Using YOLOv5 machine learning model to Detect Distracted Driving

Jace Moore[1], Minxiao Wang[2], Ning Yang[3]

[1]SI Bridges to the Baccalaureate, Information Technology, Southern Illinois University Carbondale.

[2]Electrical and Computer Engineering, Southern Illinois University Carbondale.

[3]Information Technology, Southern Illinois University Carbondale.

SIU Southern Illinois University Carbondale — SI Bridges to the Baccalaureate

NIH

Introduction

- According to the National Highway Traffic Safety Administration, roughly 5% of fatal car accidents are caused by distracted driving.
- There has been a 2% decrease in numbers of fatal accidents caused by distracted drivers since 2010.
- This decrease may have been caused by the increased countermeasures to distracted driving, such as traffic violations.
- If the countermeasures have been shown to work, then more may decrease the number even more.
- Machine learning can be used in many applications, it has seen a massive increase in the automotive industry.
- Machine learning can be used to detect distracted drivers, which can then be used either in real time, or after an accident has occurred to determine fault.
- Machine learning is a form of AI (artificial intelligence) that learns and improves from experience without being explicitly programmed to do so.
- The form of machine learning that would be used to detect distracted drivers is called Object Detection.
- Machine learning model (YOLOv5) is shown thousands of images and is told what to look for and then it finds patterns and then after that, predictions are made.
- YOLOv5 (You Only Look Once version 5) is an object detection algorithm that is famous for its speed and accuracy.
- YOLOv5 is be faster and more accurate because of how it divides images into a grid system. Each cell in the grid is responsible for detecting objects within itself, cutting down the training time severely.
- When a machine learning model learns it is called training.
- Since YOLOv5 is a premade model training with it is as simple as downloading the YOLOv5 repository from their website, then download your own dataset, then just run a single line of code to start the training.

Methodology

- A dataset called TICaM consisting of images of people in a simulated environment to simulate being in a vehicle was annotated to detect distractions.
- An additional dataset, that was created to improve accuracy from multiple angles, was also annotated.
- Annotation was done using a website called RoboFlow that allows users to annotate images for object detection.
- Annotation was done by looking at an image and determining what the driver is doing at that point by a person and then they draw a box over the driver that is labeled with a class.
- There were 8 total classes of distraction and 1 non-distracted class.
  - Normal Driving
  - Distracted Left
  - Distracted Right
  - Distracted Behind
  - Distracted Phone
  - Distracted Radio
  - Distracted Glovebox
  - Distracted Leaning Forward
- After images were annotated, RoboFlow compiled all the images and split them into three groups.
- Three groups consist of Train, Validation, Test.
  - Train – What YOLOv5 used to calculate predictions
  - Validation – Used to validated the predictions made from the train group
  - Test – Where the predictions are made to be validated
- These three groups, along with other files needed to begin training, are then given by RoboFlow.
- Google Colab was used as a remote host for training.
- Training took roughly 4 hours.
- The custom model was then downloaded from Google Colab.
- The custom model was loaded onto a laptop that was then used to detect distracted drivers in real time through a USB webcam that was mounted on the dashboard of a stationary vehicle.

