Multi-scale Subnetwork for RoI Pooling for Instance Segmentation

Tran Duy Linh and Masayuki Arai

***Abstract*—Instance segmentation is a challenging task in computer vision because object locations in an image must be predicted and segmentation must be performed inside these locations. In the present paper, we propose a new pooling module to extract a small feature map from each Region of Interest for pixel-level prediction. Instead of using RoiAlign pooling, we use a small network module and ensemble the extracted multi-scale features in a feature map. The proposed method can output a better feature map and therefore better pixel-to-pixel alignment between input and output. The results of an experiment reveal that the proposed method outperforms cutting-edge instance segmentation methods.**

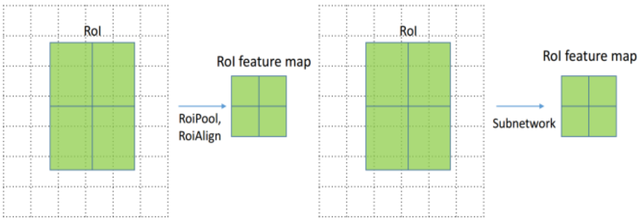
***Index Terms*—Deep learning, instance segmentation, RoI pooling module.**

I. Introduction

Object detection is one of the most fundamental tasks in computer vision. The performance of object detection has been greatly improved in recent years by the use of convolutional neural networks (CNNs). Detecting the spatial location of an object in an image involves either object detection by a bounding box or object detection by pixel-level segmentation. Object detection by pixel-level segmentation (instance segmentation) is the more challenging task because the detector outputs not only the object location but also the per-pixel classification, which is usually represented as a segmentation mask. In the present paper, we intend to realize object detection by pixel-level segmentation using a newly designed model.

In order to segment an object, some methods (e.g., FCN [1]) predict the segmentation masks and the class of an object simultaneously by generating the per-pixel multi-class categorization. However, with respect to the Mask R-CNN method [2], in the instance segmentation task, it is better to separate the class prediction and the segmentation mask prediction. Thus, we adopt the Mask R-CNN approach to add a mask branch to predict a binary segmentation mask for each class along with a box and a class prediction branch. The segmentation mask branch is an extension of the Faster R-CNN [3] method, which is jointly trained with other branches.

Current state-of-the-art detection networks [2]-[4] use a two-stage design for networks. These methods use a pre-computation method (e.g., SelectiveSearch [5]) or deep network to extract the set of object region proposals. These proposal boxes are used to extract features within the boxes (called Region of Interest (RoI) feature maps) by an RoI pooling layer and are then fed to the next stage (CNN). Fig. 1 shows the concept of using an RoI pooling layer to output fixed-size feature maps. The choice of the RoI pooling layer depends on the detection task. For example, the RoiPool [4] layer is widely used in object detection (predicting the boxes and classes). RoIAlign [2] is an improved version of RoiPool and is used for the pixel-prediction task. The quality of detection depends on the quality of the pooled RoI feature maps. A natural question to ask is whether a subnetwork module, which replaces the RoI pooling layer, achieves a similar quality. Fig. 1b shows the concept of the present study, which is based on the Faster R-CNN [3] method and replaces the pre-computed RoI with a deep network, called the region proposal network (RPN). However, in the instance segmentation task, the subnetwork must ensure pixel-to-pixel alignment between the RoI and the extracted features. The proposed subnetwork can be considered as a trainable version of the RoI pooling layer. We describe the proposed subnetwork in detail in Subsection III.



(a) RoiPool and RoIAlign (b) Proposed RoI pooling module

Fig. 1. RoI features extraction. (a) Fixed-size RoI feature maps are extracted from RoI of arbitrary size though an RoI pooling layer (e.g., RoiPool, RoiAlign). (b) Proposed method using a subnetwork to replace the RoI pooling layer.

The main contributions of the present paper are summarized as follows:

1) We introduce a new multi-scale RoI feature extraction module, which replaces the RoI pooling layer in the deep network with the instance segmentation task. The new subnetwork design allows us to train a high-quality RoI feature extractor for pixel-level prediction.
2) We perform experiments on a large-scale image dataset (COCO [6]) and show that the results are comparable to those of state-of-the-art instance segmentation methods, proving the effectiveness of the proposed approach.
3) We also analyze the effect of the proposed subnetwork for different pre-trained feature extractors (the baseline) and discus use cases.

