Analysis of Object Detection Models on Duckietown Robot Based on YOLOv5 Architectures

Toan-Khoa Nguyen, Lien T. Vu, Viet Q. Vu*, Tien-Dat Hoang
Shu-Hao Liang, *Member, RST,* and Minh-Quang Tran*, *Member, RST*

***Abstract*— Object detection technology is an essential aspect of the development of autonomous vehicles. The crucial first step of any autonomous driving system is to understand the surrounding environment. In this study, we present an analysis of object detection models on the Duckietown robot based on You Only Look Once version 5 (YOLOv5) architectures. YOLO model is commonly used for neural network training to enhance the performance of object detection models. In a case study of Duckietown, the duckies and cones present hazardous obstacles that vehicles must not drive into. This study implements the popular autonomous vehicles learning platform, Duckietown's data architecture and classification dataset, to analyze object detection models using different YOLOv5 architectures. Moreover, the performances of different optimizers are also evaluated and optimized for object detection. The experiment results show that the pre-trained of large size of YOLOv5 model using the Stochastic Gradient Decent (SGD) performs the best accuracy, in which a mean average precision (mAP) reaches 97.78%. The testing results can provide objective modeling references for relevant object detection studies.**

Index Terms*— *Object detection, Duckietown robot, YOLOv5 architectures, optimization functions.

I. Introduction

RECENTLY, the intelligence of Autonomous Guided Vehicles (AGVs) has attracted much attention from the industry. An autonomous driving vehicle of a self-driving vehicle is a highly complicated system, which builds upon a diversity of inputs from sensors including radars, cameras, and recently LiDAR sensors [2]. The system needs to detect objects in their vicinity, distinguish one object from another, and predict their possible motions with corresponding uncertainty. Numerous works were recently proposed to optimize object detection from raw sensor data, as well as to distinctively classify them from one another, where the objects themselves are the inputs [3]. The two tasks are completed individually and consecutively using a single algorithm, with the detection and discrimination modules being trained separately. However, as a result of lacking feature sharing, such stacked systems may experience excessive system latency as well as cascading errors, where gradients can be continuously transmitted back to the sensor data flow from output trajectories to the detection modules.

The AGV must be able to detect and distinguish obstacles that enabling for path planning with no human required. Therefore, reliable object detection is a crucial contributor to autonomous driving [4]. Object detection is regarded as a crucial branch in the area of computer vision and image processing with its algorithm has been focused recently in the field of deep learning. It has been witnessed the tremendous growth of machine learning and deep learning in recent years and taken real-time object identification to the next level [5], such that deep learning approaches unveiled promising performance in comparison with the traditional methods in regard to detection accuracy. Deep learning object detection consists of two types: one-stage object detection [6] and two-state object detection [7]. Among those two deep learning techniques, YOLO is considered as a practical algorithm for online object detection since bounding boxes and confidences for various categories may be created directly from complete photos by using a proper neural network.

Overall visual object detection is categorized into two types: the region proposal method. The first one is the region proposal method with convolutional neural networks (R-CNNs) [8], Fast R-CNN [9], Faster R-CNN [10], and Faster R-CNN model [10], and the second is the end-to-end method with the YOLO model [11], Single-Shot Detector (SSD) method and RetinaNet network [12], and so on. Speaking of object detection speed, the end-to-end method shows superior performance compared to the region proposal methods [13]. Due to the development of visual object recognition technology, the YOLO series of algorithms have been used in various scene detection tasks and it performs at very high precision and speed [14]. Additionally,

This paper was submitted on January 10, 2022.

Part of this paper was presented at the 2021 International Automatic Control Conference (CACS) [1]. This paper was supported by the Center for Cyber-Physical System Innovation, which is a Featured Areas Research Center in Higher Education Sprout Project of Ministry of Education (MOE), Taiwan (since 2018). Part of funding also came from the Ministry of Science and Technology (MOST) in Taiwan under Grant no. MOST 110-2222-E-011-002-. (*corresponding author: Minh-Quang Tran and Viet Q. Vu*)

Toan-Khoa Nguyen currently is with Department of Electrical Engineering at National Taiwan University of Science and Technology, Taipei 10607, Taiwan (e-mail: M10907803@mail.ntust.edu.tw).

