Efficient Scene Adaptation Technology for Object Detection Using Auto Image Synthesis

YAMAMOTO Yoshiki, HIRAI Sawa, HAMABASHIRI Hideto and OKAMOTO Yamato

Recently, because of the growing need for automation of monitoring, machine learning has become widely used in the social systems domain. However, the issue of machine learning is that the pre-trainedmodel provides poor performance when the environment changes. Especially in the social systems domain, there are various environmental changes such as place and camera angle, and additional learning through scene adaptation is essential to achieve the required accuracy.

Therefore, we tried to develop a method for more efficient scene adaptation. Until now, the cost to collect images of the new scene and label the ground truth was substantial. In this paper, we investigated a method to automatically generate images and pseudo ground-truth labels for training through image synthesis. The experiment was conducted for vehicle detection on highways and at intersections. As a result, we were able to generate a training dataset for less cost while achieving accuracy close to that of manually label . This method can be expected to further automation.

1. Introduction

In the domain of social systems, needs are mounting for automated monitoring to solve the current labor shortage. Automation needs are becoming particularly visible for such purposes as blind person's white stick detection, wheelchair detection, on-track fallen object detection, vehicle number authentication, traffic volume surveys, and wrong-direction traveling vehicle detection[1]. Monitoring is expected to involve various scenes and purposes. Hence, the required functions are also diverse, including object detection, behavioral tracking, and event detection. Assuming mainstream fixed cameras, we aim to provide object detection, the most central function of monitoring.

Object detection may be provided by having a model learn large amounts of data with labeled objects of interest for detection. Generally, machine learning-built models are, however, known to have the drawback of performing more poorly for data obtained in environments different from training data collection environments. This problem supposedly stems from data distribution differences due to environmental differences[2].The fixed camera-based object detection model discussed herein, for example, has the drawback of showing reduced detection accuracy when the shooting location or the camera angle conditions differ from those of the training data. In this case, the model would perform better after additional learning of training data prepared for each scene of its introduction. The problem is that manual labeling for an object detection model involves human checking of each object region in the images, followed by manual label of their coordinates, thereby incurring considerable costs. Especially in the domain of social systems, a diverse range of environmental differences is expected, such as shooting locations or camera angle conditions, thereby making scene-by-scene additional learning realistically impractical. Hence, a technique is required for adapting the model to scenes through efficient additional learning.

Contact : *YAMAMOTO Yoshiki* yoshiki.yamamoto@omron.com

2. Related studies

For efficient adaptation to scenes, a technique was proposed for generating training data from combinations of separately prepared foregrounds and backgrounds. T. Hodan et al. proposed a technique for pasting computer-generated 3D model objects onto various backgrounds in real image data while taking into consideration how such objects were exposed to light and how they overlapped one another[3]. G. Georgakis et al. proposed a technique for pasting foregrounds, in other words,

object images shot at various locations by cameras mounted on autonomous traveling robots, onto backgrounds, in other words, images shot by the robots in scenes to which to adapt the model while taking into consideration the depth information of the backgrounds[4].

These techniques are built on the implicit premise that foregrounds are obtained from scenes different from those to which to adapt the model. Moreover, because of the assumption that foregrounds and backgrounds differ from each other in scenes from which they are obtained, these techniques need high-volume computing to perform pasting without any pasting boundaries, shadows, or any other causes of visual awkwardness. For instance, T. Hodan et al.'s technique needs 120 seconds per image to generate a 640×480-pixel image and a ground-truth labels using a 400-node CPU cluster.

Hence, we propose a technique that assumes the acquisition of both backgrounds and foregrounds from identical scenes, in other words, ones to which to adapt the model, to generate images and ground-truth labels rapidly by following easy steps for specifying the coordinates and size for foreground pasting. This technique relies on neither manual labeling nor high-volume computing to enable training data generation, as well as additional learning by a model adapted to each specific scene.