The remainder of the present paper is organized as follows. In Section II, we review related methods that use a deep network for the segmentation task. Section III introduces the design of the proposed subnetwork and its implementation in

Manuscript received September 13, 2018; revised November 20, 2018.

Tran Duy Linh and Masayuki Arai are with Graduate School of Science and Engineering, Teikyo University, 1-1 Toyosatodai, Utsunomiya, Tochigi, Japan (email: arai@ics.teikyo-u.ac.jp).

DOI: 10.7763/IJCTE.2018.V10.1227

detail. The experimental results are reported in Section IV, and a discussion is presented in Section V. Finally, Section VI presents our conclusions.

II. Related Research

A. Faster R-CNN and Mask R-CNN

Faster R-CNN [3] is a state-of-the-art object detection method. In the Faster R-CNN method, detection is performed in two stages: the RoI extraction stage and the proposal classification stage. In the first stage, a feature extractor is applied to the entire image in order to generate the feature maps, and an RPN is used to extract the set of class-agnostic box proposals. These proposals are fed to a detection network to output the class-specific and box offset for each proposal. Since the RPN shares the full-image feature maps with the detection network, Faster R-CNN is very efficient in RPN extraction, as compared to other methods that use a pre-computed RPN (e.g., R-CNN [7]).

Although the Mask R-CNN [2] method is used for a different task (instance segmentation), this method has a strong connection with Faster R-CNN. Fig. 2 shows the differences between these two methods. The outputs of each model used to define the interested task are referred to as network heads. Mask R-CNN extends the Faster R-CNN by adding a mask branch for predicting an object mask in parallel with the existing bounding box prediction branch and the class prediction branch. The mask branch predicts a fixed-size mask ($m \times m$) for every class, resulting in an N binary mask, where N is the number of classes. Mask R-CNN is an effective framework for instance segmentation. The proposed method is based on the Mask R-CNN method, but the quality of segmentation is improved.

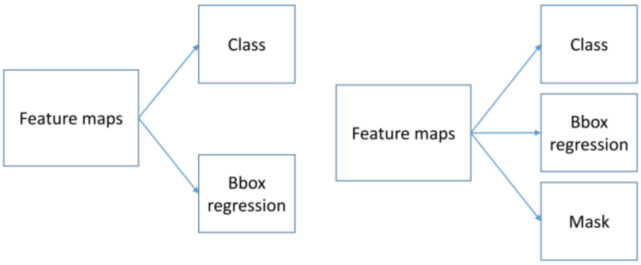


Fig. 2. Faster R-CNN (left) and Mask R-CNN (right). The main difference between these two methods is the number of outputs.

B. RoI Feature Extraction

Proposal-based methods such as Faster R-CNN and Mask R-CNN are incorporated into a pooling method in order to pool the features within each RoI into fixed-size features. The first method is RoiPool [4], which converts a features size $h \times w$ into a small feature map size $H \times W$ (e.g., 7×7). RoiPool operates by max-pooling a $H \times W$ grid of sub-windows of approximate size $h/H \times w/W$. RoiPool is a special case of spatial pyramid pooling in SPPNet [8]. Although RoiPool works well in the object detection task, it appears to hurt pixel-level prediction performance because the RoI and the extracted features are misaligned. In order to address this problem, Mask R-CNN introduces the RoiAlign pooling layer to replace the RoiPool layer. RoiAlign is a quantization-free layer that preserves the spatial localizations by using a bilinear interpolation [9] to compute the input features at four regular sample locations at each continuous RoI bin followed by max pooling. The extracted features have better-preserved spatial correspondence than RoiPool.

C. Multi-scale Feature Ensembling

In the deep network, combining multi-scale features can improve the performance. For segmentation tasks (including semantic segmentation and instance segmentation), some methods use this strategy by computing the partial scores for each class over multiple scales, such as FCN [1] and Hypercolumns [10]. Numerous methods use a similar strategy for object detection tasks, such as the FPN [11] and HyperNet [12].

Unlike the above methods, the proposed method is based on the concept of incorporating multi-scale features and precisely maintaining the RoI alignment. We replace the RoIAlign layer in the Mask R-CNN method by a subnetwork that ensembles the features within each RoI at multi-scales. This approach enables us to train the RoI pooling module and achieve better RoI feature representation.