Lien T. Vu is with the Faculty of Mechanical Engineering and Mechatronics, Phenikaa University, Ha Dong, Hanoi 12116, Viet Nam (e-mail: lien.vuthi@phenikaa-uni.edu.vn)

Viet Q. Vu and Tien-Dat Hoang are with the Faculty of International Training, Thai Nguyen University of Technology, 3/2 Street, Tich Luong ward, 250000 Thai Nguyen, Vietnam (e-mail: vuquocviet84@gmail.com; hoangdat@tnut.edu.vn)

Shu-Hao Liang is with the Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei 10607, Taiwan (e-mail: shuhaoliang@mail.ntust.edu.tw).

Minh-Quang Tran is with the Industry 4.0 Implementation Center, Center for Cyber-Physical System Innovation, National Taiwan University of Science and Technology, Taipei 10607, Taiwan and also with the Department of Mechanical Engineering, Thai Nguyen University of Technology, 3/2 Street, Tich Luong ward, 250000 Thai Nguyen, Vietnam (e-mail: minhquang.tran@mail.ntust.edu.tw).

the YOLO system has the ability to process all of the image's features and the majority of the objects can be predicted. [15].

YOLOv5 model is the fifth and also the latest generation of YOLO. According to various experiments, it outperforms the rest of the YOLO model in terms of both speed and accuracy. Previous studies in the literature show that the YOLOv5 model has been recently successfully used to detect mold on food surfaces for the first time in the present study [15]. Moreover, in some recent studies, YOLOv5 has even been utilized to detect a variety of objects such as apples, mushrooms, marine ships, face masks, vehicles, safety helmets, and, etc. Therefore, it can detect and distinct obstacles from the captured images [16].

This study implements the popular autonomous vehicle learning platform which is Duckietown's platform and dataset [17] to demonstrate the comparison of different object detection models and optimizers in order to give a detailed analysis in object detection and classification tasks. In which, the latest version of the YOLO model, YOLOv5, is implemented to test object detection models. Various YOLOv5 architectures with different sizes are utilized to analyze object detection models on the Duckietown's platform and dataset. Moreover, the performances of different optimizers are also evaluated and optimized for object detection. Furthermore, we explore the major parameters to improve the performance of the object detection model using YOLOv5.

II. Materials and Methods

Duckietown robots are fully autonomous, in which every decision making is processed onboard using Raspberry Pi and NVIDIA Jetson Nano boards [18]. The configuration of a Duckietown robot is shown in Fig. 1 (a). It is equipped with a fish-eye lens camera in the front, two DC motors, a 32GB memory card, and an onboard battery with a power of 5V. A Python-based Robotic Operating System (ROS) is included and utilized to support the communication between the perception, planning, and control functions on the Duckietown robots.

Fig. 1. (a) Duckietown robot, (b) cones, and (c) duckies.

Duckietown dataset is known as an object detection benchmark dataset from AI Driving Olympics. In this study, the collected datasets consist of duckies and cones classes, as illustrated in Fig. 1 (b) and (c). The total of 1006 samples in the dataset are then separated into the training and testing set with the split percentage of 80:20, respectively. The training process was implemented using NVIDIA Tesla V100 on the Industry 4.0 Center, Taiwan Tech. Several YOLOv5 models, including with and without pre-trained models using Stochastic Gradient Descent (SGD) and Adam optimizers, are deployed to perform the object detection for the Duckietown robot. The difference between pre-trained model and the ordinary model is the pre-trained model is already trained on one or multi public benchmarks and has optimal weights compared to the ordinary model is just an architecture with initial random weights.