3. About our proposed technique

Our proposed technique consists mainly of the following three processes. In the first process, a background and foregrounds are extracted respectively from each on-site image, to which the model should adapt, and are then stored in a database. In the second process, the background and foregrounds stored in the database are combined in various patterns to generate many synthetic images and pseudo ground-truth labels. In the third and final process, the generated synthetic images and pseudo ground-truth labels are used for additional learning of the object detection model. (See Fig. 1 for the processing flow.)

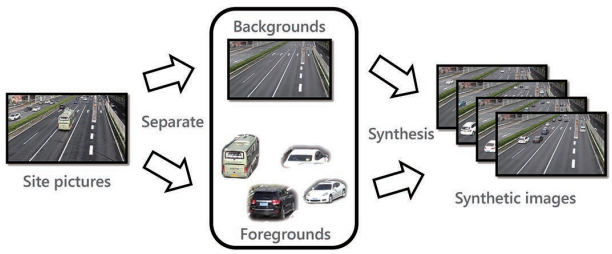


Fig. 1 Processing flow of our proposed technique

3.1 Object detection algorithm

For object detection techniques, a variety of different algorithms have been proposed, including Faster R-CNN[5], SSD[6], and YOLO[7]. Among these algorithms, SSD features lighter and faster processing compared with the others. In the domain of social systems, high frame rate image processing is required to detect motor vehicles traveling at high speed or pedestrians whose direction of movement can change irregularly. Besides, in situations with an underdeveloped communication environment or for apps for robots and other apps requiring high real-time responsiveness, a model is required to run lighter than conventionally to rely on limited computing resources, such as on-site edge terminals, for image processing without communication with servers. From the above perspectives, we adopted SSD, suitable for use in the domain of social systems, as the object detection algorithm to verify our proposed technique.

3.2 Background and foreground extraction method

Background and foreground extraction are possible through a variety of methods, including manual operations, the background subtraction method, and pre-trained machine learning models.

We chose to generate average images from video still images and use the resulting images as background images. We decided to perform foreground extraction using Mask-RCNN[8], a widely used method of instance segmentation, from the perspectives of efficiency improvement and accuracy. Mask-RCNN uses publicly available standard models not customized to any environment to extract portions determined as foreground regions. The boundaries between the foregrounds to be pasted and the background undergo transparency processing and smoothing for compatibility's sake.

4. Verification of the foreground pasting method

4.1. Experimental data and evaluation method

Assuming pre-existing security cameras, we conducted experiments on an object detection model for detecting vehicles captured in the video still images shot by a fixed camera installed overlooking an expressway. Incidentally, the images for use in our experiments were shot with prior permission by the competent road administrator and were used and controlled according to our in-house regulations. (Image control number: G190035−000)

A 30-minute-long video was shot at the spot in Fig. 2 for use as the source material for training data generation. Then, after the lapse of a certain time, another 30-minute-long video was shot. A total of 1,800 still images were obtained at one per second from the footage for use as the evaluation data.

Fig. 2 Video still image of the experimental site (expressway)

To obtain the background for synthesis, we generated 50 average images from the 1,800 still images obtained at one per second from the 30-minute video and stored the generated average images in a database. The reason that we don't use only one average image is that, we followed this procedure to avoid such risks as the foreground's inclusion in that one image. Because second-by-second foreground extraction results in repeated emergence of the same vehicles, we extracted foregrounds from a total of 450 still images, each obtained every 4 seconds from the video footage using Mask-RCNN and stored all these foregrounds in the database. Then, combining the background and foregrounds stored in the database, we generated 1,800 images and pseudo ground-truth labels, both of which were then learned trained by the model approximately 350 times, respectively, to evaluate its performance.

