SOLUTION DESIGN DOCUMENT

PROJECT STATEMENT

The number of different road signs has increased tremendously due to the rising number of vehicles. Thus, an automated system to recognise and classify these traffic signs may prove essential to help to prevent road accidents.

SOLUTION MOTIVATION

Tens of thousands of people die each year[1] as a result of motor vehicle accidents. Road deaths can be caused by a variety of reasons, however driver neglect of road signs, obstruction of the traffic signs, and drivers being distracted account for.

Having a system that can detect and classify road signs could prove to be an important driver aid. The programme could have the potential to be used at the level of notifying the driver of, for example, a 30mph speed limit sign they may have missed, as speeding alone caused 26% of all road traffic accidents in 2019.[1]

The system could have the potential to significantly improve the accuracy of fully autonomous vehicle speed recognition, especially where limited-time road works change speed limits. Many car companies now are experimenting with full autonomy. This system could play a key part by becoming a critical aid to safety. The system could thrive, as it should be able to accurately detect and classify roads according to their signage. If the system could work in real-time, road works identification and specifically, short-term works could be reported and the data shared with other road users.

A Traffic Sign Detection and Recognition (TSDR) system plays a significant role in the driver assistant model by regulating traffic, demonstrating the state of the road, and making the drivers and pedestrians aware of new road signs which may have been ignored because of inattentiveness, thereby ensuring an extra layer of safety from accidents.

SOLUTION REQUIREMENTS

Methodology
The methodology must be robust. Sign detection: onboard sensors and estimation tools should provide a 360-degree view of the driver environment all the time.

Speed
The solution must be capable of working in real-time.

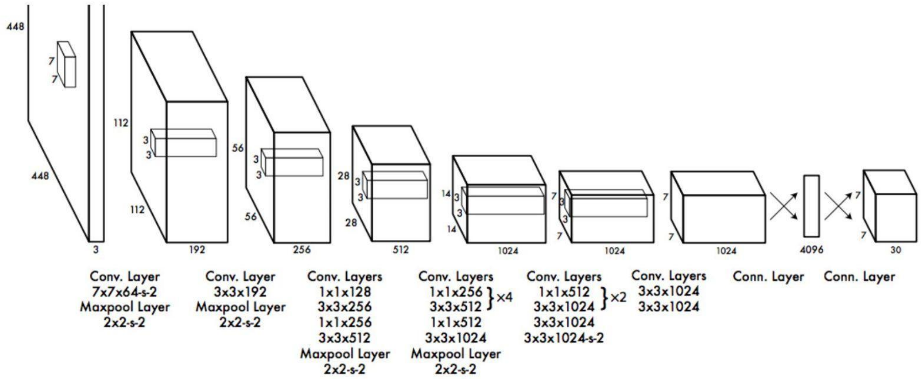
Accuracy
The solution should be accurate to within an agreed tolerance.

Cost
The solution should be cost effective and not cost-prohibitive in order to assist global roll-out.

Data Sharing
The solution must allow for the road signage data to be made available to other applications within the vehicle or that can be used in conjunction with other safety features.

Legal
The solution must not infringe the data protection laws of any country or effect anyone's human rights.

SOLUTION DIAGRAM



This is the Architecture of the YOLO algorithm that is being implemented:
The detection network has 25 convolutional layers followed by 2 fully connected layers. Altering 1x1 convolutional layers reduces the features space from the preceding layers. We pretrain the convolutional layers on the imageNet classification task at half the resolution (224 x224 input image) and then double the resolution for detection.

DATA DESCRIPTION

When deciding on the dataset there should be three main things to look for:
1. Simple enough to be able to Train.
    - a large dataset will take a large amount of time to train.
2. Varieties of signs eg. different countries, traffic lights.
3. The Dataset must include labels and bounding boxes.

SOLUTION DESCRIPTION

How the System Will Work

- First implement the YOLO algorithm using the CPU.
- Train though the command line.
- Build a UI around it
- Try implementing the use of a GPU.

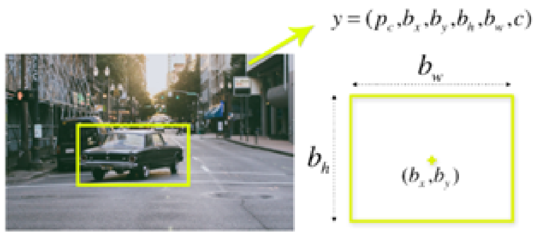
Algorithm Choice

One of the main decisions to make is the Algorithm to go for.
In the research process it was clear that the Most viable algorithm options to base our project on, were either YOLO or R-CNN.