YOLOv5 architecture for the object detection is composed of three important pieces: backbone piece for extracting feature, neck piece for fusing feature, and head piece. The first part of YOLOv5 architecture is a convolutional network that aggregates and forms image features at different granularities from the original images using various layers of convolution and pooling [19]. It can be seen in Figure 2, the backbone network comprises four generated layers of feature maps with different feature sizes. Therefore, the neck network has a series of layers to mix and combine image features to achieve more contextual information and avoid information loss, and to pass them forward to prediction. In addition, in the fusion process of YOLOv5 architecture, the feature pyramid structures of the feature pyramid network (FPN) [20] and the pixel aggregation network (PAN) [21] are utilized. Then the important semantic features from the top feature maps are passed down through the FPN structure to the lower feature maps. Simultaneously, the PAN structure delivers strong localization features from lower to higher feature maps. The feature fusion capability of the neck network is leveraged by the combination of two structures. In detail, three feature fusion layers create the new feature maps. The larger the area of the image that each grid unit in the feature map corresponds to, the smaller the size of the feature maps. Finally from these new feature maps, the head network detects and classifies objects.

In the proposed model, the focus module in the architecture slices and concatenates images in order to generate the important features. The CBL module is made up of the convolution, normalization, and Leaky *relu* activation function modules [22]. YOLOv5 model has two types of cross-stage partial networks (CSP) [23]. One is applied to the backbone network, while the other is used in the neck. The CSP network uses cross-layer connectivity to connect the network's front and back layers, it helps to enhance the inference speed and remain the precision. The structure of the two types of CSP networks differs only slightly. The CSP network in the backbone is made up of one or more residual units, whereas the CSP network in the neck is made up of CBL modules that replace the residual units. Furthermore, the SPP module refers to the spatial pyramid pooling module, which performs maximum pooling with various kernel sizes and fuses the features by concatenating them together. The image features can be presented at a higher level of abstraction based on dimensionality reduction operations through the pooling layers to mimic the human visual system.

