YOLO for High Resolution Images without Retraining

Daisuke MINAMI and Kiyoshi NISHIKAWA
Dept. of EECS, Tokyo Metropolitan University
E-mail: minami-daisuke@ed.tmu.ac.jp, kiyoshi@tmu.ac.jp

***Abstract*—In this paper, we consider the application of YOLO (You Only Look Once) to high-resolution images, i.e., so-called 4K, or 8K images. When those large images are applied to YOLO, in general, the original image will be resized before being processed by YOLO. In this case, the detection accuracy varies according to the size of the resized image. As an example, when a 4K image is resized to 320×320 pixels, YOLO would fail to detect objects that make up less than about 0.7% to 2% of the entire image. Or, it would fail to detect large objects that occupy about 2% to 10%, or more of the entire image, when 4K image is used without resizing. These values are confirmed in this paper through our experiments. Then, we propose a novel YOLO construction method that enables us to expand the area of the detectable objects when applied to high-resolution images. The advantage of the proposed method is that it does not require the retraining using high-resolution images. Instead, the proposed method uses images of multiple resolutions and combines the detection results. The effectiveness of the proposed method is confirmed by the experiments.**

I. Introduction

In this paper, we consider detectable size of objects in YOLO (You Only Look Once) and propose a novel configuration of YOLO for expanding the size for improving the detection accuracy. When YOLO is used for detecting objects from high-resolution images, such as so called 4K (images with approximately 4000 pixels on one side) or 8K (images with approximately 8000 pixels on one side) ones, the effectiveness of the proposed method increases. The advantage of the proposed method is that it achieves the expansion based on the existing model and the improvement can be achieved without retraining of the model.

YOLO is one of the standard object detection models based on CNN (convolutional neural network). Object detection technology based on CNN has been utilized in a wide variety of applications including autonomous driving systems [1], robot vision [2], etc. Various models for object detection have been proposed so far, and they can be categorized into two types, i.e., two-stage and one-stage models, based on the network architectures. The two-stage type performs region proposal and image classification in separate processes, while the one-stage type performs these two processes simultaneously. R-CNN [3], Faster R-CNN [4] are typical two-stage type detection models, and, on the other hand, YOLO [5] and SSD (Single Shot MultiBox Detector) [6] are well known one-stage type models. Of those models, YOLO is known for its high speed processing and detection accuracy, and, currently, it is regarded as one of the standards of detection models.

Most of conventional studies on object detection have focused on improving detection accuracy and increasing the processing speed. Thanks to those researches, real-time detection from video sequences, or processing on generic PC without GPUs have been realized and utilized in a wide variety of applications. In particular, YOLO has seen active research, with many researchers proposing improved versions, making it a representative model that enables fast and accurate object detection.

In contrast, this study aims to improve object detection from a different perspective than those existing studies. Namely, we examin the size of objects that can be detected using YOLO. We consider the relation between the number of convolutional layers in standard YOLO configuration and the range of the size of detectable objects. It is shown that we can select the optimum number of convolutional layers according to the size of each object. Based on the consideration, we propose the novel structure of YOLO which uses multiple images which are generated by resizing with different ratio the original image for detection. The effectiveness of the proposed configuration increases when we use the high-resolution images, such as 4K or 8K, as the input to YOLO. Conversely, less improvement would be obtained by using the proposed method to the standard resolution images, such as about 1000×1000 pixels in size. In this case, almost all of the objects in the image are in the detectable range of the standard YOLO.

The motivation of this study arises from the recent proliferation of digital images, where the performance of imaging devices such as cameras has improved, making it easy to obtain high-resolution images. Hence, this paper focuses on the object detection from those high-resolution images where importance of consideration on the detectable range of sizes of objects is high, and, based on the consideration, we propose modification of the structure of YOLO to extend the number of detectable objects.

II. Background

Here, we describe the object detection using YOLO version 7, or YOLOv7, which is the 7th version of YOLO. Note that, in this paper, we mainly use the version 7 of YOLO, and hence, in the following, we use the term 'YOLO' to indicate its version 7.