Results

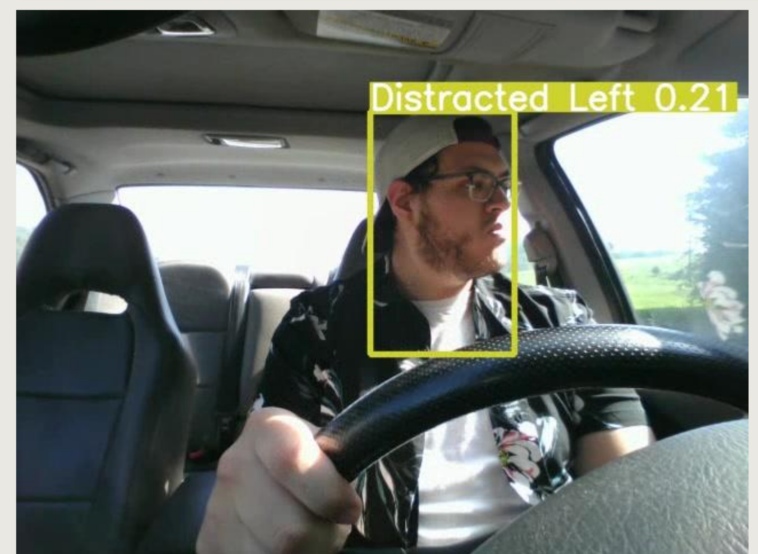

Image 1 shows the Distracted Driver model successfully detecting a driver, distracted by looking left with a 0.21 confidence level.

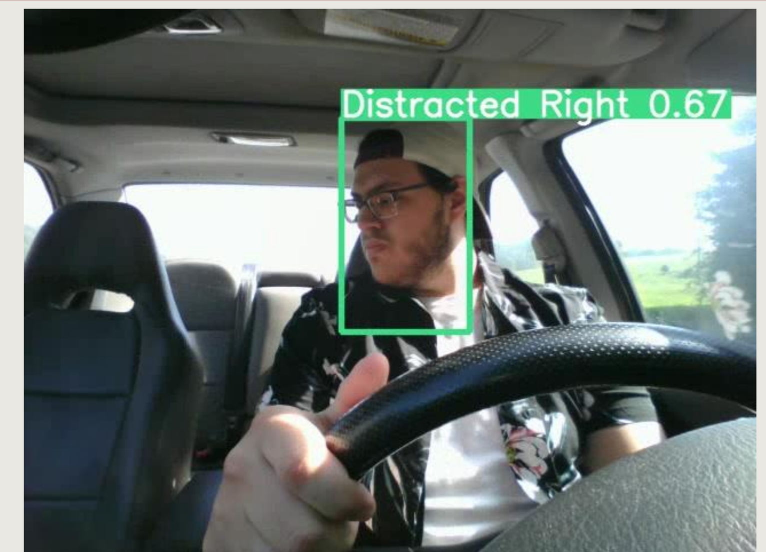

Image 2 shows the Distracted Driver model successfully detecting a driver, distracted by looking right with a 0.67 confidence level.

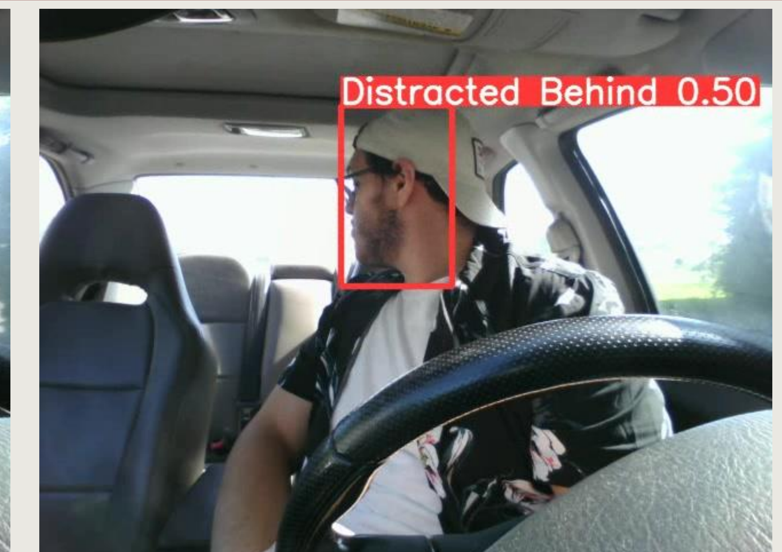

Image 3 shows the Distracted Driver model successfully detecting a driver, distracted by looking behind with a 0.50 confidence level.