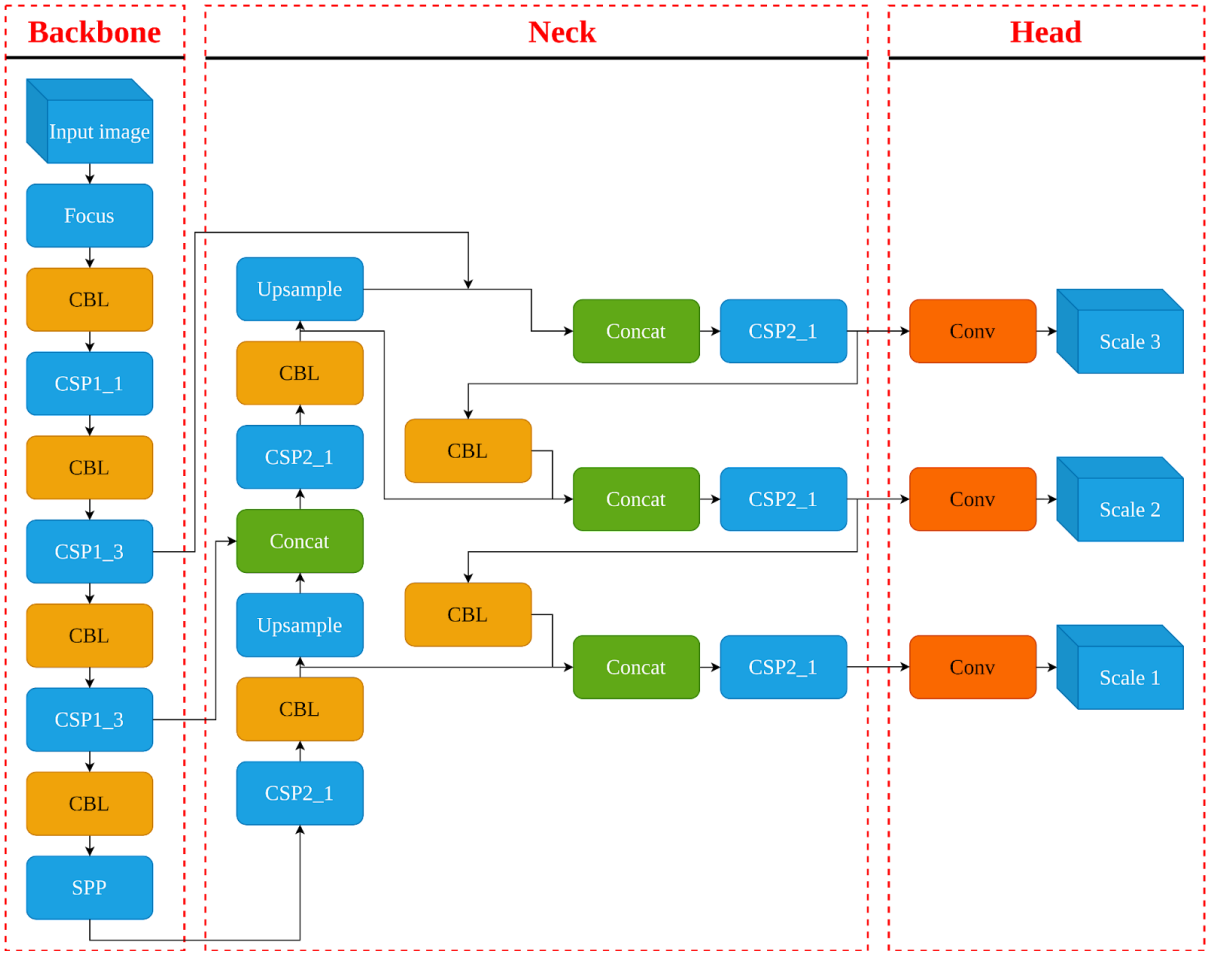

Fig. 2. The architecture of the YOLOv5 method. The network consists of three important pieces: backbone, neck, and head. The backbone network aggregates and forms image features at different granularities, the neck network mixes and combines image features to pass them forward to prediction, and the head network consumes features from the neck and takes box and class prediction steps.

It is primarily concerned with the compression of the input feature map. On the one hand, it shrinks the feature map and reduces the network's computational complexity; on the other hand, it performs feature compression and extracts the main features. In the end, the concat module performs tensor concatenation. In this work, the SGD algorithm [24] and Adam optimization algorithm [25] are implemented as optimizer functions to update the weights in the network iteratively based on training data in order to examine the performance of the YOLOv5 models. The weight update rule of the Adam optimization algorithm is formulated in Eq. (1) [26],

$$\theta_{t+1} = \theta_t - \frac{\eta \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{1}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{2}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{3}$$

$$m_t = (1-\beta_1)g_t + \beta_1 m_{t-1} \tag{4}$$

$$v_t = (1-\beta_2)g_t^2 + \beta_2 v_{t-1} \tag{5}$$

in which θ presents the model weights at time t; η is the learning rate; ϵ describes a small term preventing division by zero; m_t presents the aggregate of gradients at time t; v_t is the squared gradient where the hyper-parameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages m_t and v_t; g_t denotes the gradient at subsequent timestep t and $g_t = \nabla_\theta f_t(\theta)$ with the objective function f_t.

Whereas the SGD algorithm is described in Eq. (6), it examines one by one sample at each iteration and updates the weight vector ω iteratively using a time-dependent weighting factor, as shown in Eq. (6).

$$\omega_{t+1} = \omega_t - \frac{\eta}{t}\frac{\delta}{\delta\omega} l \quad f_t \quad x_t \quad , y_t \tag{6}$$

in which x_t is training example, and y_t is the label, η presents the update factor or step size utilized to update the solution ω_t at step t. f is the function that is linearly parameterized, and l presents the loss function. The weight update of the network

based on the SGD algorithm is usually much faster because of one update at a time, thus it can be used to learn online.