The metric used for the performance evaluation was Average Precision (AP). This indicator is employed in PASCAL VOC[9] or MS COCO[10] datasets widely used for object detection and takes a value between 0 and 1. The proximity of this value to 1 indicates that the degree of infrequency of misdetections and non-detections is very rare. In our experiments, with the ground-truth criterion threshold called IoU being fixed at 0.5, we used an AP of 0.5 calculated by 101-point interpolation AP. The AP of 0.5 is the average precision factor when the recall factor takes in the range of 0, 0.01, ⋯, and 0.99, 1.0.

$$AP_{0.5} = \frac{1}{101}(Pre.(0) + Pre.(0.01) + \cdots + Pre.(1.00))$$

Although shot to capture the region of interest for object detection, the road images in our experiments still contained unintended area. For example, our original intention for this time was to perform performance evaluation using the expressway area in the middle of Fig. 2. The image, however, contains ordinary roads on both sides. Our experiments excluded such unintended regions from the range of object detection to limit the region for the AP calculation. (The calculation result values are indicated as AP (Mask) in the table below.)

4.2 Constraints on foreground pasting

Generally, training data for machine learning is desired to show a distribution similar to that of the data actually observed at the site into which to introduce the model. With training data with less variation than on-site observed data, the model would fail to detect correctly unlearned patterns as they occurred. On the other hand, if the training data contains data unobservable on-site in reality, such as flying wheeled vehicles or gigantic wheeled vehicles too large to exist in reality, misdetections will increase, resulting in reduced performance. Based on the above, we incorporated our on-site findings into the following three constraints on image generation by our proposed technique to ensure that the generated synthetic image would be an accurate reflection of the distribution of the data actually observed on-site:

Constraint 1: Limit the vehicle size and the paste area.
Constraint 2: The average number of foregrounds per image must be the same between the synthetic image and the on-site data.
Constraint 3: Overlapping vehicles must occur at a certain probability.

Table 1 summarizes the results of the comprehensive verification of the effectiveness of each constraint. Note that Experiments 1−4, 2−3, and 3−1 were performed under the same conditions.

Each ✓ indicates the application of a relevant constraint, while each × means the non-application of a relevant constraint.

Table 1 Constraints on pasting and the precision of the synthetic image learning model

Expt. No.	Resizing constraint	Paste-area constraint	Average number of vehicles pasted (range)	Ave. No. of overlapping vehicles	AP (Mask) %
1−1	×	×	10 (1−19)	0	65.82
1−2	×	✓	10 (1−19)	0	66.37
1−3	✓	×	10 (1−19)	0	67.86
1−4	✓	✓	10 (1−19)	0	69.60
2−1	✓	✓	1 (1−1)	0	58.90
2−2	✓	✓	5 (1−9)	0	65.87
2−3	✓	✓	10 (1−19)	0	69.60
2−4	✓	✓	20 (11−29)	0	68.58
3−1	✓	✓	10 (1−19)	0%	69.60
3−2	✓	✓	10 (1−19)	20%	85.40
3−3	✓	✓	10 (1−19)	40%	87.52
3−4	✓	✓	10 (1−19)	60%	89.24

Each of the following sections presents a hypothesis on each of the three constraints and their verification results.

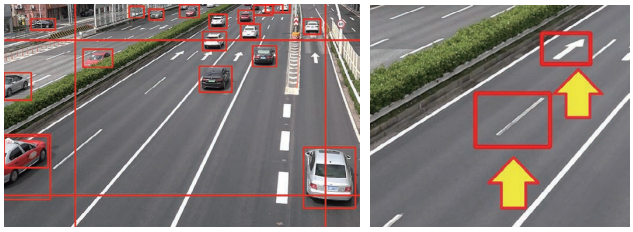
4.3 Paste-area/resizing constraints [Experiment 1]

This section considers the paste-area and resizing constraints for pasting foregrounds onto backgrounds. When foregrounds are pasted onto the same location as when extracted, the resulting synthetic image will contain less variation. Hence, foregrounds are also pasted onto different locations than when extracted to generate synthetic images of various patterns. Note here that scene-specific constraints exist on the points of vehicle emergence because no vehicle emerges at the place of a wall or a median strip in a real video shooting site. Because of the principle of SSD, an SSD model tends to learn likely points of foreground occurrence. Therefore, paste-area constraints, such as limiting pasting only to a road surface, are expected to work effectively.

