Airy YOLOv5 for Disabled Sign Detection

Akhrorjon Akhmadjon Ugli Rakhmonov
Computer Science and Engineering Department
Kyungpook National University
Daegu, South Korea
r.akhror@knu.ac.kr

Barathi Subramanian
Computer Science and Engineering Department
Kyungpook National University
Daegu, South Korea
achu_samriti@yahoo.com

Taehun Kim
Dipvision
Daegu, South Korea
dipvision.ceo@gmail.com

Jeonghong Kim
Computer Science and Engineering Department
Kyungpook National University
Daegu, South Korea
jhk@knu.ac.kr

***Abstract* — Designated parking spaces for individuals with disabilities are only meant to be used by vehicles with proper handicapped signage. Real-time monitoring is necessary to ensure that only authorized vehicles are parked in these spaces and to prevent unauthorized vehicles from using them. First, this research proposes to replace the backbone of a baseline YOLOv5 model which has 9 blocks with 6 EfficientNet blocks with less parameters but still have a higher accuracy in detecting disabled signs among other signages on the windshield of cars. Second, to compensate for the loss of blocks we have included an attention mechanism before detection part in our architecture which allows us to focus on the important regions needed for the task. Additionally, we propose to use a better optimizer AdamW to prevent overfitting. Based on these improvements, we have created a new object detector named Airy YOLOv5. To evaluate the effectiveness of our proposed method, a dataset containing images of cars with disabled signage on their windshields will be gathered and labeled. Experiments using this dataset show that our model achieves a better F1 score of 0.67 with 5 percent less parameters compared to the baseline model**.

Keywords— depthwise separable convolution; disabled signage; small object detection; supervised learning;

I. INTRODUCTION

Object detection, especially for smaller objects, can be difficult because of the limited resolution and context information present in the image [1]. Smaller objects tend to have less significant details as they go through each layer of the convolutional backbone of object detectors. Many real-time object detection systems are constrained by the computational resources available, particularly when the processing takes place on the same device that captures the image. Handicap parking spaces in parking lots are typically located near the entrance of the building and offer the most convenient and direct access to it. These spaces are reserved for vehicles displaying a disabled person sign. In addition, individuals with disabilities who drive may be eligible for certain privileges such as free parking in city-owned lots and discounted rates in public parking lots, as specified by local regulations. Due to the convenient location of handicap parking spaces, some drivers illegally park in these spaces. To ensure that only authorized vehicles are parked in these spaces and to apply designated benefits for disabled drivers, accurate real-time detection and recognition is necessary.

Attempts have been made to enhance the detection of smaller objects in images [2], but many of these methods involve focusing on a specific region of the image [3], [4] or using two-stage detectors, which are more accurate but slower in terms of processing, making them less suitable for real-time applications. This is why many single-stage detectors have been developed for this type of applications [5]. Another solution would be to increase the resolution of the input image, but it would lead to a significant increase in processing time.

YOLOv5 is a widely used single-stage object detector [6] that is known for its performance, speed, and clear, flexible architecture. It can be easily modified and runs on a widely available platform. However, many attempts to optimize YOLOv5 mainly focus on adjusting specific parameters or augmenting the training set, without considering structural changes to the model to better adapt it for a specific use case. While YOLOv5 is a powerful tool, it is designed to be a general-purpose object detector and shows inferior accuracy when applied to detect small objects.

Baseline Yolov5 model uses C3 layer in the backbone which consists of 3x3 convolutional filters with a stride of 1. On the other hand, the EfficientNet [13] architecture was designed with a focus on model efficiency, which is why it uses depth-wise separable convolutions and optimized filter sizes in its layers. These design choices result in fewer parameters and computational costs, while still maintaining strong performance on various tasks.

In depth-wise separable convolutions, the convolution operation is split into two parts: a depth-wise convolution and a point-wise convolution. The depth-wise convolution applies filters to each channel separately, while the point-wise convolution combines the results from each channel into a single output feature map. This separation allows the network to learn spatial relationships and channel-wise relationships separately, which reduces the number of parameters needed.

Additionally, EfficientNet uses a scaling method to adjust the size and depth of the model to fit the desired computational budget and performance targets. This means that the network can be made larger or smaller depending on the needs of the task at hand, further reducing the number of parameters when necessary.