A. *Effects of the size of the image on the detectable objects*

When a high-resolution image is applied to YOLO, it is observed that the detection accuracy for specific objects varies depending on the size of the image processed by YOLO. We show examples of the characteristics in Fig. 1.

The difference in the conditions to obtain these figures is in the size of the input image when applied to YOLO. The size of the original image is 4000×4000 pixels, and we applied the image to YOLO after resizing it. For left of Fig. 1, the image is resized to 320×320 pixels, and for right Fig. 1 to 4000×4000 pixels.

Other conditions are same for both figures. Namely, we used the pre-trained model of YOLO obtained from [7]. The detection thresholds were set with a classification confidence of 0.6 or higher. The detected objects are enclosed in red rectangular regions.

Let us examine the effect of the differences in size of the input image for YOLO. Focusing on the person at the front, it can be seen that the person is detected when resized to 320×320 pixels, but not when resized to 4000×4000 pixels. On the other hand, focusing on the motorcycles and people enclosed in the orange box at the back right, it can be observed that three motorcycles and three people are detected when resized to 4000×4000 pixels, whereas only one motorcycle and one person are detected when resized to 320×320 pixels.

Thus, when we use a high-resolution image, it is confirmed that the detectable objects using YOLO varies according to size of the input image. More precisely, the range of the size of an object that can be detected is determined by the size of the input image which is obtained by resizing the original image.

This characteristic may not be considered as a serious problem when the size of the original image is around 1000×1000 pixels. In contrast, as the size of the original image become larger, such as 4K or 8K, it would be the characteristic to be considered. Because, in that case, the detection accuracy depends on the choice of size of the input image which is obtained by resizing the original image. In YOLO, a user can specify the size of the image for processing independently from the size of the original image, and the original image is resized accordingly based on the specification before detection. When the original image is high-resolution one, as shown in the above examples, the detectable objects may vary depending on the selected size for processing. Hence, when considering object detection from high-resolution images, setting the appropriate size for processing becomes crucial because they may contain objects of a wider range of sizes in the image.

B. *Object detection from a image of single resolution*

Next, we confirm that there are relation between the number of convolutional layers applied and the size of objects, that can be detected by extracting features through these convolutions.

YOLO features an architecture where convolution is performed with channel partitioning known as ELAN[8], and upsampling is conducted midway through convolution. Because

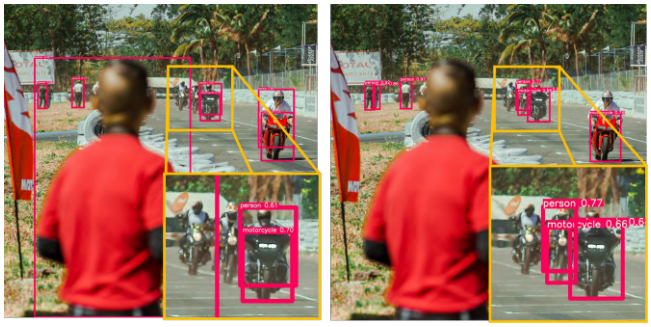

Fig. 1: Detection results using YOLO with input image sizes of 320×320 pixels (left) and 4000×4000 pixels (right).

of this processing, there is no straightforward expression to show the relationship between the number of convolutional layers and the detectable sizes of objects in YOLO. Instead, we show the relationship using the simpler structure of SSD300 in Table I. The relation was shown in [9]

We can say, as expected, for detecting the small objects, we need fewer convolution layers and conversely, more convolution layers are needed for large objects. These simple relations will be emphasized when a high-resolution image is applied to YOLO. Because YOLO has specific anchor box (AB) sizes prepared for each feature map used, it can only detect objects whose features match the size of the AB extracted by the specific feature map theoretically. Hence, our purpose in this paper is to propose a novel structure of YOLO that enables to detect both large and small objects from high-resolution images using existing YOLO model without retraining.

For the objects whose size is smaller than the specific limit, YOLO cannot extract enough features for detection. Conversely, the object whose size is larger than the specific limit, the number of convolutional layers used in YOLO is not enough for extracting features for those objects. These problems are not a problem for normal-sized images, but they are noticeable when high-resolution images are used as input.