Note also that a camera image shows nearside vehicles larger and far-side vehicles smaller according to the law of perspective. Therefore, the size ratio must be calculated when pasting a foreground at different coordinates than when extracted. A foreground to be pasted on a near side should be scaled up, while one to be pasted on a far side should be scaled down. Then, the resulting synthetic image can be expected to become closer to the actual observed image. Note, however, that excessive scaling up/down will result in a distorted foreground image. Hence, the foreground must be pasted at a point where it can remain within the range between an upper and a lower limit set for the enlargement and reduction ratios.

To verify the effectiveness of these constraints, we comparatively examined four patterns for with and without the resizing and paste-area constraints. With no coordinate constraint applied, foregrounds were pasted randomly. Meanwhile, with a coordinate constraint applied, foregrounds were pasted only at coordinates within a pre-specified road area. With the resizing constraint not applied, foregrounds pasted at points different than when extracted were not resized. With the resizing constraint applied, foregrounds were resized and pasted only within a range where they would fall within the scaling ratio range of 75% to 133%.

The experimental results show that the pre-trained detection accuracy was highest when both constraints were employed. As shown by the results of Experiment 1−1 (Fig. 3(a)), without the resizing constraint, detection frames may appear all over the screen. This problem occurs when the model learns the occurrence of an oversized vehicle that results from a foreground pasted without consideration of the change in size. In contrast, Experiment 1−4 (Fig. 3(b)) shows the model's tendency to misdetect white lines.



(a) Experiment 1−1 (b) Experiment 1−4

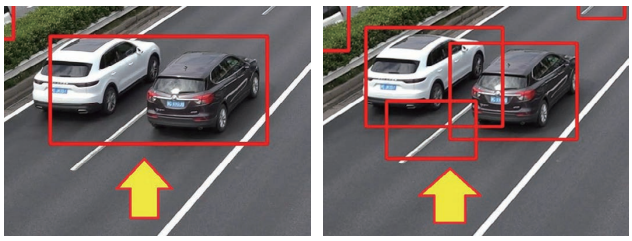
Fig. 3 Results of Experiment 1

4.4 The number of vehicles to be pasted [Experiment 2]

Some images may contain only one vehicle, while some other images may show multiple vehicles captured simultaneously. With a disproportionately large number of synthetic images that contain an extremely large or small number of foregrounds, inconsistencies with on-site data may result. Therefore, by limiting the average number of foregrounds to be pasted on an image to that of foregrounds in an on-site image, synthetic images can be expected to resemble actual observed images. To ascertain the effectiveness of adjusting the number of foregrounds, we compared a total of the following four patterns regarding pre-trained performance: 10 vehicles per image, which is the average number of vehicles in the experimental data, one vehicle per image as an example of an undersized number of vehicles, five vehicles per image, and 20 vehicles per image as an example of an oversized number of vehicles. Note that the resizing and paste-area constraints, both confirmed by Experiment 1 to be effective, were adopted for all four patterns.

The experimental results revealed that the model showed extremely low after additional learning performance for the case with one vehicle while showing the highest performance for the case with ten vehicles. The model tends to detect mutually proximate multiple vehicles collectively as one vehicle when the number of vehicles to be pasted on single images is one.

On the other hand, when that number is 20, the model tends to exhibit a propensity for over-detection, which may take the form of misdetection of the background or multiple detection frames applied around one vehicle.