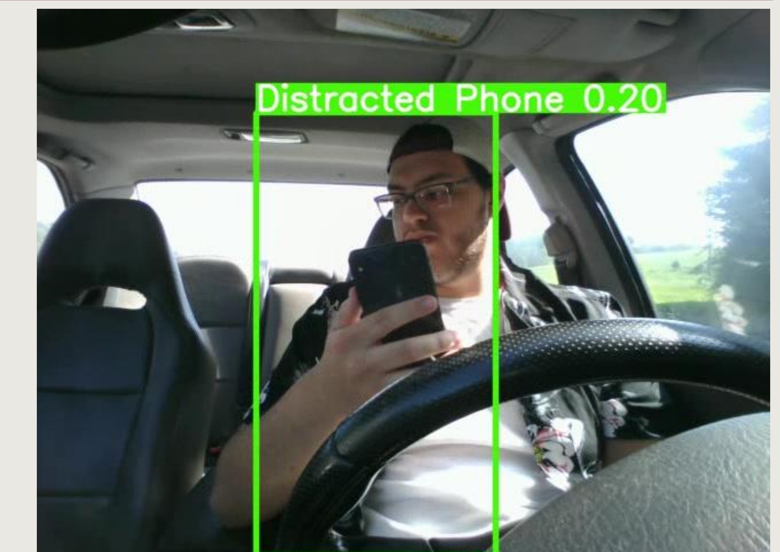

Image 4 shows the Distracted Driver model successfully detecting a driver, distracted by looking at their phone with a 0.20 confidence level.

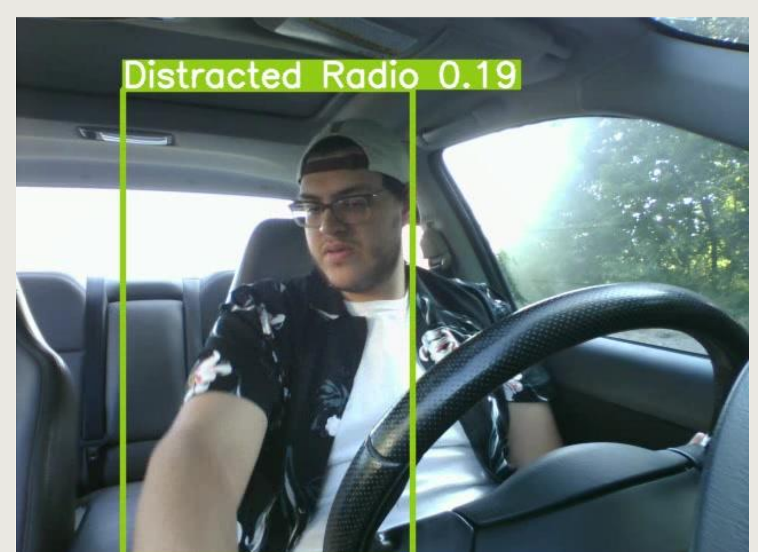

Image 5 shows the Distracted Driver model successfully detecting a driver, distracted by looking at their radio with a 0.19 confidence level.

Image 6 shows the Distracted Driver model successfully detecting a driver, distracted by looking in their glovebox with a 0.29 confidence level.