Addition of convolutional layers may improve the detection ratio for large objects, however, this approach requires a significant amount of computational resources, and re-training of the network with the high-resolution images. More precise difinition of the size of the objects is given in the following consideration.

TABLE I: The relationship between the number of convolutional layers and the detectable sizes of objects in SSD300

The number of convolutions	10	15	17	19	21	23
Min size	30	60	111	162	213	264
Max size	60	111	162	213	264	315

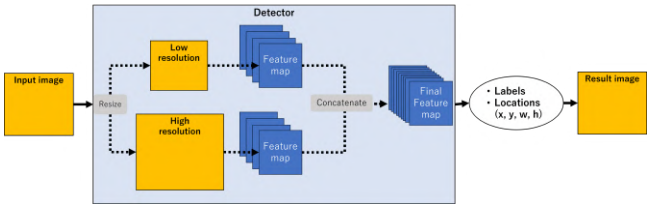

Fig. 2: The configuration of the conventional method.

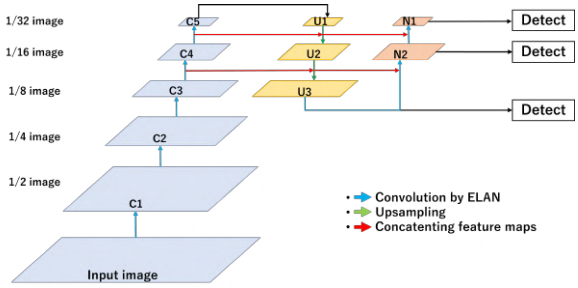

Fig. 3: Configuration of the standard YOLO.

III. Conventional Methods

A. Detection from Multi-Resolution Images

As described in the next section, the proposed method uses multi-resolution images, namely, two images with different resolutions as the inputs to the detection model. Here, we note that there are similar conventional methods which use multi-resolution images for object detection [10], [11].

In Fig. 2, we show a configuration of the method that involves resizing the image to multiple resolutions and performing convolutions in parallel. Features are extracted from multiple resolutions, and the extracted features are then combined for detection. This allows for detection across multiple resolutions[10], [11].

However, in contrast to the proposed method, it is difficult to apply them to the existing pre-trained object detection models such as YOLO or SSD. When we consider applying the methods to those pre-trained models, it is necessary to add a new layer to combine the feature maps and retrain the model. In doing so, the model becomes more complex, making optimization difficult, and training data of different resolution are also required.

B. Detection from High-Resolution Images Using SSD300

The authors[9] proposed a detection model based on SSD300, so called because it resizes the input image to 300×300 pixels, for high-resolution images without retraining. Specifically, the method enables detection of small objects (those occupying less than 1% of the image area) that were previously undetectable. The proposed method is an extension of the basic idea of [9] to YOLO for expanding the detectable range of the sizes of objects.

IV. Proposed Method

Here, we describe the proposed method. One of the advantages of the method is that it could achieves the improvement of detection accuracy using the standard YOLO model without retraining. The propose method enables the improvement of accuracy of the existing pre-trained models by merging detection results from multiple images with different resolutions.

A. Preparation

Since the proposed method uses the images of several resolutions, we first define the "size of objects" as the proportion of the object area in the original image before resizing. Because the area proportion depends on the size of the original image, this study evaluates the conventional and the proposed methods under the assumption that the size of the input image is fixed as 4000×4000 pixels for consistency. We note that this assumption is used only in the analysis, and there is no restriction on the size of the image when the proposed method is used in actual applications.

B. Configuration of YOLO

In Fig. 3 we show the configuration of standard YOLO version 7 used in this paper. During detection, YOLO performs convolution and upsampling to the input image to reduce or enlarge the size of the image. It then detects objects based on the multiple feature maps obtained by this process. YOLO uses feature maps with resolutions of 1/8, 1/16, and 1/32 of the size of the input for detection. To extract feature maps at each resolution, YOLO employs an ELAN structure which enables fast processing by distributing the number of channels during convolution.

Note that there are several models of YOLO version 7. In the following, we also use YOLOv7-e6e (yolov7-e6e.pt on [7]), the largest model of YOLO. YOLOv7-e6e uses feature maps with resolutions of 1/8, 1/16, 1/32, and 1/64 for detection. Additionally, YOLOv7-e6e employs an E-ELAN [12] structure to extract feature maps at each resolution.