(a) Experiment 2−1 (b) Experiment 2−4

Fig. 4 Results of Experiment 2

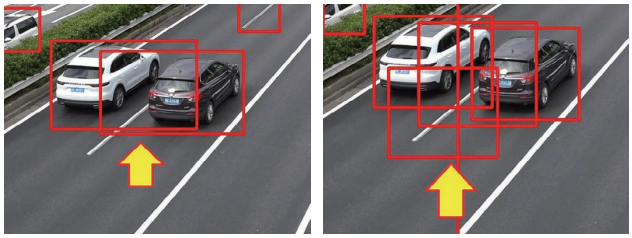
4.5 Generation of overlapping vehicles [Experiment 3]

One of the problems in object detection is the non-detection of objects with a visually inaccessible hidden portion. Assume, for example, that two objects overlap one another., the inner object, only partially observable, may be undetectable or collectively detected as a single object. Hence, one solution to this problem posed by hidden portions is to generate overlapping vehicles on purpose before synthesis to allow the subsequent learning by the model. This solution can be expected to improve the robustness of the model to partially hidden vehicles. The generation principle for overlapping vehicles is to specify coordinates while scaling up/down according to the principles presented in Section 4.3 and then perform the paste-vehicles operation for pasting a fixed number of vehicles to coordinates overlapping with other vehicles. It must be ensured that the vehicle closest to the camera will partially hide the vehicles not as close to it (Fig. 5).

Fig. 5 Vehicles overlapped by scaling up/down and repositioning

To ascertain the effectiveness of generating overlapping vehicles, we set four different proportions of overlapping pasted vehicles as 0%, 20%, 40%, and 60% to compare the four cases regarding the model's pre-trained performance. Note that the number of vehicles to paste on an image was set here to 10, which was the average number that had led to the model's highest performance as shown by the results in Section 4.4.

The experimental results showed that the model achieved the highest pre-trained model's performance for the case in which the overlapping pasted vehicles accounted for 60 percent of the total, followed by the case where the proportion to the total number of pasted vehicles was set to 40%. A look at the detection results showed that with an excessively small number of overlapping vehicles, the model tended to show reduced detection accuracy for overlapping vehicles while successfully detecting non-overlapping vehicles. By contrast, with an excessively large number of overlapping vehicles, the model applied multiple detection frames to non-overlapping vehicles and tended to show reduced detection accuracy for non-overlapping vehicles despite the successful detection of overlapping vehicles.



(a) Experiment 3-1 (b) Experiment 3-4

Fig. 6 Results of Experiment 3

Although performing best for the 60% case, the model took a considerable time to search pairs of overlapping pasted foregrounds and needed a longer search time with an increasing number of overlapping foregrounds. The average time that the model needed to generate an image with no overlaps was 2.4 seconds, significantly differing from the 6.4 seconds for the 40% case and the 14.7 seconds for the 60% case. The model did not perform very differently for the 40% and 60% cases but showed a difference of as large as 2.3 times in generation time between the two cases.

5. Performance evaluation

5.1 Setting the performance evaluation conditions

This section compares the following four methods to ascertain the effectiveness of synthetic image-based learning: (1) additional learning with manually labeled ground-truth, (2) additional learning based on images/ground-truth labels synthetically generated after a human check of the source foreground (our proposed technique: with source foreground check), (3) additional learning based on images/ground-truth labels synthetically generated without a human check of the source foreground (our proposed technique: without source foreground check), and (4)without additional learning.

In this experiment, we used video still images of two scenes (Scene 1=expressway and Scene 2=crossroads). Scene-1 expressway is the same location as in Chapter 4. Fig. 7 shows Scene-2 crossroads.

Fig. 7 Video still image of the experimental site (crossroads)

The object detection model and the training procedure used were the same as in Chapter 4. When pasting foreground based on our proposed technique, we apply the principle found most effective from the experimental results in Chapter 4.: the coordinate constraint was imposed as the principle to perform resizing in pasting to generate overlapping vehicles with the average number of simultaneously occurring foregrounds in agreement with that of the on-site data. Note, however, that the proportion of overlapping vehicles to the total number of vehicles to be generated was set to 40 percent from out of consideration of the generation time.