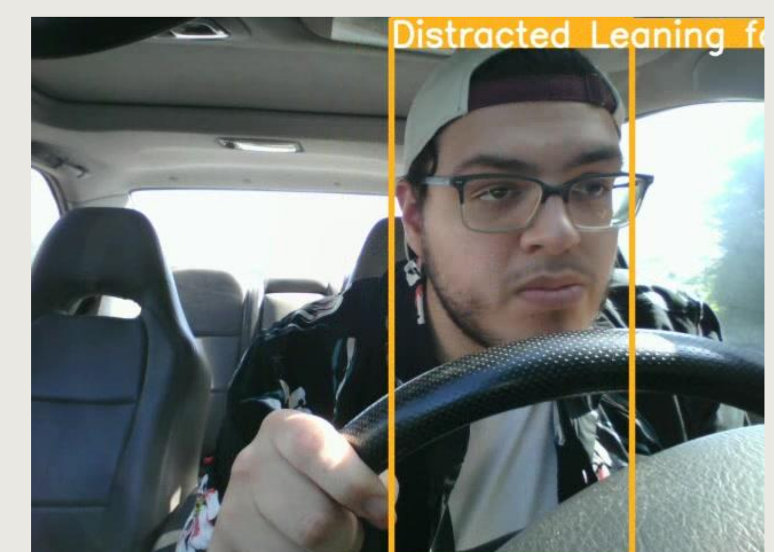

Image 7 shows the Distracted Driver model successfully detecting a driver, distracted by leaning forward with a 0.32 confidence level.

Image 8 shows the Distracted Driver model successfully detecting a driver, distracted by taking their hands off their wheel with a 0.14 confidence level.

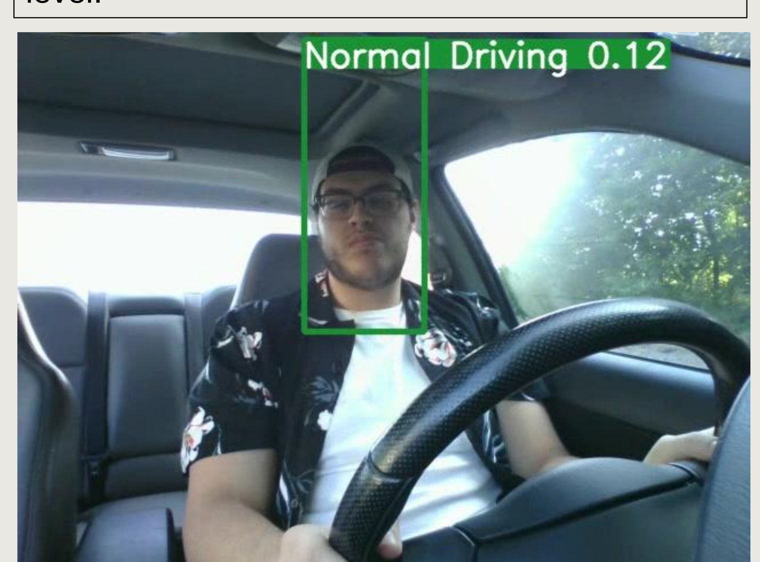

Image 9 shows the Distracted Driver model successfully detecting a driver, not distracted with a 0.12 confidence level.