C. Relation between anchor boxes and detectable range

As shown in Section II-B, the detection accuracy for specific objects varies depending on the size of the input image. Because, YOLO predetermines the sizes of the anchor boxes (ABs), which are rectangular regions used to determine the bounding regions of objects during detection.

Table II shows the sizes of ABs used in YOLO. Three types of ABs are used for each feature map as shown in Section IV-B. After resizing the image, the size of objects that can be detected corresponds to the size of these ABs. However, in YOLO, the optimal ABs are used based on the training data, so the ideal detectable object sizes may differ from the actual detectable object sizes.

D. Relation between detectable range and the size of input image in YOLO

We experimentally investigated the detectable range of YOLO for each resolution. The resolutions used must be divisible by the value of the max stride used in YOLO. Namely,

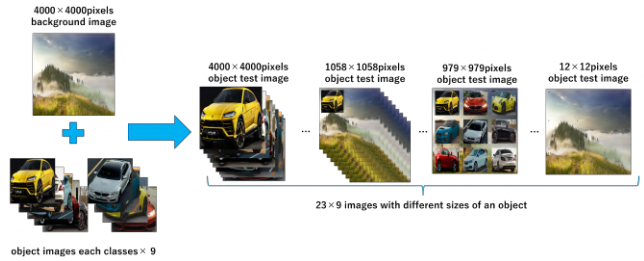

Fig. 4: Test images for investigating the detectable range of YOLO.

YOLO uses 32 as the max stride, and in YOLOv7-e6e uses 64. Therefore, in this study, we investigate the resolutions of 320, 960, 1600, 3200, and 4000 (4032).

Test images of four classes (Car, Cat, Horse, Sheep) are artificially generated as shown in Fig. 4, and, we evaluated whether each size of the object is detectable or not. The sizes of objects in the test images are ranging from 0.001% to 100% of the original image area.

Here, we regard the object as detectable if the following three conditions satisfy: (i) the predicted object class matches the ground truth label, (ii) the classification confidence is above 0.6, (iii) the IoU is above 0.75. Then, we calculate Precision and Recall, and the F-measure. If the F-measure is above 0.7, we regard that the object size is considered as detectable. Otherwise, it is classified as undetected.

The average values of these metrics for the four classes define the detectable range, as shown in Tables III and IV. In the tables, "Object size (%)" represents the percentage of the object's area in a 4000×4000 pixels background image, while "Object size (px)" denotes the actual number of pixels occupied by the object. Additionally, "Object size (After resize)" indicates the size of each object in pixels after the input image is resized to different resolutions, reflecting the sizes of ABs used by the pre-trained model. We used sizes 4000, 3200, 1600, and 960 for YOLO when calculating the average maximum value of Object size (After resizing). For YOLOv7-e6e, we used the sizes 4000, 3200, and 1600. Because, when considering that Object size (After resize) is the size of AB, unused sizes are presumed to be smaller than the maximum value of the size of AB.

Using these values, it is theoretically possible to determine the relation between the detectable object size and the resolution, and they are shown in equation (1). The equation indicates that within the original size of the image, the detectable object size becomes AB size after resizing to the Resize size.

TABLE II: Sizes of anchor boxes in YOLO

Scale of feature maps	YOLO	YOLOv7-e6e
1/8	12×16, 19×36, 40×28	19×27, 44×40, 38×94
1/16	36×75, 76×55, 72×146	96×68, 86×152, 180×137
1/32	142×110, 192×243, 459×401	140×301, 303×264, 238×542
1/64		436×615, 739×380, 925×792

$$\text{Detectable Size} = \frac{\text{Original Size} \times \text{AB Size}}{\text{Resize Size}} \quad (1)$$

E. The Flow of The Proposed Method

From the results shown in IV-D, it is seen that by using resolutions of 320×320 pixels and 4000×4000 pixels (4032×4032 pixels for YOLOv7-e6e), we can theoretically achieves the maximum detectable range.