Note also that Mask-RCNN may extract white lines or arrows as the source foregrounds by mistake when extracting vehicles. For example, a vehicle's edge or a patch of pavement extracted by Mask-RCNN and learned by a model as a vehicle, as in Fig. 8, would reduce the model's performance. To investigate the impact of wrongly learned foregrounds which are extracted by Mask-RCNN, we compare two methods: one with defective source foreground s removed by human checking(with source foreground check) and the other without (without source foreground check).

Fig. 8 Examples of unintentionally detected non-vehicle objects

5.2 Results and discussion

The experimental results for the two scenes are shown in Tables 2 and 3, respectively. In either case, the model performed significantly better with synthetic image-based learning than without additional learning, substantiating the effectiveness of synthetic image-based learning. More specifically, the model achieved improvement of 27 points, while manual label achieved 35 points. In other words, in the case of expressway image synthesis with the source foreground check, the model achieved performance improvement equivalent to 77 percent of that previously achieved manually at an operation cost equivalent to 5.5 percent of the conventional level.

Table 2 Experimental results(expressway)

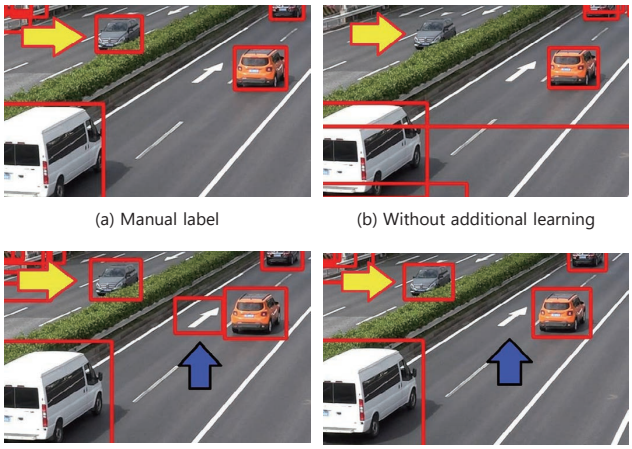
Technique	AP (Mask) %	Required person-hours (hours)
Manual label	97.52	144
Image synthesis (with source foreground check)	89.72	8
Image synthesis (w/o source foreground check)	89.56	4
Without additional learning	62.46	0

Table 3 Experimental results(crossroads)

Technique	AP (Mask) %	Required person-hours (hours)
Manual label	96.49	144
Image synthesis (with source foreground check)	85.50	8
Image synthesis (w/o source foreground check)	84.95	4
Without additional learning	80.90	0

On the other hand, for the crossroads, the model only managed to achieve improvement of 5 points, while manual label achieved 16 points.

Detection results shows that after synthetic image learning, the model detected patterns conventionally undetectable without additional learning. Let us take the cases in Fig. 9, for instance. Without additional learning, the model failed to detect the vehicle which is partially hidden behind the median strip. Meanwhile, with additional learning through synthetic images, the model successfully detected the same vehicle (yellow arrow). Although the arrow-marked portion of the pavement was misdetected as a vehicle without source foreground check, it was not misdetected after being removed after the source foreground check (blue arrow).



(a) Manual label (b) Without additional learning

(c) synthetic images (w/o source foreground check) (d) synthetic images (with source foreground check)

Fig. 9 Comparison of the detection results

When with additional learning through synthetic images, the model failed to reach the performance level achievable with the learning of manually labeled input data but fared better in terms of cost. For example, it took a cumulative total of 144 person-hours to add ground-truth labels manually to 1,800 images used in our experiments. By contrast, in the case of image synthesis, which only required parameter adjustment while checking each generated image, images and ground-truth labels were successfully generated in about four person-hours without source foreground checks and about eight person-hours with source foreground checks.

Besides, our proposed technique led to a significant reduction in computing time compared to those proposed in preceding studies discussing synthetic images. Whereas T. Hodan et al.'s technique needed 120 seconds per image to generate a 640×480-pixel image, our proposed technique successfully generated a 1,600×900-pixel image at the rate of 6.4 seconds per image.