YOLO: The process of You Only Look Once (YOLO)

YOLO is a fairly new approach to object detection, it works using a single Convolutional network which concurrently predicts Multiple bounding boxes and class probabilities for those boxes. Unlike R-CNNs YOLO does not need a complex pipeline, this is because the frame detection is done as a regression problem. The neural network is simply running on a new image at test time to predict detections. [2]

The YOLO algorithm aims to predict a class of an object and its bounding box. The bounding boxes can be described using four descriptors. The center of a bounding box (bxby), width (bw), height (bh) and the value **C** is corresponding to a class of an object. [3]



The YOLO algorithm is not searching for interesting regions that may be object. Instead, the YOLO algorithm splits the image into cells, these cells are typically 19 x 19. Each of the cells is then responsible for predicting 5 boundary boxes. [3]

Next a non-max suppression process is done, this removes all the boxes with low probability and bounding boxes with the highest shared area. [3]

Real-time predictions which are essential for the use of YOLO in driverless cars and implementation to the web will be necessary. However, YOLO does struggle in some areas. The main issue for YOLO is small objects. This is as each grid cell only predicts two boxes and can only have one class. This means If YOLO is given an image of a small object in groups, for example a flock of birds, it will struggle to distinguish each one accurately.

R-CNN: R-CNN (Recurrent Neural Networks)

The R-CNN computes and extracts specific features of objects and afterwards these are used in the classification layer to identify specific regions on the image. This results in a bounding box around one or multiple objects of interests contained within the image.

The R-CNN algorithm specifically helps as rather than having a huge number of regions selected the algorithm uses a selective search to narrow the number down to 2000 regions which is much easier to work with. The selective search algorithm generates region proposals.[4]

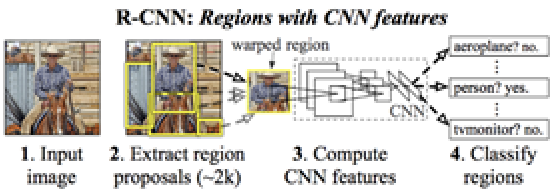


Figure 1 shows an example of how the algorithm would work when detecting objects within an image including the image classification.[4]

Analysis

This paper performs detailed experiments in driverless car usage. Both the Faster R-CNN and YOLOv4 in the experiments were trained and validated on MS COCO dataset.

	One Side of the Road			
	Precision	*Recall*	*F1*	*FPS*
YOLOv4 - 416	**100.00%**	58.13%	71.54%	68
YOLOv4 - 512	98.75%	66.04%	76.56%	63
YOLOv4 - 608	98.04%	70.49%	79.68%	60
Faster R-CNN	95.00%	**87.55%**	**89.43%**	14
	Both Sides of the Road			
YOLOv4 - 416	98.18%	56.32%	70.67%	68
YOLOv4 - 512	98.07%	64.91%	77.15%	63
YOLOv4 - 608	**99.16%**	67.86%	79.36%	60
Faster R-CNN	95.47%	**79.50%**	**85.54%**	14

The results Show that YOLOv4 has a higher precision but in both instances the faster R-CNN has significantly higher recall and F1 score than YOLOv4. The test showed that YOLOv4 can get 98.04% precision and 70.49% recall, but the Faster R-CNN achieved a 95.00% precision and 87.55% recall score. The mean recall values recorded in the test were 67.86% and 79.50%, and the mean precisions are 99.16% and 95.47% jointly for both models.

Regarding using these algorithms for a road sign detection and classification YOLOv4 Seems to be the choice to go for as it runs at 68 fps which is suitable for real time use. The study also tested YOLOv3 and ran at a 78-fps detection speed while the recall was 5% lower than YOLOv4. This would be the obvious solution due to the potential use in autonomous cars in the future which would mean the program would have to run in real time.

Proposed Solution

In this comparative analysis between algorithms, YOLO appears to be the most capable when it comes to almost all aspects required when deciding upon a detection algorithm. When it comes to detecting objects the rate and speed of detection using YOLO is almost unmatched - making it the clear choice for our program.

REFERENCES

[1] "Speeding - Injury Facts", *Injury Facts*, 2021. [Online]. Available: https://injuryfacts.nsc.org/motor-vehicle/motor-vehicle-safety-issues/speeding/#:~:text=Speeding%20was%20a%20factor%20in,attributable%20to%20speeding%20was%208%2C544. [Accessed: 24- May- 2021].

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[3] P. by J. Świeżewski, “YOLO algorithm and YOLO object detection: An introduction,” Appsilon.com, 22-May-2020. [Online]. Available: https://appsilon.com/object-detection-yolo-algorithm/. [Accessed: 02-Mar-2021].

[4] "R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms", *Medium*, 2021. [Online]. Available: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e. [Accessed: 24- May- 2021].