Based on this, we propose a novel configuration of YOLO for high resolution images by mixing the results of detection using the two images of the resolutions above. Flow of the proposed method is shown in Fig. 5 using YOLO. First, the input hige resolution image is resized to 320×320 pixels and to 4000×4000 pixels. Then, detection is performed on each of the two. According to Table III, the detector resized to 320×320 pixels is expected to detect objects ranging in size from 0.695% to 100%. Similarly, the other detectoris expected to detect objects ranging in size from 0.01375% to 2%. Afterward, the detection results from each resolution, specifically the five values representing the detected object's class name and bounding box information (class_name, x_center, y_center, width, height), are integrated. This approach is expected to detect objects ranging in size from 0.01375% to 100%.

V. Evaluation by Simulations

We conducted simulations to evaluate the effectiveness of the proposed method. Six natural images are used as the original images, and they are 4000×4000 pixels in size and contain objects of various sizes. We compared the results on low-resolution images (320×320 pixels) with those on high-resolution images (4000×4000 pixels) and with the proposed method, and evaluated them.

In order to evaluate the proposed method in a straightforward manner, we define evaluation criteria in terms of the size of objects in the image. In this study, since the resolution of the input image is varied when performing the detection, we

TABLE III: Detectable range when YOLO is used.

Resize size (px)	Object size (%)	Object size (px)	Object size (After resize)
4000	0.01375 to 2	47 to 566	47 to 566
3200	0.01425 to 4	48 to 800	38 to 640
1600	0.375 to 17.5	77 to 1673	31 to 669
960	0.085 to 50	117 to 2828	28 to 679
320	0.695 to 100	333 to 4000	27 to 320
Average			**34.2 to 638.5**

TABLE IV: Detectable range when YOLOv7-e6e is used.

Resize size (px)	Object size (%)	Object size (px)	Object size (After resize)
4000	0.01275 to 10.75	45 to 1311	46 to 1322
3200	0.014 to 16.75	47 to 1637	38 to 1310
1600	0.06 to 67.75	98 to 3292	39 to 1317
960	0.1625 to 100	161 to 4000	39 to 960
320	1.935 to 100	556 to 4000	45 to 320
Average			**41.4 to 1316.3**

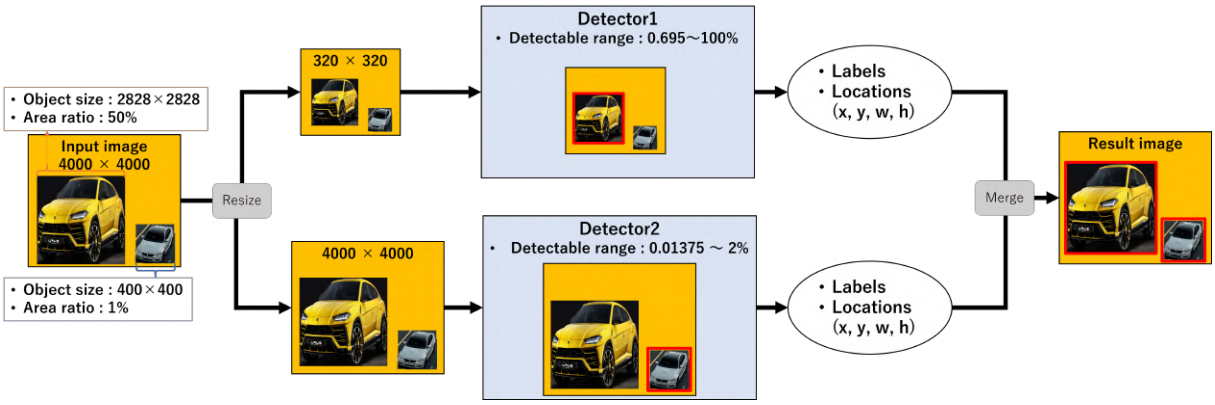

Fig. 5: Flow of the proposed method.

express the 'object size' as a ratio (%) of the area of the object in the original image, not in pixels.

The MS COCO [13] object size metrics are used to define objects. Three types of MS COCO metrics are available ('small', 'medium', and 'large'), but they depend on the size of the original image because they are defined by pixel size. Therefore, we define the ratio of the area of the object based on the geometric mean of width and height in coco test2017 (525 pixels). In order to evaluate the detection for high-resolution images, we also define in more detail the ratio of the area of objects smaller than 'small' objects ('micro', 'very tiny', 'tiny') in Table V[14], [15].