One of the remaining challenges is to study the pasting method for obtaining synthetic images closer to actual video still images while maintaining the current low operation cost. Even with partial foregrounds or non-vehicle objects removed, our current pasting method fell short of the accuracy achievable with manually labeled input data. This problem occurred probably because of the difference between the image distribution achieved with automatic pasting and the actual image distribution. We consider it necessary to smooth shadows resulting from pasting or review the method of generating overlapping vehicles.

6. Conclusions

In this study, we studied, developed, and verified the effectiveness of a technique that could generate images and ground-truth labels at high speed using an automated tool and perform additional learning for vehicle detection tasks.

More specifically, we separated the vehicles and background in each video still image shot on-site and pasted the vehicles onto the background in various ways to allow automatic generation of images and ground-truth labels. Through the elaboration of the pasting method, we narrowed the distribution gap between automatically synthesized images and real images as much as possible to reduce non-detections and misdetections. As a result, the pasting method improved accuracy, with only 5.5% of the person-hours required with manually labeled ground-truth, thereby allowing the model to adapt to scenes.

The future challenges include how to reduce the performance gap relative to manually labeled ground-truthwhile maintaining the current low operation cost. This gap occurs probably because objects still look different between when they are captured in actual images and when they are pasted onto synthetic images. From now on, we intend to explore and continuously improve a natural pasting method for this challenge and deploy resulting technical achievements not only to fixed camera vehicle detection but also to various fields such as human and animal detection or mobile cameras.

References

1) Public Service Corporation Japan Security Systems Association, "Investigative Research Report on Image Analysis," (in Japanese), https://www.ssaj.or.jp/jssa/pdf/gazou_kaiseki.pdf (accessed: May 13, 2020).
2) A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proc. 24th IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR), 2011, pp. 1521−1528.
3) T. Hodan, V. Vineet, R. Gal, E. Shalev, J. Hanzelka, T. Connell, P. Urbina, S. N. Sinha and B. K. Guenter, "Photorealistic image synthesis for object instance detection," arXiv preprint arXiv: 1902.03334, 2019 (accessed: May 13, 2020).
4) G. Georgakis, A. Mousavian, A. C. Berg and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," arXiv preprint arXiv: 1702.07836, 2017 (accessed: May 13, 2020).
5) S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Proc. 30th IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, vol. 39, no. 6, pp. 1137−1149.
6) W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Y. Fu, et al., "SSD: Single Shot MultiBox Detector," *Computer Vision – ECCV 2016, Lecture Notes in Computer Science*, vol. 9905, Cham, Switerland: Springer, 2016, pp. 21−37.
7) J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. 29th IEEE Conf. on Computer Vision and Pattern Recognition* (CVPR), Las Vegas, NV, 2016, pp. 779−788.
8) K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *IEEE Int. Conf. on Computer Vision* (ICCV), 2017, pp. 2980−2988.
9) M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303−338, 2010.
10) T. Y. Lin *et al*., "Microsoft COCO: Common Objects in Context," *Computer Vision – ECCV 2014, Lecture Notes in Computer Science,* vol. 8693, Cham, Switzerland: Springer, 2015, pp. 740−755.

About the Authors

YAMAMOTO Yoshiki
Advanced Technology Laboratory
Technology Creation Center.
OMRON SOCIAL SOLUTIONS Co., Ltd.
Specialty: Mathematics (Differential Geometry)

HIRAI Sawa
Advanced Technology Laboratory
Technology Creation Center.
OMRON SOCIAL SOLUTIONS Co., Ltd.
Specialty: Information Engineering

HAMABASHIRI Hideto
Advanced Technology Laboratory
Technology Creation Center.
OMRON SOCIAL SOLUTIONS Co., Ltd.
Specialty: Image Processing
Affiliated Academic Society: IEICE

OKAMOTO Yamato
Advanced Technology Laboratory
Technology Creation Center.
OMRON SOCIAL SOLUTIONS Co., Ltd.
Specialty: Intelligence Science and Technology
Affiliated Academic Society: JSAI