J Inf Process Syst, Vol.18, No.5, pp.701~710, October 2022
https://doi.org/10.3745/JIPS.02.0183

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Semantic Image Segmentation for Efficiently Adding Recognition Objects

Chengnan Lu[1] and Jinho Park[2,*]

Abstract

With the development of artificial intelligence technology, various methods have been developed for recognizing objects in images using machine learning. Image segmentation is the most effective among these methods for recognizing objects within an image. Conventionally, image datasets of various classes are trained simultaneously. In situations where several classes require segmentation, all datasets have to be trained thoroughly. Such repeated training results in low training efficiency because most of the classes have already been trained. In addition, the number of classes that appear in the datasets affects training. Some classes appear in datasets in remarkably smaller numbers than others, and hence, the training errors will not be properly reflected when all the classes are trained simultaneously. Therefore, a new method that separates some classes from the dataset is proposed to improve efficiency during training. In addition, the accuracies of the conventional and proposed methods are compared.

Keywords

Image Segmentation, Machine Learning, Object Detection

1. Introduction

Various neural networks have been implemented by increasing the depth of artificial neural networks based on the improvement of hardware performance. In addition, as the amount of information continues to increase, more data can be acquired and learned through a neural network. Furthermore, the ability of neural networks to perform tasks has gradually improved. However, owing to the increase in the complexity of neural networks and the amount of data, training once requires hours or days, particularly if large numbers of images and composite neural networks are used. The structures of neural networks that classify or segment images containing various classes of objects have been developed through considerable research. However, cases in which there are several classes in the dataset have not been trained extensively. Conventionally, datasets containing images of various classes are trained simultaneously. When the number of classes in the dataset increases, a dataset with new classes and the existing dataset are merged to form a new dataset, and then, the merged dataset is trained again (Fig. 1(a)).

In the process of recreating and training a new dataset, the training process for the classes that have been trained previously is duplicated, which is unnecessary. If the number of classes to be trained is

Manuscript received June 21, 2021; first revision September 27, 2021; accepted November 7, 2021.

* **Corresponding Author:** Jinho Park (c2alpha@ssu.ac.kr)
[1] Dept. of Media, Graduate School of Soongsil University, Seoul, Korea (qq1154641033@gmail.com)
[2] School of Media, Soongsil University, Seoul, Korea (c2alpha@ssu.ac.kr)

significantly smaller than the number of classes that have already been trained, training all the datasets again is not efficient. As mentioned earlier, the number of datasets is rapidly increasing, and each training session consumes a significant amount of time and resources. Moreover, if cloud services are used, unnecessary financial costs will be incurred. Nevertheless, the number of images for each class in the dataset can influence the training accuracy. Owing to the limitations of graphics memory, all datasets cannot be trained simultaneously. During training, the dataset is divided into several small subsets, and the subsets, which are called batches, are used to train the neural network. Neural networks are trained using the data in batches. If the neural networks are intensively trained using the error of the class that accounts for a large portion of the dataset, the classes with a small number of images in the dataset will not reflect the error precisely, because they cannot participate in all the batches equally. To improve the training accuracy of all the classes in the dataset and to utilize the training model for the classes that have already been trained when the number of classes increases in the dataset, we propose a new approach that independently trains the separated datasets instead of training all the datasets again (Fig. 1(b)). The contribution of this study is training separated datasets independently to improve the accuracy of the class that accounts for a small portion of the dataset, and to recycle the previously trained model effectively when the number of classes is increased.