In this study, different sizes of YOLOv5 structure including small size (YOLOv5s), medium size (YOLOv5m), large size (YOLOv5l), and extra-large size (YOLOv5x) are used. The difference between small-sized (YOLOv5s), medium-sized (YOLOv5m), large-sized (YOLOv5l), and extra-large-size (YOLOv5x) is the number of parameters of the architecture. Different types of pre-trained YOLOv5 models and their parameters, as described in Table I.

TABLE I
DESCRIPTION OF YOLOV5 ARCHITECTURES

YOLOv5 MODEL	Model Size	No. of parameters (M)
YOLOv5s	Small size	7.2
YOLOv5m	Medium size	21.2
YOLOv5l	Large size	46.5
YOLOv5x	Extra-large size	86.7

III. RESULTS AND DISCUSSIONS

With regard to calculating the efficiency of different YOLOv5 architectures for the object detection on Duckietown robot, several indexes including precision, recall, and average precision (AP) are used, they are formulated in Eqs. (7)-(9),

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$AP = \frac{1}{11} \sum_{Recall(i)} Precision(Recall_i) \tag{9}$$

where *TP* is True Positives (Predicted as positive as was correct), *FP* is False Positives (Predicted as positive but was incorrect), and *FN* is False Negatives (Failed to predict an object that was there).

The outcomes of the training model are presented in Fig. 3, in which the input image size is 416 x 416, and the batch size of 16 and 100 epochs are utilized. The performance of the YOLOv5 for the object detection is very effective in terms of the mean precision average at 0.5 (mAP@0.5).

An analysis of object detection models on the Duckietown robot based on different YOLOv5 architectures is also described in Table II. The pre-trained YOLOv5m with SGD optimizer achieved 96.90% in terms of accuracy and 97.10% in terms of mAP@0.5. In addition, the pre-trained YOLOv5l with SGD optimizer achieved 95% in terms of accuracy and 97.80% in terms of mAP@0.5 (mAP with Intersection over Union (IoU) threshold of 0.5), and 66.6% in terms of mAP@0.5:0.9 (average mAP over different IoU thresholds, from 0.5 to 0.90). The testing results show that the uses of the pre-trained model have better performance than the training model within the collected dataset. It also confirms that the SGD optimizer can enhance the performance of the model compared to the Adam optimizer on our specific dataset, as clearly presented in Fig. 4. In this paper, the parameters are not fine-tuned to get the optimal ones to achieve the best performance on the Duckietown dataset.

Moreover, Zhou et al. [27] showed that SGD has a smaller escaping time than Adam for the same basin and whose local basins have larger Radon measure tends to converge to flatter minima, this demonstrates that its better generalization performance. Overall, the pre-trained model with a larger YOLOv5 architecture using SGD optimizer has the best performance for object detection models on the Duckietown dataset.

TABLE II
COMPARISON OF DIFFERENT YOLOV5 MODLES FOR OBJECT DETECTION

Model Architectures	**Precision**	**Recall**	**mAP @0.5**	**mAP@ 0.5:0.9**
Pre-trained_YOLO v5s_Adam	0.884	0.908	0.946	0.601
Pre-trained_YOLO v5m_ Adam	0.899	0.843	0.91	0.569
Pre-trained_YOLO v5l_ Adam	0.811	0.868	0.879	0.531
Pre-trained_YOLO v5x_ Adam	0.819	0.748	0.798	0.453
YOLOv5s_SGD	0.93	0.928	0.961	0.65
YOLOv5m_SGD	0.902	0.84	0.909	0.553
YOLOv5l_SGD	0.878	0.883	0.92	0.583
YOLOv5x_SGD	0.865	0.814	0.881	0.544
Pre-trained_YOLO v5s_SGD	0.929	0.939	0.969	0.654
Pre-trained_YOLO v5m_SGD	**0.969**	**0.914**	**0.971**	**0.666**
Pre-trained_YOLO v5l_SGD	**0.95**	**0.944**	**0.978**	**0.671**
Pre-trained_YOLO v5x_SGD	0.941	0.945	0.972	0.661