In the proposed method, detections were performed using resolutions of 320×320 pixels, 1600×1600 pixels, and 4000×4000 pixels. The detection results of the proposed method are shown in Fig. 6. Additionally, Tables VI and VII show the total number of detections and the percentage (%) of that across six natural images using low-resolution, high-resolution images and the proposed method with YOLO and YOLOv7-e6e.

We defined "detected" here using the ground truth we annotated. We evaluated the number of detections where the ground truth and the class of the detected object matched, with a confidence level greater than 0.6 and IoU greater than 0.5 and 0.75, respectively.

As mentioned in Section IV-D, it is clear that the size of detectable objects differs between low-resolution and high-resolution images. For YOLO, it is observed that the detection rate of objects larger than "very tiny" is higher for low-resolution images, whereas the detection rate of objects smaller than "micro" is higher for high-resolution images. Additionally, for YOLOv7-e6e, the detection rate of objects larger than "tiny" is higher for low-resolution images, while the detection rate of objects smaller than "tiny" is higher for high-resolution images.

In addition, the total detection rate from a single resolution is 52% to 68% at best, but in both trained models, the proposed method is able to detect objects of any size evenly, improving the detection rate to over 70% to 80%.

TABLE V: The definition of object size

Size name	Object size(%)
micro	0 to 0.38
very tiny	0.38 to 1.52
tiny	1.52 to 3.05
small	3.05 to 6.10
medium	6.10 to 18.29
large	18.29 to 100

TABLE VI: The number of detections by YOLO

Resize size	micro(28)	very tiny(24)	tiny(5)	small(2)	medium(2)	large(8)	total(69)
320 (IoU≥0.5)	5 (18%)	15 (63%)	4 (80%)	2 (100%)	2 (100%)	8 (100%)	36 (52%)
320 (IoU≥0.75)	5 (18%)	15 (63%)	4 (80%)	2 (100%)	2 (100%)	8 (100%)	36 (52%)
4000 (IoU≥0.5)	14 (50%)	10 (42%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	24 (35%)
4000 (IoU≥0.75)	14 (50%)	7 (29%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	21 (30%)
Proposed method (IoU≥0.5)	16 (57%)	24 (100%)	5 (100%)	2 (100%)	2 (100%)	8 (100%)	57 (83%)
Proposed method (IoU≥0.75)	14 (50%)	19 (79%)	5 (100%)	2 (100%)	2 (100%)	8 (100%)	50 (73%)

VI. Conclusions

In this paper, we proposed a method to expand the sizes of detectable objects using YOLOv7 on high-resolution images without retraining. In the proposed method, the original image, from which the objects are detected, is resized to different resolutions, and the detection results from multiple resolutions are merged to increase the number of detectable objects.

The effectiveness of the proposed method is demonstrated through simulations by comparing the results with those obtained using standard YOLO with a single resolution. We confirmed that the method improved the detection rate from 70% to over 80% and that objects of various sizes could be detected as expected.

TABLE VII: The number of detections by YOLOv7-e6e

Resize size	micro(28)	very tiny(24)	tiny(5)	small(2)	medium(2)	large(8)	Total(69)
320 (IoU≥0.5)	4 (14%)	9 (38%)	3 (60%)	2 (100%)	2 (100%)	8 (100%)	28 (41%)
320 (IoU≥0.75)	3 (11%)	6 (25%)	3 (60%)	2 (100%)	2 (100%)	8 (100%)	24 (35%)
4032 (IoU≥0.5)	17 (61%)	24 (100%)	3 (60%)	3 (100%)	0 (0%)	0 (0%)	47 (68%)
4032 (IoU≥0.75)	14 (50%)	20 (83%)	3 (60%)	2 (100%)	0 (0%)	0 (0%)	39 (57%)
Proposed method (IoU≥0.5)	18 (64%)	22 (92%)	5 (100%)	2 (100%)	2 (100%)	8 (100%)	57 (83%)
Proposed method (IoU≥0.75)	16 (57%)	17 (71%)	5 (100%)	2 (100%)	2 (100%)	8 (100%)	50 (73%)