III. Proposed Method

The proposed model is based on the same concept as Mask R-CNN, The network consists of two components. The first stage RPN extracts the set of network proposals, and the second stage uses a Fast R-CNN [4] to perform object classification, bounding box regression, and mask prediction from features that are extracted from each candidate box through a subnetwork.

A. Subnetwork Design

Fig. 3 shows the proposed model for the mask prediction branch. The subnetwork contains three branches, and at each branch, the features in the proposed RoI are cropped into different scales. In order to preserve the alignment between the RoI and the extracted features in the corresponding RoI, we use the “crop_and_resize” function in TensorFlow [13] to crop and bilinearly resize the input images to a fixed size. We use a 1×1 convolution layer after each “crop_and_resize” operator to maintain the number of outputs to 256. Feature maps are then down-sampled to the smallest fixed-size output (e.g. 14×14) by the average-pooling layer and concatenated. Finally, we use a convolution layer to reduce number of outputs to 256, and use the convolution layer as a prediction mask, as in Mask R-CNN.

We analyze two properties of the proposed subnetwork: (i) precise RoI and pooled RoI features alignment and (ii) multi-scale feature representation. First, the goal of the subnetwork is to extract small feature maps from each RoI to fixed-size feature, in the same manner as RoiPool and RoiAlign. Moreover, the “crop_and_resize” function and the pooling layer maintain the alignment between the RoI and the output features of each subnetwork branch. According to [2], this property is crucial for the instance segmentation task. Second, the proposed three-level sub-branch covers multi-scale RoI features. The ensembling features across multiple scales has proven beneficial in computer vision tasks [8].

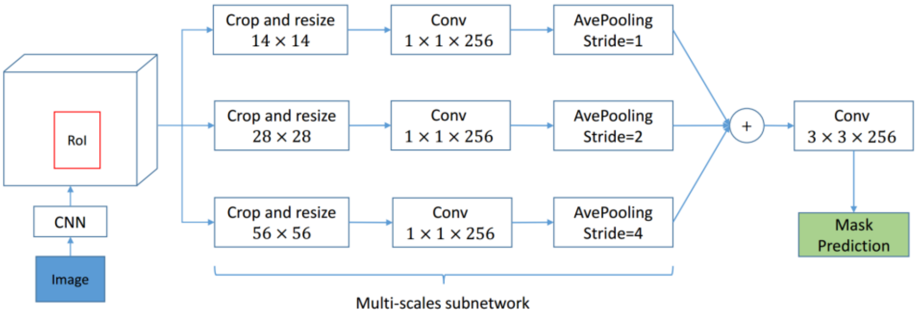


Fig. 3. Proposed subnetwork for the mask prediction branch.

B. End-to-end Training

We adopt end-to-end joint training of the RPN and the network heads. During training, the multi-task loss function is a combination of three losses: the classification loss ($\mathrm{L_{cls}}$), the box regression loss ($\mathrm{L_{box}}$), and the mask loss ($\mathrm{L_{mask}}$).

In order to predict the class and box regression, the class loss $\mathrm{L_{cls}}(p, u) = -\log(p_u)$ is the cross-entropy loss for true class u. The second loss $\mathrm{L_{box}} = L_{box}(t, t^*)$ is defined over a tuple of true bounding-box regression targets t and a predicted tuple t^*, as in Fast R-CNN [4]. The final loss $\mathrm{L_{mask}}$ is defined over K × m × m-dimensional output for each RoI, where K is the number of classes. The mask branch output is a pixel-wise binary classifier, because it outputs one mask for each class, and there is no competition between classes.

The overall training loss is defined as follows:

$$L = \mathrm{L_{cls}} + \lambda[u \geq 1]L_{box} + L_{mask}$$

where $[u \geq 1]$ is equal to 1 when $\mathrm{u} \geq 1$, and 0 otherwise. In the experiments, we set the balance loss weight λ to 1.

C. Training Details

We initialize the subnetwork convolution layers with MSRA [14] and constant initialization for biases. Through experiments, we found that normalizing the output of the convolutional layers by using a batch normalization [15] layer after convolution boosts the performance. The last convolutional layer after concatenation does not need a batch normalization layer.

