Road Vehicle Detection Through YOLOv5

Juntao Lin
Department of Electrical and
Computer Engineering
Queen's University
Kingston, ON, Canada
21jl80@queensu.ca

***Abstract*—Autonomous driving is expected to be part of the future transportation system. In urban city, there is a gap between the ideal and actual commute time. With the help of autonomous vehicle, travelers can make a much better use of their commute time. Computer vision is part of the perceptron system in autonomous vehicle that makes fully autonomous driving possible in the future. This paper focuses on the road vehicle detection problem. We validate the performance of different YOLOv5 models on a custom dataset that contains images for 5 types of vehicles. After fine tuning the YOLOv5s model with a two-state training method, the model achieved a final 0.778 mAP.**

I. Introduction

Autonomous vehicles (AV) are expected to a part of the future transportation system. In urban cities, the average commute time is often high than what people demanded. Ye et al. published an analysis on commute satisfaction. Among the 833 participants, only 20% of them traveled with their ideal commute time. The discrepancy between the actual and real commute time significantly lowered the travel satisfactory [1]. AV on the other hand can help to lower the cost of travel time. Steck et al. showed the impact of autonomous driving on value of travel time savings (VTTS). They found out driving a private AV would reduce the VTTS by 31% [2].

Road vehicle detection is a very popular application under the category of object detection and computer vision. This type of algorithm aims to identify the detailed vehicle information from the source images or video streams. In recent years, road vehicle detection based on vision information has gain more attention. Unfortunately, the performance of this type of algorithms tends to be affected by the dynamics in the environment, such as varieties of vehicle shapes, weather conditions, lighting conditions, and driving behaviors [3].

The operation of autonomous vehicles requires a very precise perception system that reflect the nearby environment accurately. The perception system often includes a machine learning or deep learning algorithm that provides many options to process sensors' data. For example, through feature extraction algorithms, visual data can be transformed into semantic information, and then fed to the navigation module used in the AV.

Deep convolution networks have proved their accuracy and efficiency on image classification. With sufficient training data, deep learning models with Convolution Neural Network (CNN) pipeline can achieve the state-of-the-art performance. Object detection is a more challenging task compares to the image classification. Object detection algorithms must identify the class of the object, as well as the location of the object. These algorithms first generate multiple proposals that indicate the possible object locations. Followed by combining and refining the proposals to obtain an accurate localization. While designing such an algorithm, tradeoffs between speed, accuracy, and simplicity should be considered [4].

II. Related Work

A. Object Proposal

Shi et al. [5] proposed a PointRCNN network to detect object from raw point cloud data. The proposed network uses an encoder-decoder structure based on the PointNet++ [6] to generate point-wise feature vectors. Analyzing the feature vectors allows the network to classify the point cloud data into foreground and background. Foreground points are closely associated with the location and orientation of the object. Using the features from the foreground points, the network generates 3D box proposals.

B. 2D Object Detection

Ranft et al. [7] reviewed the roles of machine learning in autonomous driving that include localization, mapping, object classification, object localization, as well as computing architectures used for real-time applications.

Girshick [4] proposed a Fast Region-based Convolutional Network (Fast R-CNN) model for object detection. The model takes the whole input image with proposals as inputs. Convolution layers and max pooling layers extract features from the inputs, the extracted features will be saved in a feature map. For each object proposal, a Region of Interest (ROI) pooling layer is used to extract deeper features from the feature map and saved in a fixed length feature vector. The feature vector will be fed to a series of Fully Connected (FC) layers to generate the label and the location of the object. The Fast R-CNN was validated through the Visual Object Classes Challenge 2007 (VOC2007) [8], VOC2010 [9], and VOC2012 [10]. The R-CNN model achieved over 68% of mean Average Precision (mAP) across the three datasets.

C. 3D Object Detection

Chen et al. [11] proposed a 3D object detection model for monocular camera images. This model uses a region proposal algorithm based on semantic segmentation, instance segmentation, shape, context, and location to generate class specific candidate bounding boxes in 3D. These bounding boxes are the inputs for a standard CNN pipeline to obtain the 3D object detection. The model was validated through the KITTI dataset and achieved the averages of 83%, 74%, 67% Average Precision (AP) on KITTI's [12] easy, moderate, and hard test set respectively.

D. YOLO Series

Table 1 below summarizes the performance of different version of YOLO models compared with other state-of -the-art models.