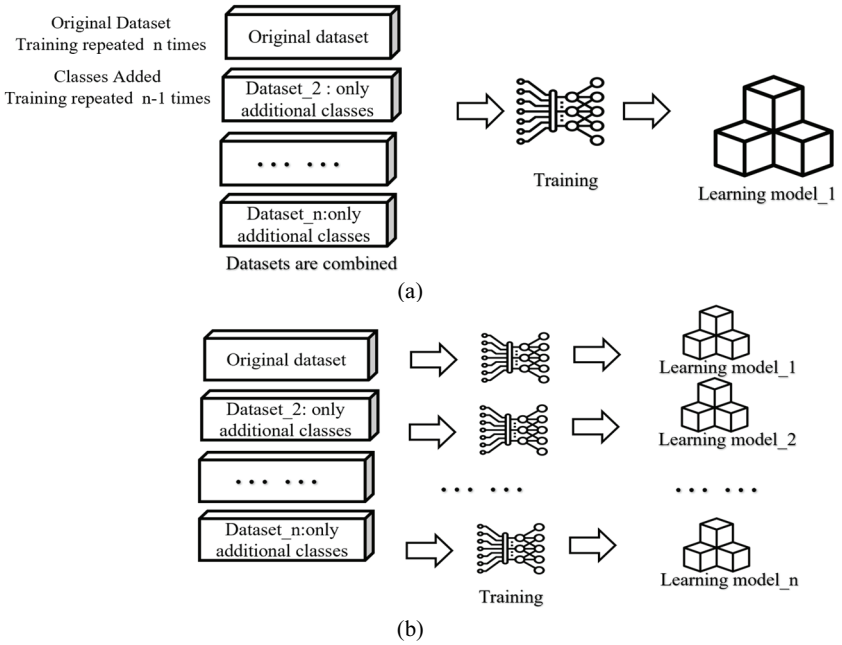


Fig. 1. Comparison of the conventional and proposed methods. (a) If all classes are trained simultaneously, the previously existing classes will be trained repeatedly. (b) All cases are trained once.

2. Related Works

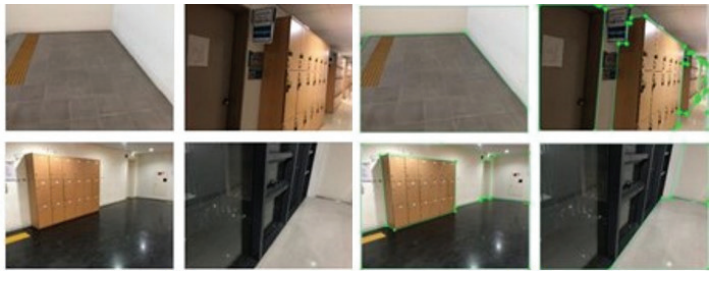
Various neural network structures have been proposed through the development of various technologies, such as image classification [1,2], object detection [3,4], and image segmentation [5,6]. Image classification is used to determine specific objects in an image. The performance of convolutional neural networks (CNNs) is usually improved through scaling up. An increase in the depth of the network from

ResNet-18 to ResNet-200 improved the performance of ResNet [7,8]. Object detection techniques exhibit slightly higher accuracy than image classification because they draw a bounding rectangle around the object. Image segmentation is the most accurate method for recognizing objects in an image. The object detection and image classification methods are based on the features of an image, whereas image segmentation is based on the pixel level. A CNN, which has several layers, extracts features from images using kernels to reduce the width and height of input images and increase the channel dimension. The output of image classification or object recognition is a vector, whereas that of image segmentation is an image. Obtaining the output of an image from a neural network is similar to the process of extracting image features using a CNN, but it requires a decrease in the channel dimension and an increase in the width and height [9]. Two types of image segmentation techniques exist: semantic [10,11] and instance segmentation. Semantic segmentation recognizes the same objects in an image as a group and displays them in the same color. By contrast, in instance segmentation, all objects in an image are independently identified and shown in different colors [12,13]. Image segmentation is of interest in various fields. U-Net [14] is a cell image segmentation technique that combines representative artificial neural networks with biomedical sciences. As generative adversarial networks (GANs) have recently become popular, attempts have been made to combine GANs and segmentation in biomedical sciences [15-17]. Image segmentation is necessary for autonomous driving systems, which receive images from a camera mounted on a vehicle. It is necessary to analyze the image and identify the obstacles around the vehicle accurately [18,19]. In the neural network structure used for image segmentation, the dimensions of the output layer are determined by the number of classes in the dataset. In addition, various neural network structures have been developed because the technology is attracting attention in various fields; however, several studies have not mentioned the method to be used when the number of classes in the dataset increases. Therefore, the existing learning models need to be recycled when the number of classes in the dataset increases.