We adjust the hyperparameters from Mask R-CNN [2] during training. The image is resized to a shorter edge of 800 pixels, and each GPU has a batch size of one image. For data augmentation and to prevent overfitting, we apply random flipping to the training dataset. We train the dataset in a 2-GPU system with a total of 720k iterations. The base learning rate is 0.0025 (the learning rate is reduced 10 times after 480k iterations and 640k iterations). The weight decay is 10^{-4}, and the momentum is 0.9.

A RoI is considered to be positive if it has an Intersection over Union (IoU) with a ground-truth box of at least 0.5. We maintain the ratio of positive to negative proposal RoI's as 1:3. The resolutions of RoI features for ResNet-50 and ResNet-101 are 14 × 14 and 28 × 28, respectively.

IV. Experimental Results

A. Dataset and Evaluation Metrics

We performed experiments on the COCO dataset [6], which is a large-scale object detection dataset that consists of 80 object categories. The training dataset contains approximately 118k images, including all of the training image (coco_2014_train) and a subset of valuation images (coco_2014_minusminival). Each object in the image is annotated with a bounding box and a binary mask inside the bounding box. We used the COCO API [16] to evaluate our results, which are measured by average precision (AP) over IoU in various thresholds and object scales. We report the AP in the 5,000-image valuation set (the minival set). The IoU threshold and the object scales are shown in Table I.

TABLE I: COCO Evaluation Settings

Evaluation terms	IoU (%)	Object size: A
AP	50-95	All
$\mathrm{AP_{50}}$	50	All
$\mathrm{AP_{75}}$	75	All
$\mathrm{AP_S}$	50-95	$\mathrm{A} < 32 \times 32$
$\mathrm{AP_M}$	50-95	$32 \times 32 < A < 96 \times 96$
$\mathrm{AP_L}$	50-95	$96 \times 96 < A$

B. Results

We tested our proposed model with different network settings. In order to demonstrate the effect of the proposed subnetwork, we used a different baseline for feature extraction over an input image. The results of the Mask R-CNN method, which uses the RoiAlign pooling layer, and the proposed method, which uses the subnetwork for RoI feature extraction, are compared. In order to make a faithful comparison, we attempt to train the methods under the same conditions, including the baseline, weight initialization, number of iterations, and system configuration. We also explored the effect of the proposed multi-scale feature pooling under the feature pyramids baseline (e.g., the feature pyramid network (FPN) [11]).

The first experiment is performed with ResNet-50 [17], which has a depth of 50 layers, and the features are extracted from the final convolutional layer of fourth stage. In this experimental setting, we do not use the feature pyramid extraction for baseline features. The results are shown in Table II. Overall, the proposed model APs are improved for all IoU threshold and object size settings. At a high value of IoU (75%), a gap in AP of 1.1% indicates that the proposed model is beneficial under good conditions.

TABLE II: Instance Segmentation Results for the ResNet-50 Baseline

Model	AP	$\mathrm{AP_{50}}$	$\mathrm{AP_{75}}$	$\mathrm{AP_S}$	$\mathrm{AP_M}$	$\mathrm{AP_L}$
Mask R-CNN	31.4	52.8	33.0	12.1	34.5	49.6
Proposed model	**31.8**	**53.5**	**34.1**	**12.7**	**34.8**	**50.8**

The second experiment is performed with a higher quality feature extractor, i.e., ResNet-101 [17] with FPN design. The FPN is used to augment a feedforward network (e.g., ResNet-101) with a top-down pathway and lateral connections. The FPN extracts the feature pyramid from a single-resolution input image. The extracted RoI features from different pyramid levels can be used for box detection and mask prediction. The overall AP is better than that of the model using the ResNet-50 baseline, which shows the relationship between the instance segmentation performance and the classification performance of the baseline on ImageNet [18]. Table III shows the experimental results in detail. The obtained results are better than those of Mask R-CNN at various settings. For IoU's ranging from 0.5 to 0.95, we obtained a mean AP of 37.7%, which is little bit greater than that of the Mask R-CNN model by 0.2%. However, the gaps are smaller than those in the first experiment. We believe that since the feature extractor (ResNet-101-FPN) outputs pyramid features, the effect of the proposed multi-scale subnetwork is not strong, as in the first experiment, which used a non-pyramid feature extractor.