Reference	Dataset Used	Algorithms	Findings
Li et al., 2021 [13]	Remote sensing images collected from GF-1 and GF-2 satellites.	SSD Faster R-CNN YOLO v3	YOLOv3 has higher mAP and FPS than SSD and Faster R-CNN algorithms.
	Training: 826 images.		
	Testing: 275 images.		
Benjdira et al., 2019 [14]	UAV dataset	Faster R-CNN YOLOv3	YOLOv3 has higher F1 score and FPS than Faster R-CNN.
	Training: 218 Images		
	Test: 52 Images		
Zhao et al., 2019 [15]	Google Earth and DOTA dataset	SSD Faster R-CNN YOLOv3	YOLOv3 has higher mAP and FPS than Faster R-CNN and SSD.
	Training: 224 Images		
	Test: 56 Images		
Kim et al., 2020 [16]	Korea expressway dataset	SSD Faster R-CNN YOLOv4	YOLOv4 has higher accuracy, SSD has higher detection speed.
	Training: 2620		
	Test: 568		
Dorrer et al., 2020 [17]	Custom Refrigerator images	Mask-RCNN YOLOv3	The detection of YOLOv3 was 3 times higher but the accuracy of Mask RCNN was higher.
	Training: 800 Images		
	Test: 70 Images		
Rahman et al., 2021 [18]	Custom Electrical dataset	YOLOv4 YOLOv5l	YOLOv4 has higher mAP compared to YOLOv5l algorithms.
	Training: 5939		
	Test: 1400		
Long et al., 2020 [19]	MS COCO dataset	YOLOv3 YOLOv4	YOLOv4 has higher mAP compared to YOLOv3.
	Training: 118,000		
	Test: 5000		
Bochkovskiy et al., 2020 [20]	MS COCO dataset	YOLOv3 YOLOv4	YOLOv4 has higher mAP and fps than YOLOv3.
	Training: 118,000		
	Test: 5000		
Ge et al., 2021, [21]	MS COCO dataset	YOLOv3 YOLOv4 YOLOv5	YOLOv5 has higher mAP than YOLOv3 and YOLOv5l, YOLOv3 has higher FPS than YOLOv4 and YOLOv5l.
	Training: 118,000		
	Test: 5000		

Table 1: Comparing YOLO models with other object detection algorithms [22]

III. Methods

There are multiple YOLOv5 models available. Figure 1 summarizes the characteristics for different models. The YOLOv5 model has three main components: model backbone, model neck, and model head.

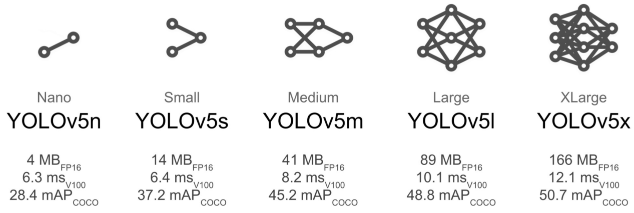


Fig. 1. YOLOv5 model specifications that include parameter size (MB), processing time (ms), and the performance measured in mAP [23].

A. Model Backbone

The module backbone is responsible for extracting features and patterns from the input images. The model backbone inherits the Cross Stage Partial Network (CSPNet) architecture to perform feature extraction. The vanilla CSPNet model speed up the ResNet, ResNeXt, and DenseNet models by 20% without lowering the accuracy [24]. The comparison of CSPNet and other models is shown in Figure 2.