IV. CONCLUSION

This paper presents an analysis of object detection models on the Duckietown robot based on YOLOv5 architectures. The YOLOv5 model has been successfully used to recognize the duckies and cones on the Duckietown. Moreover, the performances of different YOLOv5 architectures are analyzed and compared. The results indicate that using the pre-trained model of YOLOv5 architecture with the SGD optimizer can provide excellent accuracy for object detection. The higher accuracy can also be obtained even with the medium size of the YOLOv5 model that enables to accelerate the computation of the system. Furthermore, once the object detection model is

optimized, it is integrated into the ROS in the Duckietown robot. In future works, it is potential to investigate the YOLOv5 with Layer-wise Adaptive Moments Based (LAMB) optimizer instead of SGD, applying repeated augmentation with Binary Cross-Entropy (BCE), and using domain adaptation technique.

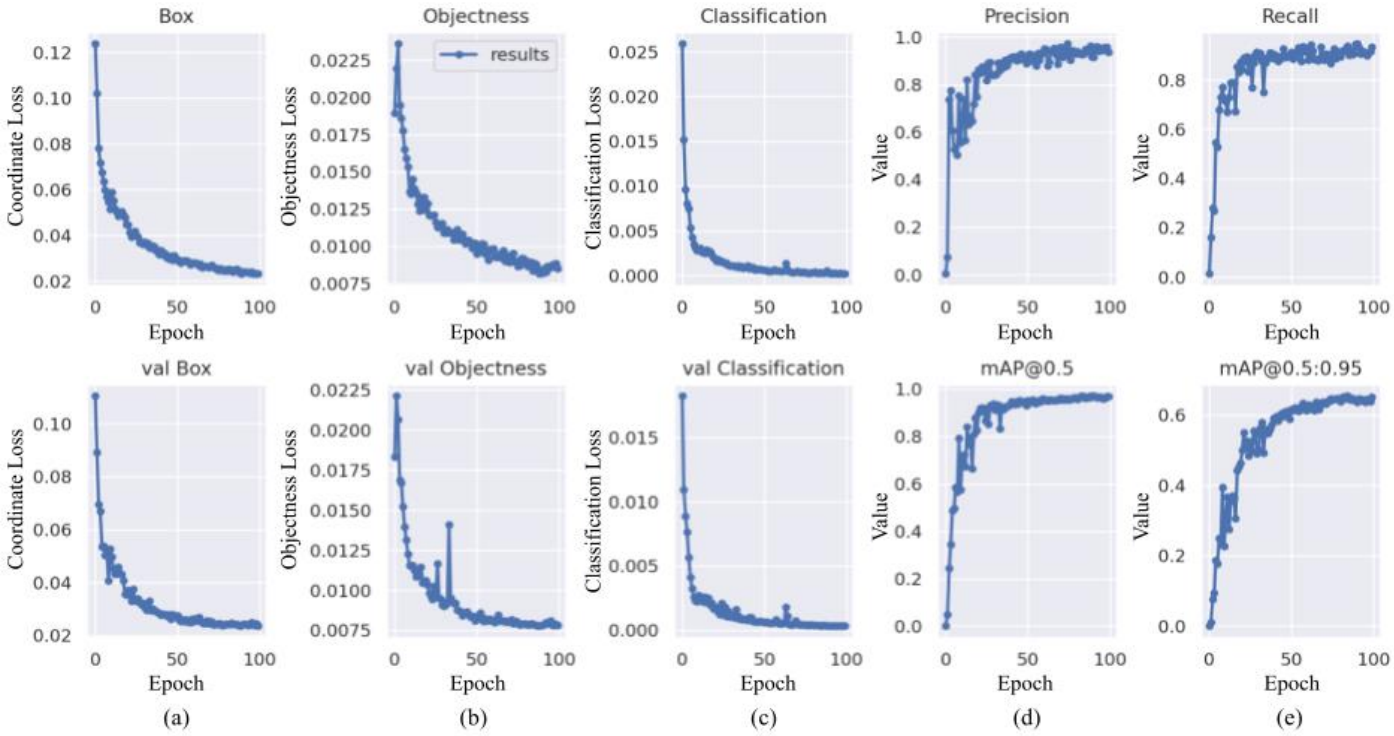