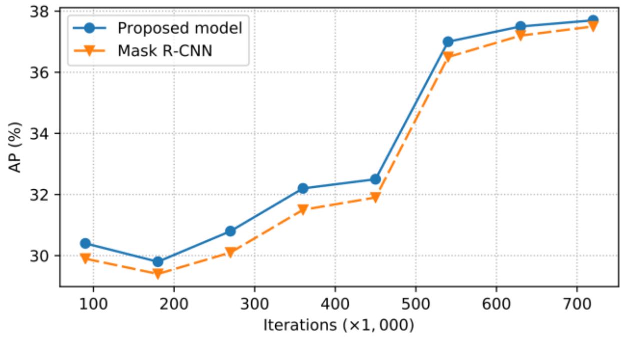


Fig. 2. Average precision evolution during training of Mask R-CNN vs. the proposed model using the ResNet-101-FPN baseline. The results are evaluated in the COCO minival set.

Fig. 4 shows the AP during training of Mask R-CNN and the proposed model. We trained both models under the same configuration and learning rate. The curve indicates the benefit of using the proposed subnetwork to replace the RoiAlign layer of the Mask R-CNN model. The proposed model increases the AP quicker than the Mask R-CNN, especially at beginning iterations, because the proposed model embeds the RoI features at multi-scale, and thus outputs more robust features for mask prediction.

C. Running Time and Memory Usage

We report the model running time of the proposed method in Table IV. Since the proposed subnetwork uses more parameters than the RoiAlign layer, the model running time is slower and uses more memory than Mask R-CNN. However, an increased time of less than approximately 20% and an increased memory usage of approximately 6% are reasonable.

TABLE III: INSTANCE SEGMENTATION RESULTS FOR THE RESNET-101-FPN BASELINE

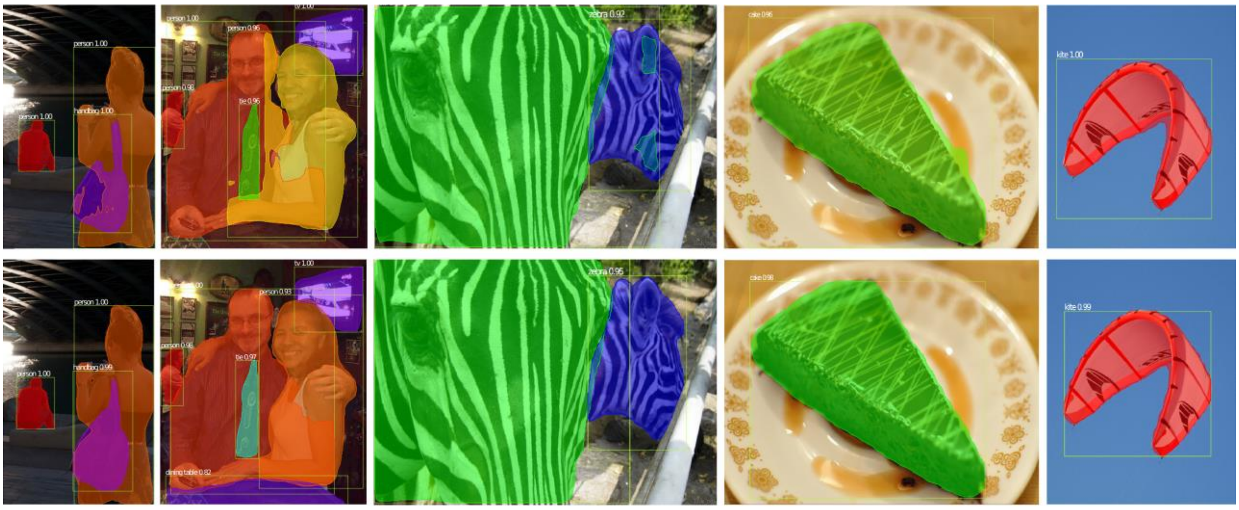
Model	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
Mask R-CNN	37.5	60.6	39.9	17.7	**41.0**	55.4
Proposed model	**37.7**	**60.7**	**40.2**	**17.8**	40.9	**55.6**

TABLE IV: RUNNING TIME (S) AND MEMORY USAGE (GB) OF THE PROPOSED MODEL COMPARED WITH MASK R-CNN

Model	Baseline	Test time	Memory
Mask R-CNN	ResNet-50	0.163	8.5
Proposed model	ResNet-50	0.195	9.3
Mask R-CNN	ResNet-101-FPN	0.251	6.6
Proposed Model	ResNet-101-FPN	0.260	7.0

D. Examples

We visualized the outputs of the proposed model in Fig. 5. The output segmentation is good under difficult conditions, such as crowded or small objects. Comparing with Mask-RCNN, our proposed model can output better segmentation regions (Fig. 5) and more precise bounding boxes (Fig. 5e).