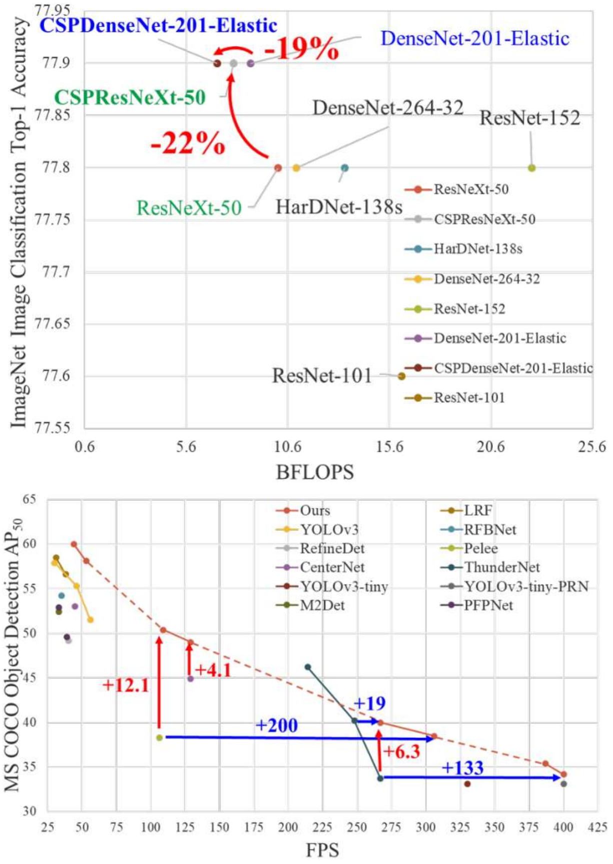
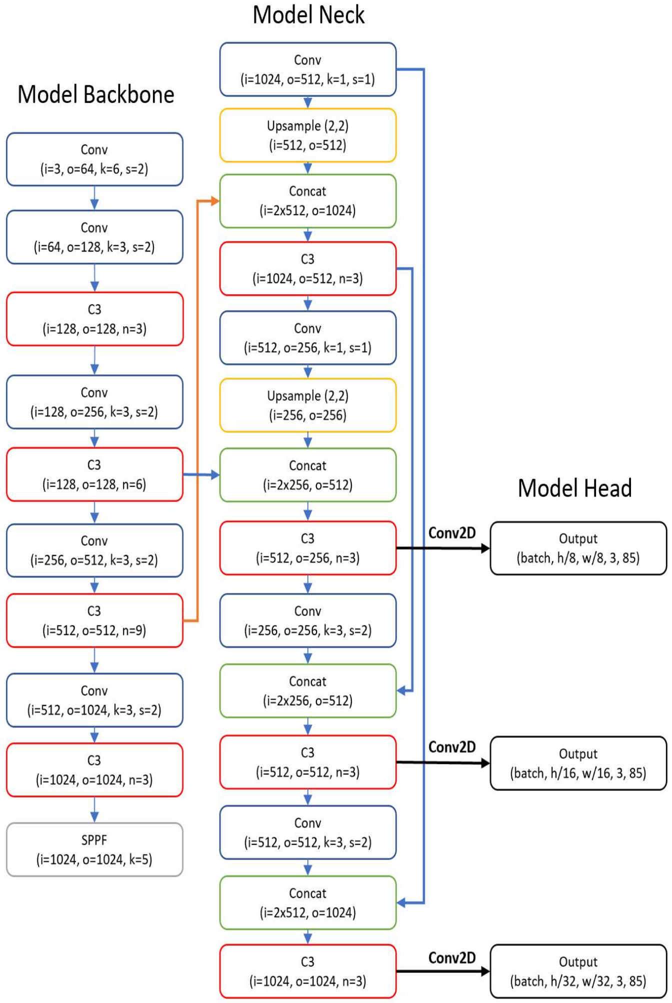


Fig. 2. Comparison of CSPNets, ResNet, ResNeXt, and DenseNet [24]

The model backbone uses the CSP-Darknet53 network structure that consists of 5 convolution layers, 3 C3 layers, and a Spatial Pyramid Pooling – Fast (SPPF) layer.

Every convolution layer uses 2D convolution, 2D batch normalization, and SiLU activation function. The C3 layer includes 3 convolution layers. The input received by the C3 layer will go through 2 convolution layers in parallel. The outputs of the two layers are concatenated, and then fed to the 3rd convolution layer as input.

At each convolution layer in the model backbone, the stride size is 2 (not including the C3 layer). After the input image propagate through the entire backbone model, the channel size will increase from 3 channels (usually input image contains RGB channels) to 1024 channels. The image size will be downsampled to 1/32 of the original size. At the end of the model backbone, a SPPF layer is added to make sure the model is more robust towards object deformations. The output of the SPPF layer will be fed to the first convolution layer in the model neck as input.



