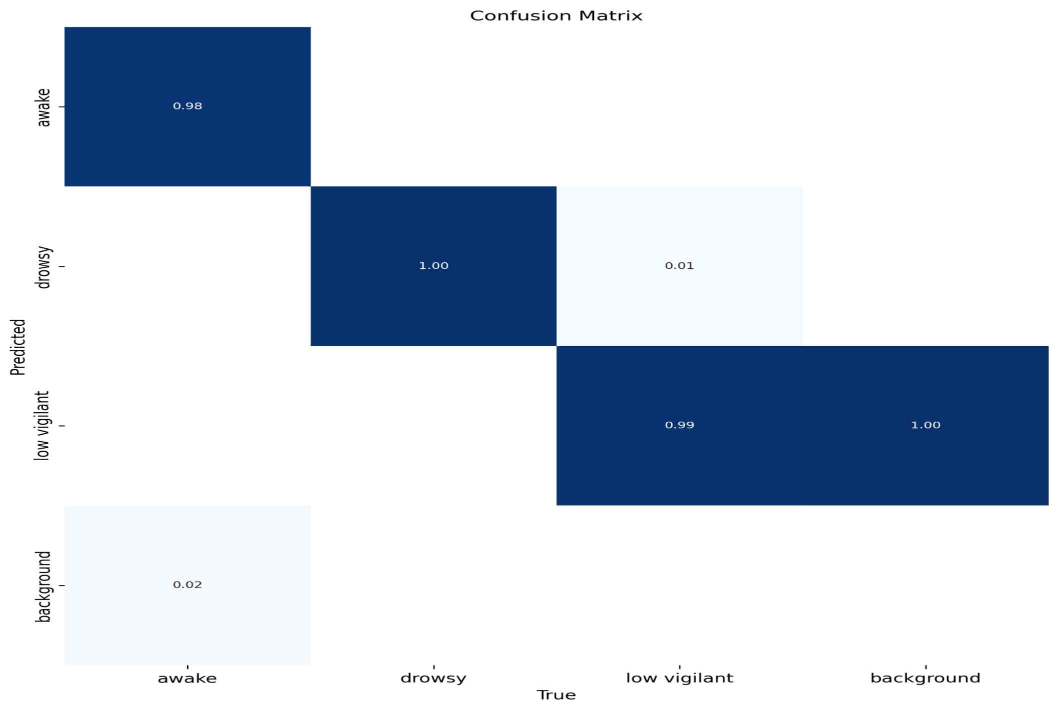
Yolo Model :