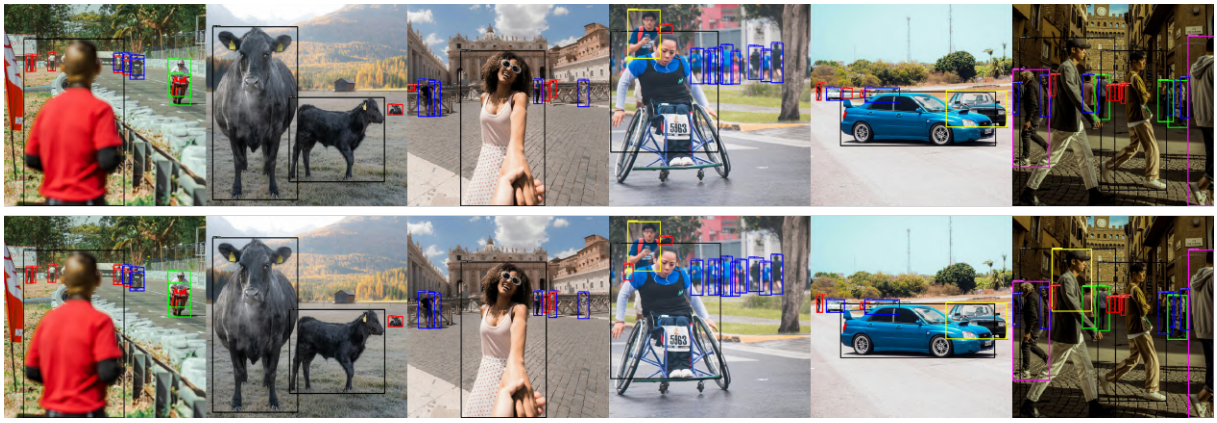

Fig. 6: Six images processed by YOLOv7 and YOLOv7-e6e using the proposed method. The top figure shows the objects detected by YOLOv7, while the bottom figure shows the objects detected by YOLOv7-e6e. The colors of the bounding boxes (BB) in the images are changed according to the sizes of the detected objects: (micro: red, very tiny: blue, tiny: green, small: yellow, medium: pink, large: black).

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number JP22K12170.

REFERENCES

[1] A. Balasubramaniam and S. Pasricha, *Object detection in autonomous vehicles: Status and open challenges*, 2022. arXiv: 2201.07706.

[2] S. Hoshino and K. Niimura, "Robot vision system for human detection and action recognition," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 24, no. 3, pp. 346–356, 2020.

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

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

[5] R. Joseph, D. Santosh, G. Ross, and F. Ali, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[6] L. W. et al, "SSD: single shot multibox detector," in *Computer Vision – ECCV 2016*, L. Bastian, M. Jiri, S. Nicu, and W. Max, Eds., 2016, pp. 21–37.

[7] *GITHUB: wongkinyiu/yolov7*. [Online]. Available: https://github.com/github-linguist/linguist.

[8] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, *Designing network design strategies through gradient path analysis*, 2022. arXiv: 2211.04800.

[9] I. Kei and N. Kiyoshi, "Detection method from 4k images using ssd300 without retraining," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp. 877–883.

[10] L. Guosheng, S. Chunhua, van den Hengel Anton, and R. Ian, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[11] F. Clement, C. Camille, N. Laurent, and L. Yann, "Learning hierarchical features for scene labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1915–1929, 2013.

[12] W. Chien-Yao, B. Alexey, and L. H.-Y. Mark, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 7464–7475.

[13] L. T.-Y. et al, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, F. David, P. Tomas, S. Bernt, and T. Tinne, Eds., 2014, pp. 740–755.

[14] K. Aleksandra, M. Karol, and B. Dominik, "Where to look for tiny objects? roi prediction for tiny object detection in high resolution images," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 721–726.

[15] W. Jinwang, Y. Wen, G. Haowen, Z. Ruixiang, and X. Gui-Song, "Tiny object detection in aerial images," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 3791–3798.