Model Neck
Model Backbone
Model Head
Conv (i=3, o=64, k=6, s=2)
Conv (i=64, o=128, k=3, s=2)
C3 (i=128, o=128, n=3)
Conv (i=128, o=256, k=3, s=2)
C3 (i=128, o=128, n=6)
Conv (i=256, o=512, k=3, s=2)
C3 (i=512, o=512, n=9)
Conv (i=512, o=1024, k=3, s=2)
C3 (i=1024, o=1024, n=3)
SPPF (i=1024, o=1024, k=5)
Conv (i=1024, o=512, k=1, s=1)
Upsample (2,2) (i=512, o=512)
Concat (i=2x512, o=1024)
C3 (i=1024, o=512, n=3)
Conv (i=512, o=256, k=1, s=1)
Upsample (2,2) (i=256, o=256)
Concat (i=2x256, o=512)
C3 (i=512, o=256, n=3)
Conv (i=256, o=256, k=3, s=2)
Concat (i=2x256, o=512)
C3 (i=512, o=512, n=3)
Conv (i=512, o=512, k=3, s=2)
Concat (i=2x512, o=1024)
C3 (i=1024, o=1024, n=3)
Conv2D
Output (batch, h/8, w/8, 3, 85)
Conv2D
Output (batch, h/16, w/16, 3, 85)
Conv2D
Output (batch, h/32, w/32, 3, 85)

Fig. 3. YOLOv5 architecture. 'i' is input channel, 'o' is output channel, 'k' is kernel size, 's' is stride size, 'n' is number of convolution blocks.

B. *Model Neck*

The model neck generates the feature pyramid that helps the model to process objects with different sizes. This technique helps to generalize scaling for objects with the same label, as well as identifying objects not seen in the training set. The YOLOv5 model uses the Path Aggregation Network (PANet) [25] structure as the model neck. The detailed model neck structure is shown in Figure 3. The PANet helps to improve the information flow within the YOLOv5 model with enhanced bottom-up paths. The bottom-up path shortens the travel path between the features in lower layers and the deeper layer. Therefore, the localization information can be accessed by the topmost layers more accurately [25]. The bottom-up paths are visualized in Figure 3 indicated by the orange and blue arrows that connect to concatenation blocks.

C. *Model Head*

The model head performs the final prediction on objects that includes bounding box locations, object labels, and probability for each object class.

IV. EXPERIMENT RESULT

In this section, the experiment results are presented for road vehicle detection through different YOLOv5 models. The optimizer and the loss function used for the experiments are respectively SGD and Binary Cross-Entropy with Logits Loss.

A. *Evaluation*

The performance of the model will be measured in precision, recall, mAP, and F1 score. The Intersection over Union (IoU) measures the percentage overlay of two bounding boxes. The quality of the proposal is proportional to the area overlay between the model prediction and the ground truth. For each object class, the AP can be calculated by taking the area under the precision-recall curve at a preset IoU threshold. The mAP is the mean value of APs from each object class.

$$\text{Precision} = \frac{TP}{TP + FP} \qquad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \qquad (2)$$

$$\text{F1} = \frac{TP}{TP + \frac{1}{2}FP} \qquad (3)$$

$$IoU(A, B) = \frac{A \cap B}{A \cup B} \qquad (4)$$

$$mAP = \frac{1}{n}\sum_{i=1}^{n} AP_i \qquad (5)$$

B. *Dataset*

The custom dataset [26] used in this project has 1321 images that include the following classes: car, motorcycle, truck, bus, bicycle. Multiple objects can appear in the same image. The dataset has been divided into a training set of 1142 samples, and a validation set of 125 samples.

C. *Fine Tuning vs. No Fine Tuning*

The YOLOv5 model is pretrained with the COCO dataset [27] that has more than 200K labelled images, 1.5 million object instances, and 80 object categories. The COCO dataset includes the five vehicle classes in the custom dataset. The following experiment will test performance of the YOLOv5s model on the custom dataset with and without fine tuning.