3. Proposed Solution

3.1 Dataset Preparation

In this study, 700 indoor images were collected from seven buildings, and the objects were given labels, such as floors, doors, walls, and lockers. The labeling tool “Label Me” was used to label the images (Fig. 2). To check the effect of the number of images per class on a dataset, 600 images containing only a floor, door, and wall were collected. Subsequently, 100 images labeled with lockers were collected.



(a) (b)

Fig. 2. Data images for indoor environments. (a) Images collected from various indoor environments. (b) Add labels to target objects, such as floors, door walls, lockers, in collected images.

The target image uses the palette mode rather than the normal RGB mode to represent labels effectively. During the labeling process, a JSON file that generates target images from input images was created. This is because images in the palette format show reduced capacity, as only the color type of the pixel is represented instead of the RGB value. Thus, each pixel in the RGB images had a capacity of 24 bits, because there were three channels. In contrast, the images in the palette format had only one channel. Therefore, each pixel in the palette mode had a capacity of 8 bits. Fig. 3 shows the target images in the palette format created using JSON files.

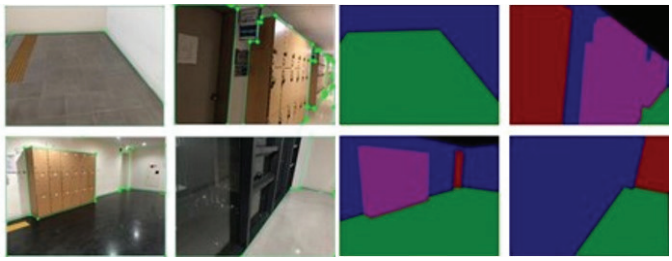


Fig. 3. Target image stored in the palette format with red, green, blue, and purple representing the door, floor, wall, and locker, respectively.

3.2 Training and Evaluation

Fig. 4(a) shows the solution proposed herein. When training for partial classes of the dataset was completed, some classes were trained separately if there were additional classes to be learned or if the number of partial classes of the dataset was significantly smaller compared with the number of other classes. Consequently, at the end of the training, several models were created, with the number of models equal to the number of subsets in the dataset.

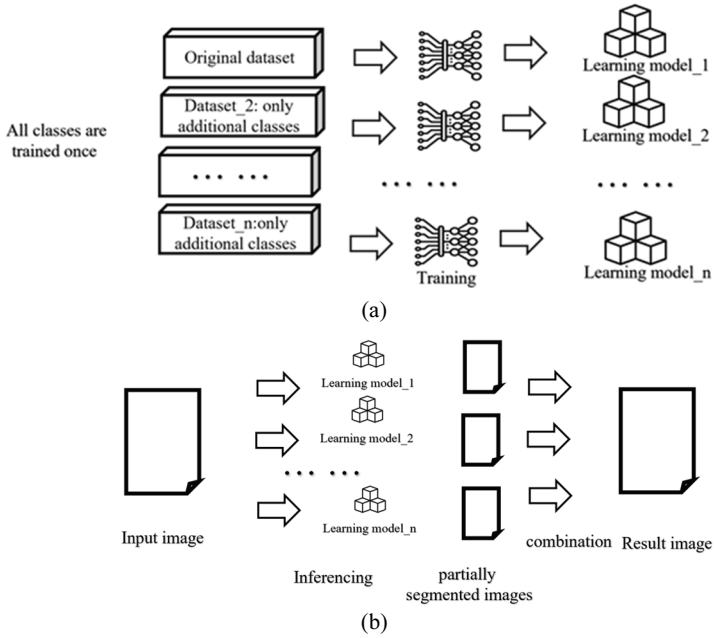


Fig. 4. Training and inference methods proposed herein. (a) During training all classes are trained only once. (b) During inferencing each model segments corresponding objects.