(a) (b) (c) (d) (e)

Fig. 3. The example comparisons of Mask-RCNN (top row) and our proposed method (bottom row) on COCO dataset, each detection includes bounding box, label, confidence and segmentation region.

V. DISCUSSION

In the experimental results section, we performed the proposed network for two baselines (with and without pyramid feature extraction). Although the proposed model outperforms the Mask R-CNN using the same baseline, the gaps in the first model (using ResNet-50 as a baseline) are larger than those in the second model (using ResNet-101-FPN as a baseline). This is because the ResNet-101-FPN already performs masks prediction in multi-level layers. The proposed multi-scale RoI pooling subnetwork effectively augments pooled feature maps at multiple feature resolutions. Thus, the proposed model shows a great improvement using a non-pyramid feature extractor as a baseline.

Another problem with the proposed subnetwork is to decide hyperparameters, such as the number of sub-branches, the output size of cropped RoI features, and the number of outputs for each convolutional layer. Because of the limited GPU memory, we only perform three sub-branches at three sizes (14×14, 28×28, and 56×56).

VI. Conclusion

In the present paper, we introduce a simple but effective subnetwork, which replaces the RoI pooling layer used in the instance segmentation task. We crop the RoI features at multiple resolutions and concatenate the outputs to extract rich, multi-scale RoI feature maps. The experimental results reveal the advantage of using the proposed model design on mask prediction. In terms of the model complexity increment, the running time and memory usage is reasonable, as compared to the model that uses a RoIAlign pooling layer.

References

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *in Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431-3440.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *in Proc. the IEEE International Conference Computer Vision (ICCV)* , October 2017, pp. 2980-2988.

[3] 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*, pp. 91-99, 2015.

[4] R. Girshick, "Fast r-cnn," arXiv preprint arXiv:1504.08083, 2015.

[5] J. R. Uijlings, K. E. Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154-171, 2013.

[6] T. Y. Lin *et al*., "Microsoft coco: Common objects in context," *in Proc. the European Conference on Computer Vision*, September 2014, pp. 740-755.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580-587.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. the European Conference on Computer Vision*, September 2014, pp. 346-361.

[9] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," *Advances in Neural Information Processing Systems*, pp. 2017-2025, 2015.

[10] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 447-456.

[11] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *CVPR*, vol. 1, no. 2, p. 4, July 2017.

[12] T. Kong, A. Yao, Y. Chen, and F. Sun, "Hypernet: Towards accurate region proposal generation and joint object detection," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 845-853.

[13] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, and M. Kudlur, "Tensor flow: A system for large-scale machine learning," *OSDI*, vol. 16, pp. 265-283, November 2016.

[14] K. He, K. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. the IEEE International Conference on Computer Vision*, 2015, pp. 1026-1034, 2015.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[16] T. Y. Lin and P. Dollar, *Ms Coco Api*, 2016.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[18] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A arge-scale hierarchical image database," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248-255.

Tran Duy Linh is a PhD student at Graduate School of Science and Engineering, Teikyo University, Japan. He received the B.E. degree in information system from the Hanoi University of Science and Technology, Hanoi, Viet Nam in 2010 and received the M.S. degree in computer science from Ho Chi Minh City University of Technology, Ho Chi Minh City, Viet Nam, in 2015. His research interests include image and video processing. He is presently engaged in research on object detection using deep learning. He is a member of the Information Processing Society of Japan.

Masayuki Arai is a professor in the Graduate School of Sciences and Engineering at Teikyo University. He received his B.E. degree from Tokyo University of Science in 1981 and Dr. eng. degree from Utsunomiya University in 1995. His research interests include pattern recognition, natural language processing and information visualization. He is a member of the Information Processing Society of Japan and IEEE.