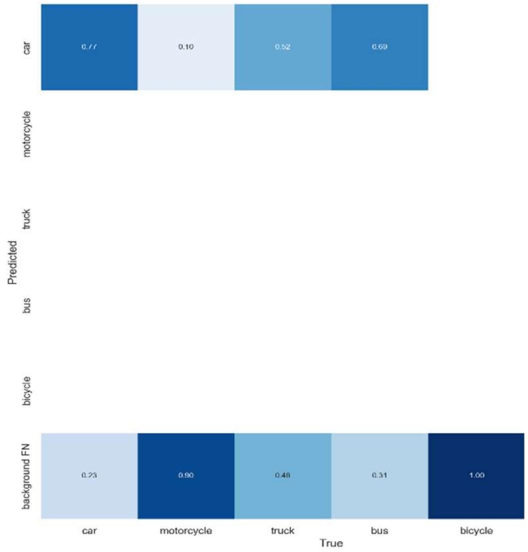


Fig. 4. Training YOLOv5s with 1 epoch

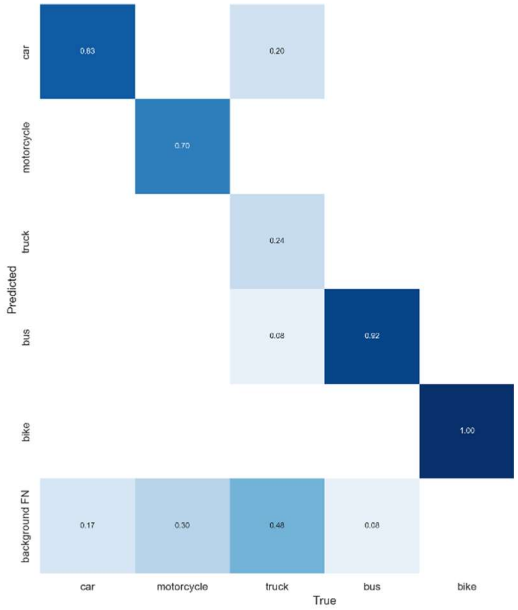


Fig. 5. Training YOLOv5s with 50 epochs

Figure 4 and Figure 5 show the differences in the model's performance when the YOLOv5s model was trained for 1 and 50 epochs. Although the COCO dataset includes images for all type of vehicles appeared in the custom dataset, the model had a poor performance in recognizing motorcycle, truck, bus, and bicycle at a 0.157 mAP without fine tuning. Most of the misclassified objects were recognized as either car or background. During fine tuning, the training and validation losses decreases significantly over training epoch. After 50 epochs of training, the model reached 0.775 mAP.

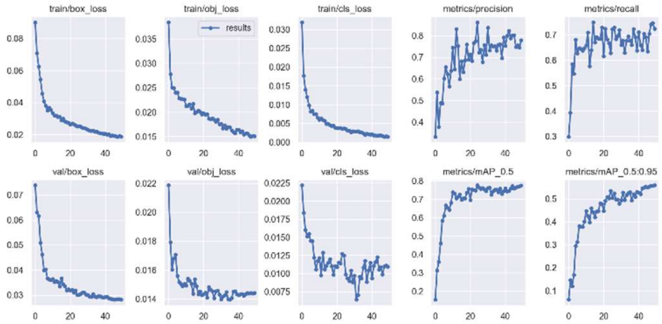


Fig. 6. Training and validation losses

D. *Performance of Different YOLOv5 models*

YOLOv5 models with different sizes are trained and validated using the same dataset. The training epoch on each model is set to 50. The performance of each model is concluded in Table 1 below. The YOLOv5x is not included due to computational complexity.

Model	mAP@0.5	mAP@0.5:0.95	box_loss	obj_loss	cls_loss
YOLOv5n	0.75445	0.52634	0.03065	0.01415	0.00964
YOLOv5s	0.77552	0.55914	0.02819	0.01441	0.01095
YOLOv5m	0.74246	0.55609	0.02846	0.01508	0.01402
YOLOv5l	0.76635	0.57751	0.02838	0.01549	0.01066

Table 1. Comparison of different YOLOv5 model

Each model's performance was evaluated for mAP at 0.5 IoU threshold (mAP@0.5), average of mAP values over different IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95), box loss, objectness loss, and classification loss.

Increasing the model size does not necessarily improve the performance. The YOLOv5s has the best mAP among the other YOLOv5 models.

E. *Detection Accuracy per Object Class*

Other than the confusion matrix, the performance of the model can also be visualized in the PR curve. Figure 7 shows the performance of the YOLOv5s model at 0.5 IoU threshold. The model has a good performance on most of the object classes, except the truck. The AP for truck is only 0.438, where other classes have AP between 0.70~0.97. Since the custom dataset has only 1000+ samples and not all the images are labelled properly, we decide to add truck images with precise labels to the training set.