Fig. 3. Training model results of pre-trained YOLOv5s for object detection. (a) Coordinate Loss (b) Objectness Loss (c) Classification Loss (d) Precision (e) Recall.

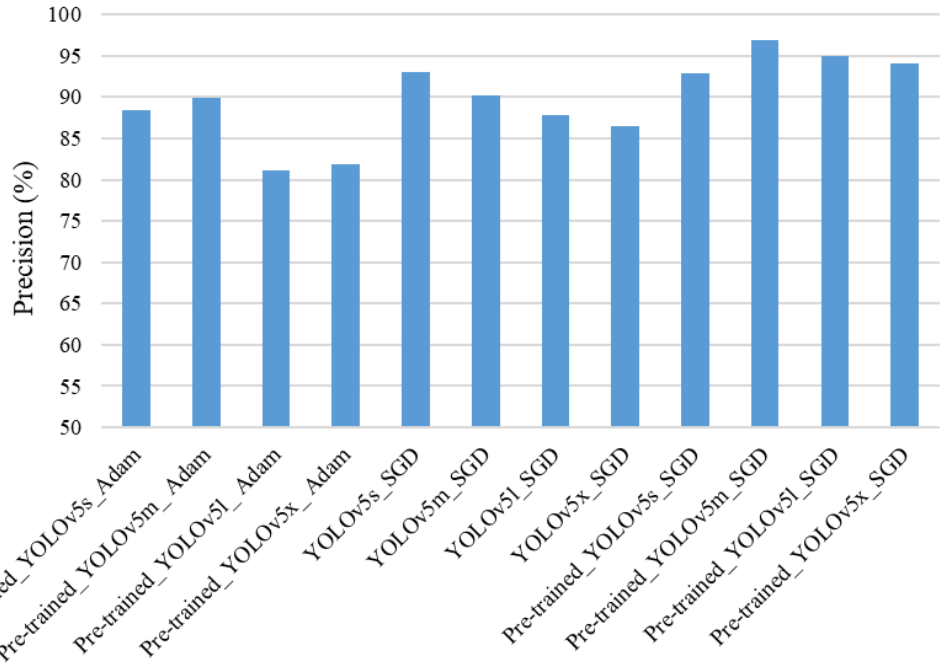

Fig. 4. Performances of difference YOLOv5 architectures for object detection.

REFERENCES

[1] Khoa Nguyen Toan, Minh-Quang Tran, Shu-Hao Liang, An Analysis of Object Detection Models on Duckietown Robot based on YOLOv5 Architectures. *2021 International Automatic Control Conference (CACS)*, 1121, Chiayi, Taiwan, 2021.

[2] Yeong DJ, Velasco-Hernandez G, Barry J, Walsh J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors*. 21(6), 2140, 2021. https://doi.org/10.3390/s21062140.

[3] Fadadu, S., Pandey, S., Hegde, D., Shi, Y., Chou, F. C., Djuric, N., Vallespi-Gonzalez, C. Multi-view fusion of sensor data for improved perception and prediction in autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2349-2357, 2022.

[4] J. Janai, F. Güney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles: Problems, datasets and state of the art, *Foundations and Trends® in Computer Graphics and Vision*, vol. 12, no. 1–3, pp. 1-308, 2020.