Moreover, attention layers are added to the model's architecture to weigh the importance of different regions in the input image during the feature extraction process. This allows the model to focus on the regions of the image that are most relevant to the task of object detection, and to ignore less important regions. Attention layers are usually implemented using a mechanism called self-attention which allows the model to learn to weigh the importance of different regions by looking at the relationship between them. This improves the model's ability to identify objects and their locations in the image, even when they are partially occluded or in a cluttered environment.

Considering the existing problem of detecting small objects in real-time scenarios and the limitations of existing models, this research proposes modifications to the YOLOv5-s object detector to improve its performance in detecting small objects. Specifically, in the application in detecting disabled sign among other sign on the windshields of the cars. We present a modified model that leverages the benefits of EfficientNet layers in the backbone and attention layers before detection layers of the model, about which discussed above, to perform the task more effectively while maintaining real-time processing speeds.

The main contributions of this paper are as follows.

1) This research proposes a modified architecture of the YOLOv5 model which can be utilized in real-time due to its light size. This is because of 6 EfficientNet blocks instead of C3 blocks in standard YOLOv5 architecture since they need less computational resources while maintaining competitive accuracy.

2) The proposed method employs attention layers before detection procedure which compensates the accuracy of the detection of small disabled signs.

3) The proposed model employs a better optimizer, AdamW, than Adam optimizer which is used in a baseline model in terms of generalizability. Consequently, the proposed model demonstrates less loss during validation time.

The organization of the rest of the paper is as follows. Section 2 explores some existing methods in object detection domain. Section 3 explains the methodology of our proposed approach. The experimental results are presented in Section 4. Finally, Section 5 offers conclusions and suggestions for future work.

II. Related Work

Object detectors can be divided into two categories: one-stage and two-stage detectors. Two-stage detectors, such as Fast R-CNN [14] break down the detection task into generating region proposals and classifying them. Afterwards, Faster R-CNN [8] was proposed to improve the accuracy of the detection. While there have been efforts to improve their ability to detect small objects, they often prioritize performance over inference time. Despite this, two-stage detectors have seen significant improvements through simplifying their structure and optimizing data flow. Even though these kinds of detectors showed high performance, the inference time is still inferior when it is needed to detect a particular object in real-time.

The single shot detector (SSD) [15] was designed without a region proposal stage and utilized a single shot approach. In contrast to Faster R-CNN, which only employed the final layer for detection, SSD utilized multiple layers for detection to enhance its ability to detect objects of varying scales. However, SSD performed poorly on small objects, due to shallow layers without deep semantic information.

YOLO is a widely used family of object detectors that have gained popularity in recent years. YOLOv1 [9] simplifies the process by treating object detection as a regression task, which makes it possible to build faster models that can operate in real time. However, the limitations of the model are low accuracy when it is used to detect small objects.

YOLOv2 [16] proposed a series of enhancements to the original YOLOv1 framework. To improve recall, it removed the fully connected layers and introduced the use of anchor boxes to predict bounding boxes. Additionally, unsupervised learning techniques were utilized to automatically determine the bounding box scales and ratios from the training data Also other proposed techniques such as batch normalization, high-resolution classification, and multi-scale training dramatically improved detection accuracy while keeping the fast speed. Nevertheless, the existence of multiple object classes in one cell caused low accuracy.

YOLOv3 [17] built upon the improvements made in YOLOv2 by implementing further enhancements. One notable change was the use of binary cross-entropy loss for class prediction, which was deemed more appropriate than softmax loss for handling cases where a single bounding box encompasses multiple classes. A new backbone network with ResNet module was proposed for improved speed and accuracy, especially on small object detection. But ResNet still possessed a lot number of learnable parameters which hindered the model to be effectively deployed in real-time.