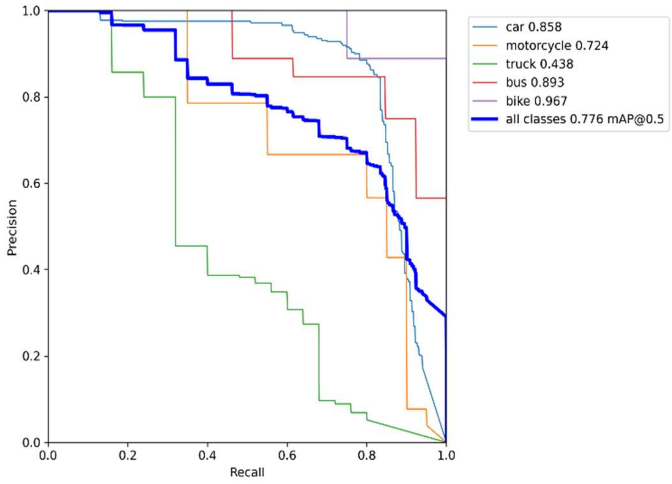


Fig. 7. YOLOv5s PR curve at 0.5 IoU threshold

Adding additional 15 truck images to the original training set does not influence the model's performance, neither as adding the truck samples to the validation set.

In the next experiment, the truck samples are separated from the original dataset. Both truck samples and the custom dataset are considered as independent dataset, a two-stage training method is proposed. In stage one, the YOLOv5s model will be trained with the truck samples (15 images) for 50 epochs. In state two, the YOLOv5s model will be further trained on the custom dataset (1321 images) for another 15 epochs. Figure 8 shows result of the simulation. The mAP of the YOLOv5s model remains the same, the AP for truck object increased from 0.438 to 0.5.

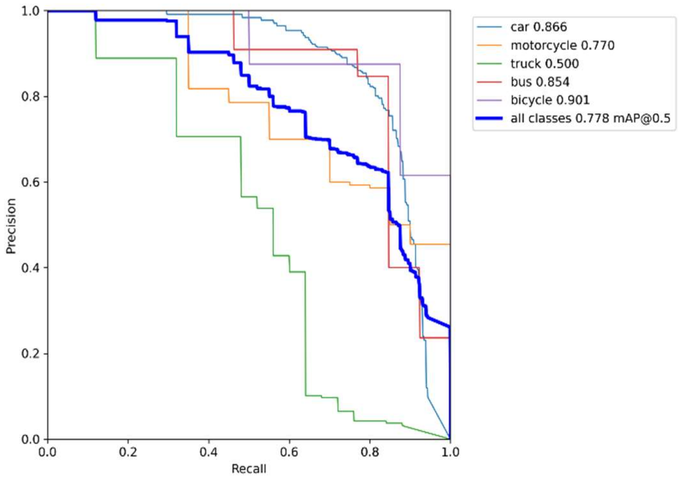


Fig. 8. YOLOv5s PR curve at 0.5 IoU threshold, addition training with truck images

F. Results and Factors Influence the Performance

There are multiple factors that affect the detection accuracy of the YOLOv5 model. In this section, we will focus on the impacts of dataset quality on the model's performance, followed by demonstrating the results achieved by the model. The custom dataset contains 1121 samples of images of road vehicles. Multiple vehicles of the same type and different types of vehicles can appear in the same image. To understand why the YOLOv5s model has such a different performance on different vehicles, we decide to take a closer look at the dataset. Table 2 summarizes how many times the target vehicles appeared during the training and validation.

	Car	Motorcycle	Truck	Bus	Bicycle
Number of instances in training	2268	221	133	176	76
Number of instances in validation	230	20	25	13	8

Table 2: Number of instances for each type of vehicle

Apparently, there is a data imbalance between the object classes. The number of cars vs. other vehicles is at a 10~20:1 ratio. This imbalance causes the model to focus on the car detection rather than other vehicles. Figure 8 shows that the PR curve for car has the smoothest gradient with a relatively high AP. Although bicycle has a higher AP than car, we suspect that is due to a lack of sample in the validation set, given there are 230 instances of car and 8 instances of bicycle in validation. Figure 9 shows the F1 scores for the vehicles. As expected, car objects have the highest F1 score as the confidence increases.