[5] Zhao, Z. Q., Zheng, P., Xu, S. T., Wu, X. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), pp. 3212-3232, 2019. https://doi.org/10.1109/TNNLS.2018.2876865.

[6] Chen, K., Li, J., Lin, W., See, J., Wang, J., Duan, L., ... Zou, J. Towards accurate one-stage object detection with ap-loss. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5119-5127, 2019.

[7] L. Du, R. Zhang, and X. Wang, Overview of two-stage object detection algorithms, *Journal of Physics: Conference Series*, vol. 1544, no. 1, p. 012033, 2020/05/01 2020, doi: 10.1088/1742-6596/1544/1/012033.

[8] Girshick, R., Donahue, J., Darrell, T., Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[9] Girshick, R. Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[10] Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, *39*(6), pp. 1137-1149, 2016. https://doi.org/10.1109/TPAMI.2016.2577031.

[11] Elsisi M, Tran M-Q, Mahmoud K, Lehtonen M, Darwish MMF. Deep Learning-Based Industry 4.0 and Internet of Things towards Effective Energy Management for Smart Buildings. *Sensors*. 2021; 21(4):1038. https://doi.org/10.3390/s21041038.

[12] Tan, L., Huangfu, T., Wu, L. *et al.* Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification. *BMC Med Inform Decis Mak*, 21**,** 324, 2021. https://doi.org/10.1186/s12911-021-01691-8.

[13] Srivastava, S., Divekar, A.V., Anilkumar, C. *et al.* Comparative analysis of deep learning image detection algorithms. *J Big Data* 8**,** 66 (2021). https://doi.org/10.1186/s40537-021-00434-w.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779-788.

[15] F. Jubayer, J. A. Soeb , Mitun K. Paul , Pranta Barua , S. Kayshar , M. Rahman, A. Islam. Mold Detection on Food Surfaces Using YOLOv5. *Preprints*, doi:10.20944/preprints202105.0679.v1.

[16] X. Wu, D. Sahoo, S. C. H. Hoi, Recent advances in deep learning for object detection, *Neurocomputing*, vol. 396, pp. 39-64, 2020/07/05/ 2020, doi: https://doi.org/10.1016/j.neucom.2020.01.085.

[17] "Duckietown gitbub of yolov5." https://github.com/duckietown/yolov5/tree/master/models.

[18] L. Paull *et al*., "Duckietown: An open, inexpensive and flexible platform for autonomy education and research," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1497-1504, doi: 10.1109/ICRA.2017.7989179.

[19] Zeiler, Matthew D., Graham W. Taylor, Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pp. 2018-2025. IEEE, 2011.

[20] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117-2125. 2017.

[21] Wang, Wenhai, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8440-8449. 2019.

[22] Balagourouchetty, Lakshmipriya, Jayanthi K. Pragatheeswaran, Biju Pottakkat, and G. Ramkumar. GoogLeNet-based ensemble FCNet classifier for focal liver lesion diagnosis. *IEEE journal of biomedical and health informatics* 24, no. 6 (2019): 1686-1694.

[23] Wang, Chien-Yao, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, I-Hau 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 workshops*, pp. 390-391. 2020.

[24] R. G. J. Wijnhoven, P. H. N. de With, Fast Training of Object Detection Using Stochastic Gradient Descent, *2010 20th International Conference on Pattern Recognition*, 2010, pp. 424-427, doi: 10.1109/ICPR.2010.112.

[25] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980, 2014.

[26] M. -Q. Tran, M. -K. Liu, Q. -V. Tran, T. -K. Nguyen, Effective Fault Diagnosis Based on Wavelet and Convolutional Attention Neural Network for Induction Motors, *IEEE Transactions on Instrumentation and Measurement*, 2021. doi: 10.1109/TIM.2021.3139706.

[27] Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S. C. H. Towards Theoretically Understanding Why Sgd Generalizes Better Than Adam in *Deep Learning. Advances in Neural Information Processing Systems*, 33, 2020.