YOLOv5 [10] was released shortly after YOLOv4 [11], but the two models are not directly related. The authors of YOLOv5 and YOLOv4 are not the same, and there has been some debate over whether YOLOv5 should be considered a successor to YOLOv4. YOLOv5 has similar performance to YOLOv4 and shares a similar design. One of the main differences is that YOLOv5 is implemented using the PyTorch framework, whereas YOLOv4 is based on the Darknet framework. This makes YOLOv5 more accessible and easier to use in a wider range of development environments. Furthermore, models in YOLOv5 are significantly smaller, faster to train and more usable in real-world applications. Fig. 1 illustrates the default structure of YOLOv5 model.

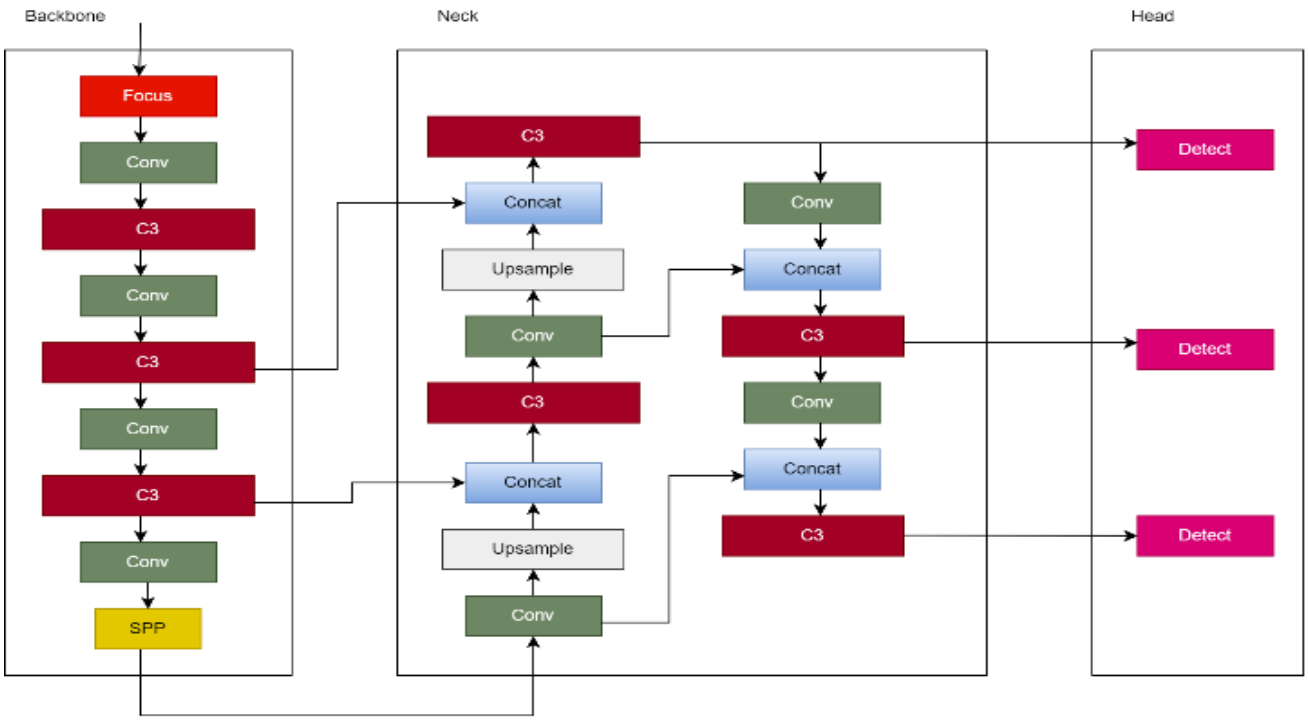

FIGURE 1: The default architecture of YOLOv5 model.

There have been attempts to create systems that focus processing on specific regions of the input image [3], which allows for better resolution and can overcome the limitation of having fewer pixels defining an object. However, this approach is better suited for systems that are not time-sensitive, as they require multiple passes through a network at different scales. This concept of focusing on certain scales can still influence the way certain feature maps are treated.

Moreover, there is a lot to be gained by analyzing how feature maps are handled, rather than just changing the backbone. Different types of feature pyramid networks (FPN) such as [12] aggregate feature maps in different ways to improve the backbone in various ways. Rakhmonov et al. [13] also showed the effectiveness of this technique in their research. However, inserting FPN layers to a model means increasing the number of learnable parameters. As a result, the speed of the model will suffer, and real-time applicability will become more difficult.