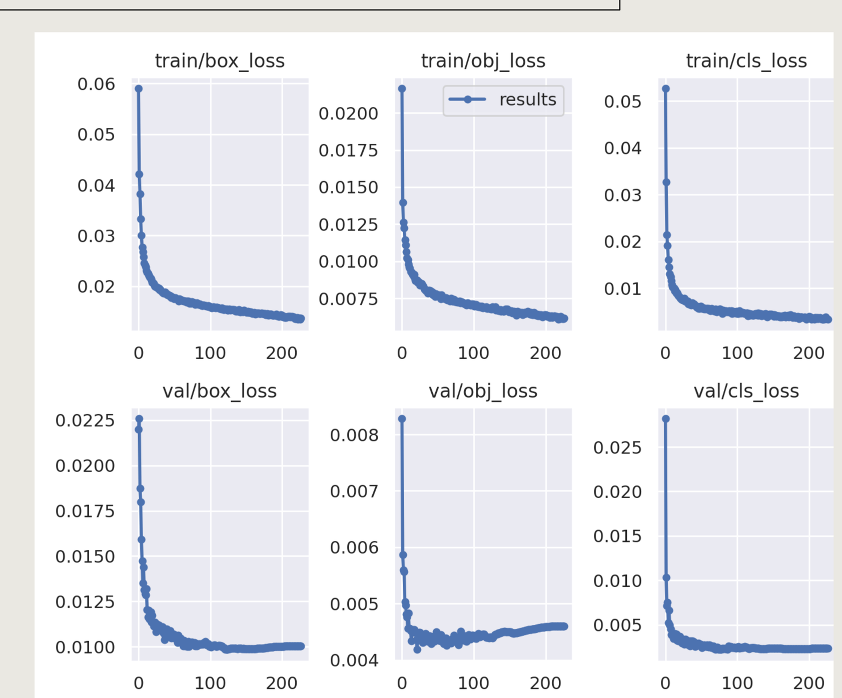

Figure 1 shows a total of 6 graphs that each correspond to the distracted driving YOLOv5 model's performance while training. (From left to right) The box_train graphs show the bounding box regression loss. The obj_loss graphs show the confidence of object presence. The cls_loss graphs show the classification loss.

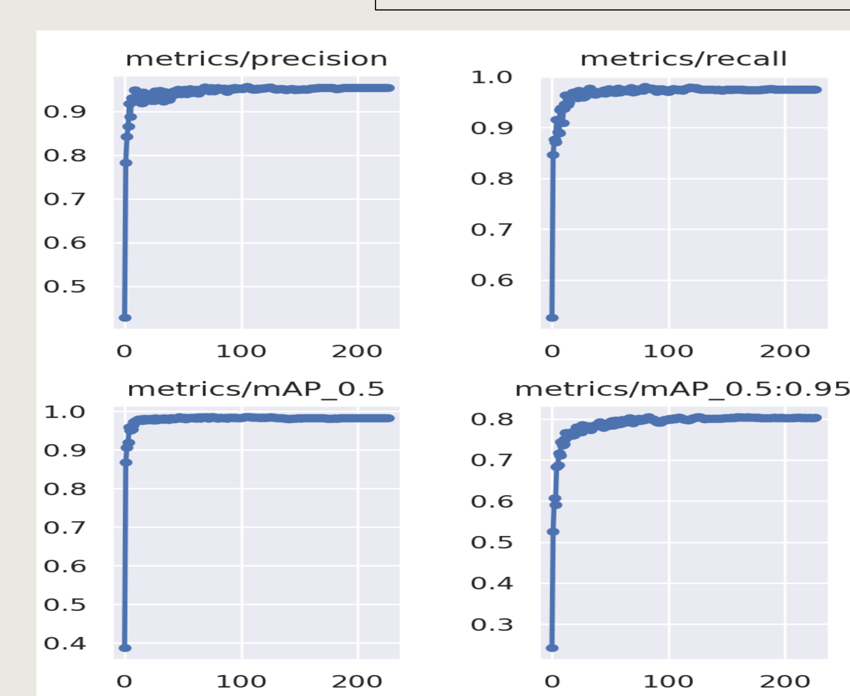

Figure 2 shows a total of 4 graphs that each correspond to the distracted driving YOLOv5 model's performance while training. (From left to right) The precisions graph shows how accurate the predictions were. The recall graph shows how many of the true positives were found. The mAP graphs show the mean average precision of the overall model, at 0.5 and 0.95 respectfully

Conclusion

- The results show that this custom YOLOv5 model had
  - A mAP at 0.5 of 98%
  - A mAP at 0.5:0.95 of 80%
  - A recall of 97%
  - A precision of 95%
- From these results, it can be concluded that this distracted driving model was successful as detecting when a driver is distracted in real time.

Limitations

- One of the biggest limitations to the model is the size and diversity of the dataset used to train the custom model.
- Many of the images in the dataset are from the same angle and background and do not give a wide variation to how the camera can be setup while trying to perform real time detections.

Acknowledgements

Thank you to Laxmi Sagwan-Barkdoll, and the Southern Illinois Bridges to the Baccalaureate program at Southern Illinois University, Carbondale, as well as Dr. Ning Yang and Minxiao Wang for their help with this project. I would also like to thank the TICaM: {A} Time-of-flight In-car Cabin Monitoring Dataset team for letting me use their dataset in my research.