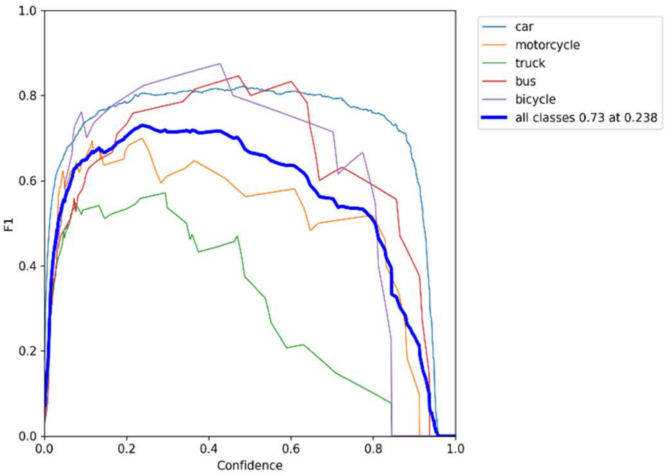


Fig. 9. F1 curves of different vehicles

Another factor affects the dataset quality is the labeling. Figure 10 shows the ground truth used in the training set; Figure 11 shows the labels generated by the YOLOV5s model. When there are multiple objects in the same image, the quality of the labels included in the dataset is not optimal. In some cases, the labels generated by the YOLOv5 network are better than the dataset. In Figure 9, the image in the last row, second column, includes a motorcycle on the road, two cars piled against the wall, and a wheel that is partially captured. The dataset labelled the not functional cars and the wheel to be the objects in this image, whereas the YOLOV5s model marked these objects as background, and successfully identified the motorcycle.

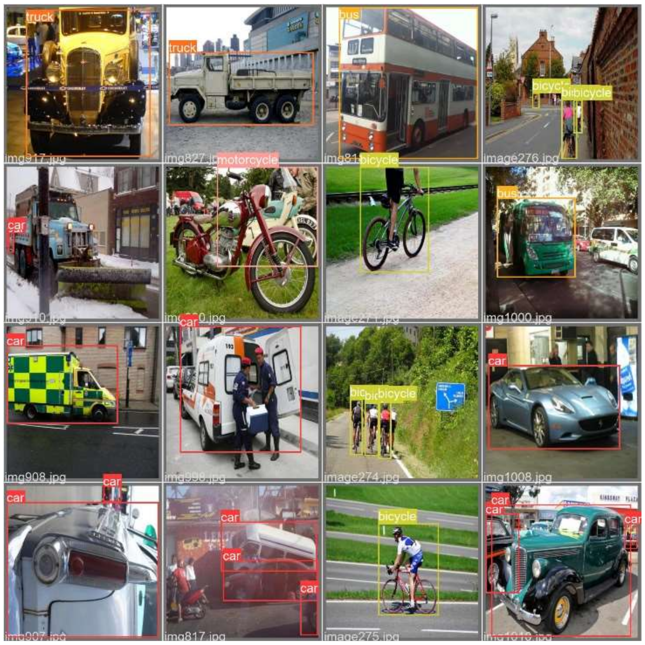


Fig. 10. Images and labels in the validation set

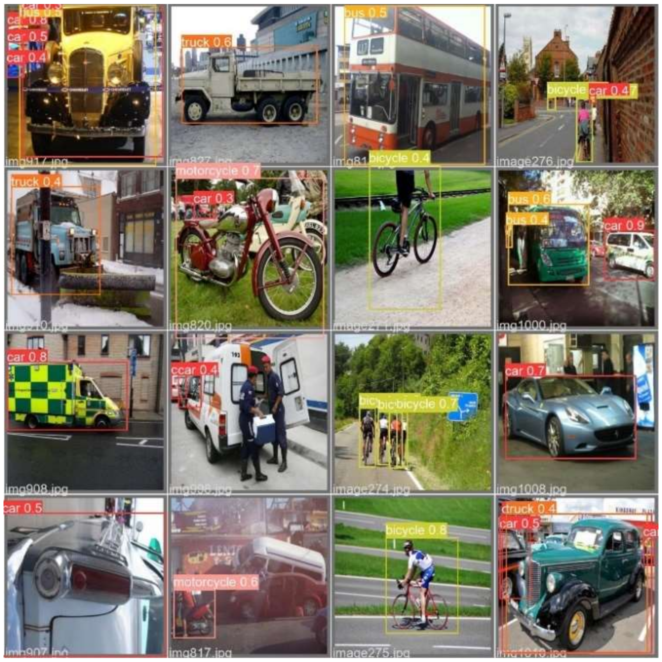


Fig. 11. Road vehicle detection by YOLOv5s model

V. Conclusion

The YOLOv5 models have a great potential for detecting road vehicles. Fine tuning is very important while performing object recognition on custom datasets such as images collected from deployed sensors. The performance of the YOLOv5 model is proportional to the quality of the training data. The quality of a custom dataset can be evaluated by the number of instances of each target object, and the quality of labeling. We demonstrated that the AP of a single object class could be improved by using the two-stage training technique that requires additional training data for the target object class. Among the YOLOv5 models, the YOLOv5s has the best performance and reaches a final mAP of 0.778 at 0.5 IoT threshold.