III. Proposed Methodology

This section thoroughly describes the proposed model. YOLOv5 offers four different scales for its model, namely S, M, L and X, which represents Small, Medium, Large, and Xlarge respectively. These scales apply varying multipliers to the depth and width of the model, maintaining the overall structure constant, but varying the size and complexity of each model. In the proposed method, since only small objects need to be detected, the structure of the S model is altered and compare the results with default Yolov5-S model results.

A. Data Pre-Processing

In the work, custom dataset is used which consist of images of automobiles with handicapped driver sign on the windshield along with other signages. Moreover, data augmentations are applied to increase the diversity of the dataset. These augmentations are RandomCrop, RandomGrayscale, RandomHorizontalFlip and GaussianBlur with the probability of p=0.3.

The annotation process of our dataset is done manually, we first crop the image to maintain focus on the sign. We use a tool called labelImg to draw a bounding box around the disabled sign and other signs and annotate them accordingly.

B. Data Learning Step

After pre-processing the dataset, the model is trained to solve the task of detecting disabled signages and drawing bounding boxes around them. The power of transfer learning approach is used to benefit from the pretrained model with significantly large dataset. In the backbone, 6 blocks of EfficientNet discussed in Section I trained on MS-COCO [18] dataset is used. Since it contains all important weights that are beneficial in detecting the disabled signs by the model.

C. Inference Step

During this step, the model is evaluated with unseen data to check the performance to detect the handicapped signage among other signs on the rear window of the vehicles.

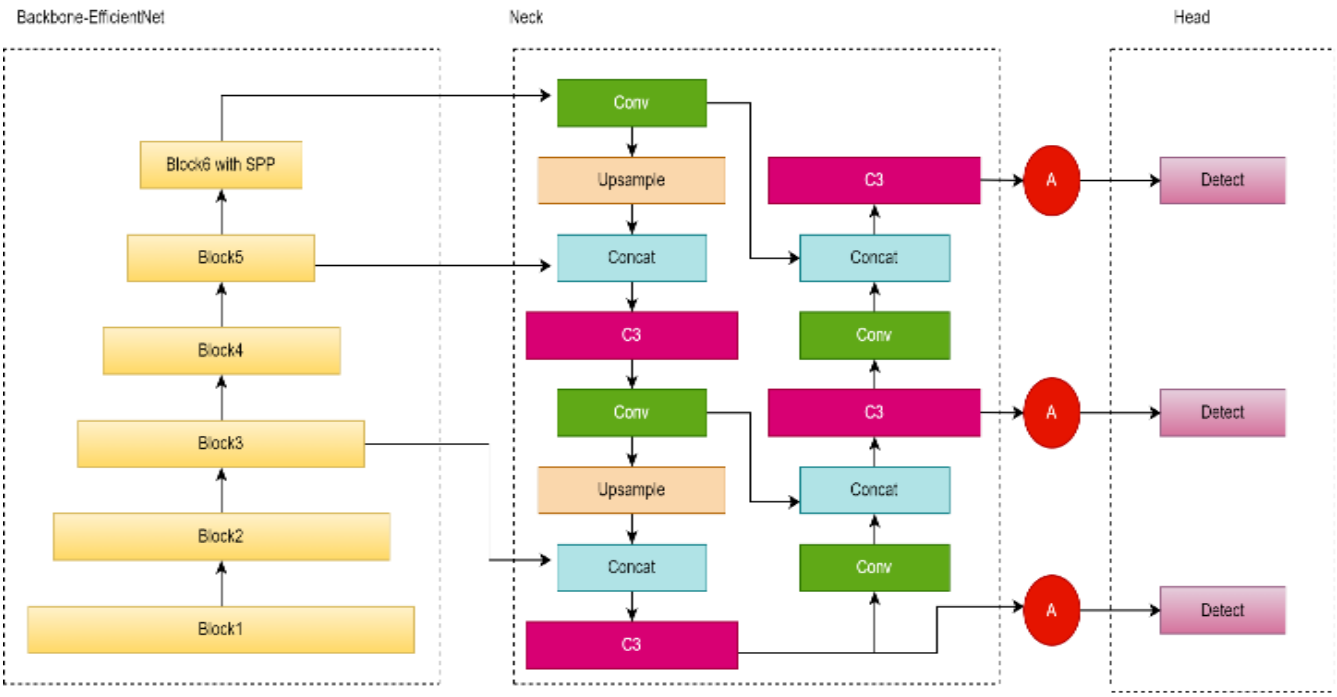

FIGURE 2: The architecture of the proposed model.