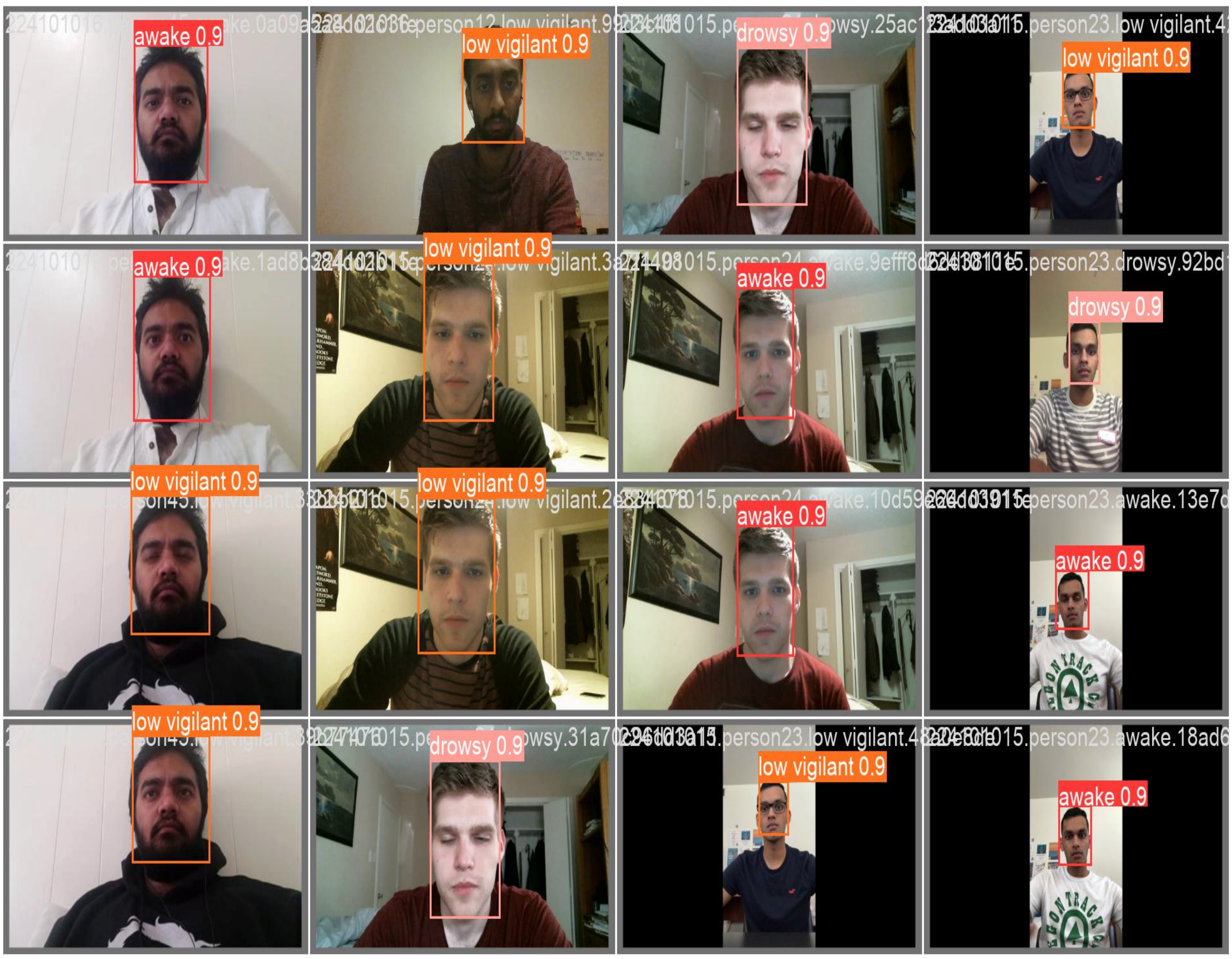
To view the full results follow the link

https://wandb.ai/ipda526/yolov5-drowsiness-detection/runs/ol2p2qop

Metrics, Confusion Matrix and Validation batch results for the above model




Confusion Matrix
awake
drowsy
low vigilant
background
Predicted
0.98
1.00
0.01
0.99
1.00
0.02
awake
drowsy
low vigilant
background
True


awake 0.9
low vigilant 0.9
drowsy 0.9
low vigilant 0.9
awake 0.9
low vigilant 0.9
awake 0.9
drowsy 0.9
low vigilant 0.9
low vigilant 0.9
awake 0.9
awake 0.9
low vigilant 0.9
drowsy 0.9
low vigilant 0.9
awake 0.9

Conclusions:-

- ***Our Novelty:-***
  - YOLOv5 is very popular for real time object detection and trained on a very large dataset.We will be using fine-tuning on this large model based on our dataset to customize it for drowsiness detection and hope to get a good performance in real time.
  - We have developed our own baseline model from scratch.
- ***Future Works:-***
  - In this project we have not achieved that much performance which we expected to achieve initially.We believe that it is because we have not cropped the faces of our training data ,So our model [In both baseline and YOLOv5] is unnecessarily learning those features which are not relevant for our objective.In future we will try to use only cropped faces as training data to achieve better performance.
  - We have also observed that In Real life ,The state Drowsy is not a sudden event.It rather follows some sequence of steps.In future We will try to use RNN and LSTM in our model to learn those dependencies to get better performance for a model which will work in real time.

References:-

1. Deep Drowsiness Detection using YOLO, Pytorch and Python
2. UTA-RLDD Dataset
3. Ultralytics, providing very efficient yolo implementation
4. Wandb, for hyperparameter tuning
5. Jonathan hui's blog on yolo
6. About Metrics used in yolo