VI. References

[1] Runing Ye, Jonas De Vos and Liang Ma, "Analysing the association of dissonance between actual and ideal commute time and commute satisfaction," *Transportation Research Part A: Policy and Practice,* vol. 132, pp. 47-60, 2020.

[2] F. Steck, F. Bahamonde-Birke, S. Trommer and B. Lenz, "How Autonomous Driving May Affect the Value of Travel Time Savings for Commuting," *Transportation Planning Applications,* vol. 2672, no. 46, pp. 11-20, 2018.

[3] A. Mukhtar, L. Xia and T. B. Tang, "Vehicle Detection Techniques for Collision Avoidance Systems: A Review," *IEEE Transactions on Intelligent Transportation Systems,* Vols. 16, no. 5, pp. 2318-2338, 2015.

[4] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV),* pp. 1440-1448, 2015.

[5] S. Shi, X. Wang and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 770-779, 2019.

[6] C. R. Qi, L. Yi, H. Su and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on," *Neural Information Processing Systems,* p. 5099–5108, 2017.

[7] B. Ranft and C. Stiller, "The Role of Machine Vision for Intelligent Vehicles," *IEEE Transactions on Intelligent Vehicles,* vol. 1, no. 1, pp. 8-19, 2016.

[8] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007)," *IJCV,* 2007.

[9] M. Everingham , L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010)," *IJCV,* 2010.

[10] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012)," *IJVC,* 2012.

[11] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Findler and R. Urtasun, "Monocular 3D Object Detection for Autonomous Driving," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2147-2156, 2016.

[12] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.,* vol. 32, no. 11, pp. 1231-1237, 2013.

[13] M. Li, Z. Zhang, L. Lei, X. Wang and X. Guo, "Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of Faster R-CNN, YOLOv3 and SSD," *Sensors,* 2020.

[14] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar and K. Ouni, "Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3," *In Proceedings of the International Conference on Unmanned Vehicle Systems-Oman (UVS),* pp. 1-6, 2019.

[15] K. Zhao and X. Ren, "Small aircraft detection in remote sensing images based on YOLOv3," *In Proceedings of the IOP Conference Series: Materials Science and Engineering,* 2019.

[16] J.-A. Kim, J.-Y. Sung and S.-H. Park, "Comparison of Faster-RCNN, YOLO, and SSD for real-time vehicle type recognition," *In Proceedings of the IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia),* pp. 1-4, 2020.

[17] M. Dorrer and A. Tolmacheva, "Comparison of the YOLOv3 and Mask R-CNN architectures' efficiency in the smart refrigerator's computer vision," *J. Phys. Conf. Ser.,* 2020.

[18] E.U. Rahman, Y. Zhang, S. Ahmad, H.I. Ahmad and S. Jobaer, "Autonomous vision-based primary distribution systems porcelain insulators inspection using UAVs," *Sensors,* vol. 21, no. 3, p. 974, 2021.

[19] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han and E. Ding, "PP-YOLO: An effective and efficient implementation of object detector," *arXiv,* 2020, arXiv:2007.12099.

[20] A. Bochkovskiy, C.-Y. Wang and H.-Y.M Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv,* 2020, arXiv:2004.10934.

[21] Z. Ge, S. Liu, F. Wang, Z. Li and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv,* 2021, arXiv:2107.08430.

[22] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs," *Remote Sensors,* 2022.

[23] G. Jocher, Available online: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data (accessed on 16 April 2022)..

[24] C. Wang, H. M. Liao, Y. Wu, P. Chen, J. Hsieh and I. Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 390-391, 2020.

[25] K. Wang, J.H. Liew, Y. Zou, D. Zhou and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," *In Proceedings of the IEEE International Conference on Computer Vision (ICCV),* p. 9197–9206, 2019.

[26] A. Benjumea, I. Teeti, F. Cuzzolin and A. Bradley, "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," *arXiv,* 2021, arXiv:2112.11798.

[27] H. Park, Y. Yoo, G. Seo, D. Han, S. Yun and N. Kwak, "C3: Concentrated-Comprehensive Convolution and," *arXiv,* 2018, arXiv:1812.04920.