D. Proposed Model Architecture

The proposed architectural changes include using more efficient and less heavy in terms of computational operations a 6-block-EfficientNet in the backbone of baseline YOLOv5 model while preserving spatial pyramid pooling (SPP) layer in the last layer. Because by using an SPP layer, the model can maintain a constant feature map size, regardless of the size of the input, allowing it to make predictions with high accuracy even when objects are small or large in the input image.

Moreover, in order to maintain high accuracy while detecting tiny objects the proposed model also enjoys the presence of attention layer just before the detection process in the neck of the model since these kinds of layers were proven to make the model focus more on the regions where designated object is present. The overall architecture of the proposed model is shown in Fig. 2.

Additionally, we use a different optimizer than the original YOLOv5 model, which is AdamW which is a variant of the Adam optimization algorithm, which is an extension of stochastic gradient descent (SGD). AdamW includes weight decay, which regularizes the model by penalizing large weight values, which helps in preventing overfitting.

IV. E XPERIMENATAL RESULTS.

In this section, we represent comprehensive information about the conducted experiments and their results, as well as provide a comparison of the proposed method's experiment results with the ones of baseline model.

A. Datasets Description

The dataset is our custom dataset collected taking photos of cars with disabled signs on their front windows using mobile phones. This dataset has 1025 images with the dimensionality of 1920x1080 which we resize to 800x800. We split the dataset into two sets: train and validation data, which can be represented as 90% and 10% of the whole data accordingly.

B. Training Details

1) Experiment Settings: The proposed model was implemented using Python version 3.9.13 on a personal computer with 32GB of RAM and an Intel i5 2.90GHz CPU, running the 64-bit version of Windows 10.

2) Evaluation Metrics: The performance of the model is evaluated using loss metrics, which incorporates box, objective, and classification losses. Additionally, precision, recall and F1 score metrics are also used for a better illustration of the performance. The formulas for them are as follows:

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

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

$$F1 = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (3)$$

where TP is a true positive, TN is a true negative, FP is a false positive, and FN is a false negative.

C. Experiment Results

After only 20 epochs, our model for detecting disabled signs outperformed the baseline model in terms of precision, recall and F1 score, which gives us confidence that with more training it will likely show even better performance.

TABLE I. PERFORMANCE COMPARISON

Model	P	R	F1
Baseline YOLOv5-s	0.6	0.55	0.57
Ours	0.65	0.7	0.67

Table 1 illustrates that the precision (P), the recall (R) and the F1 score of the proposed model are higher by 0.05, 0.15, and 0.1, respectively, when compared to baseline model. The results of train and validation losses of our model and baseline model are illustrated in Fig. 3.

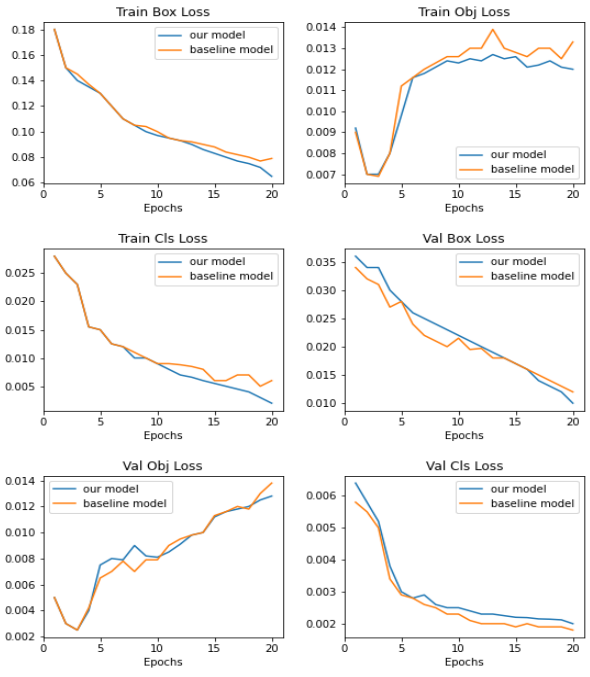

FIGURE 3: The performance of the models in terms of losses.

Fig. 3 shows that our model shows better results since the box loss and objective losses are less than those of baseline model both during the training time and validation time. While the classification loss during the training time is less than that of its counterpart but slightly more in the validation time. These results are convincing since the number of parameters are 5% less than the number of parameters in the baseline model.

V. Conclusion and Future Work

In this work, we proposed an airy YOLOv5 model for real time object detection using our custom dataset. We replaced the 9-block-backbone of the baseline model with 6 blocks of EfficientNet for the parameter reduction purpose and added attention mechanism before detection layers to better capture designated areas where the objects of interest are present.

Additionally, we employed a different optimizer than that of a baseline model for a better generalization and reduced overfitting. The contributions we made helped us to reduce the number of parameters while demonstrating better performance than baseline model results. Our model was able to outperform its counterpart in terms of precision, recall, F1 score and demonstrated less box loss and objective loss values. As a future work, we look forward to further optimizing the performance of Airy YOLOv5 model on other benchmark datasets such as COCO.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A3043970). This study was also supported by the BK21 FOUR project (AI-driven Convergence Software Education Research Program) funded by the Ministry of Education, School of Computer Science and Engineering, Kyungpook National University, Korea (4199990214394).

References

[1] G. Cao, X. Xie, W. Yang, Q. Liao, G. Shi and J. Wu, "Feature-fused SSD: Fast detection for small objects," in *Ninth International Conference on Graphic and Image Processing (ICGIP 2017)*, vol. 10615. SPIE, 2018, pp. 381-388.

[2] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le, ''An evaluation of deep learning methods for small object detection,'' *Journal of electrical and computer engineering*, vol. 2020, pp. 1–18, 2020.

[3] B. Singh, M. Najibi, A. Sharma, and L. S. Davis, ''Scale normalized image pyramids with autofocus for object detection,'' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3749– 3766, 2021.

[4] B. Singh, M. Najibi, and L. S. Davis, ''Sniper: Efficient multi-scale training,'' *Advances in neural information processing systems*, vol. 31, 2018.

[5] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, ''Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving,'' in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2017, pp. 129–137.

[6] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, ''Yolo-z: Improving small object detection in yolov5 for autonomous vehicles,'' *arXiv preprint arXiv:2112.11798,* 2021.

[7] S. Ren, K. He, R. Girshick, and J. Sun, ‘‘Faster r-cnn: Towards real-time object detection with region proposal networks,’’ *Advances in neural information processing systems*, vol. 28, 2015.

[8] G. Plastiras, C. Kyrkou, and T. Theocharides, ‘‘Efficient convnet-based object detection for unmanned aerial vehicles by selective tile processing,’’ in *Proceedings of the 12th International Conference on Distributed Smart Cameras*, 2018, pp. 1–6.

[9] J. B. G.Jocher, A.Stoken. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models. 2021.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, ‘‘Yolov4: Optimal speed and accuracy of object detection,’’ *arXiv preprint arXiv:2004.10934,* 2020.

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, ‘‘Feature pyramid networks for object detection,’’ in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[12] J. H. Kim. A. Rakhmonov, and B. Subramanian, “Disabled Sign Recognition Using Single Shot Detection FPN,” in *KICS 2022*, 2022.

[13] M. Tan and Q. Le, ‘‘Efficientnet: Rethinking model scaling for convolutional neural networks,’’ in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

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

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, ‘‘Ssd: Single shot multibox detector,’’ in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[16] J. Redmon and A. Farhadi, ‘‘Yolo9000: better, faster, stronger,’’ in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[17] J. Redmon and A. Farhadi, ‘‘Yolov3: An incremental improvement,’’ *arXiv preprint arXiv:1804.02767*, 2018.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, ‘‘Microsoft coco: Common objects in context,’’ in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, pp. 740–755.