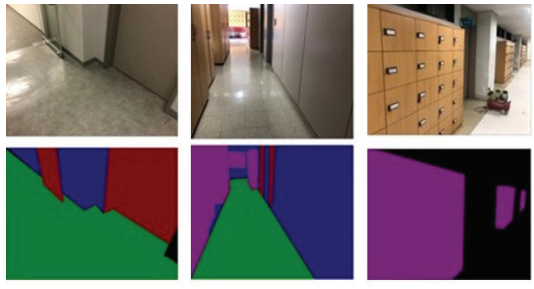
During inference, the proposed novel learning scheme generates several images corresponding to the number of models. As all classes that can be identified by all models are different, only the parts selected simultaneously by the overall models as the background should be treated as the real background. Fig. 4(b) illustrates the process of creating the resultant image by combining the images from the previous step.

4. Experiments

Two experiments were conducted to compare the experimental results intuitively. The conventional method was used in the first experiment, and the method proposed herein was used in the second experiment. Both experiments used DeepLab V3 as the backbone and ResNet-101 as the encoder. The batch size was 8, and the total number of epochs was 500. The batch size is the number of images that are simultaneously loaded into the memory on all days. The epoch is the number of times that all the datasets are repeated during training. The dataset is divided into three groups. The first group is a subset with a floor, door, wall, and locker, the second group is a subset with a floor, door, and wall, and the third group is a subset with only a locker. As shown in Fig. 5(a), only the floor, door and wall of the building were labeled. In the second group, the floor, wall, and door, along with the locker inside the building, were labeled simultaneously, as shown in Fig. 5(b). The remaining images were individually labeled as locker images in the building, as shown in Fig. 5(c).

The experiment was conducted in the following manner to compare the conventional and proposed methods. The conventional method collected the datasets again and relearned them when the number of classes in the dataset was increased. First, training was performed using a dataset with labels only for the floor, door, and wall, as shown in Fig. 5(a). The time taken for training was 7.75 hours. Then, training was performed again through a dataset with labels for the floor, wall, door, and locker by following the conventional method, as shown in Fig. 5(b). The time taken for training was 8.78 hours.

In this paper, we propose that classes added to the dataset should be learned separately. Fig. 5(c) shows a separate dataset that labels the locker among the image datasets. It required 2.28 hours to complete the training process to segment the locker, which is four times shorter than the time taken by the existing method when comparing the additional time, except for 7.75 hours.



(a) (b) (c)

Fig. 5. (a) Conventional image and image labeled with wall (blue), floor (green), and door (red). (b) Image in the palette mode, which is labeled with wall (blue), floor (green), door (red), and locker (purple). (c) Image labeled with locker (purple).

4.1 Loss Visualization

Fig. 6 shows the changes that occurred during the training process. The red curve represents the results of training only for the lockers. The blue curve shows the results of training for the floor, door, and wall. The orange curve indicates the results of training for the floor, door, wall, and locker simultaneously. The error is the smallest when the training occurs separately for changes in the classes in the dataset. The experimental conditions are listed in Table 1.

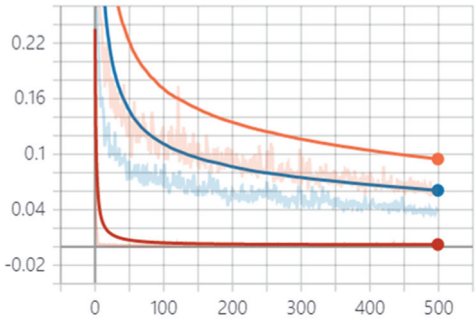


Fig. 6. Error changes in the training process.

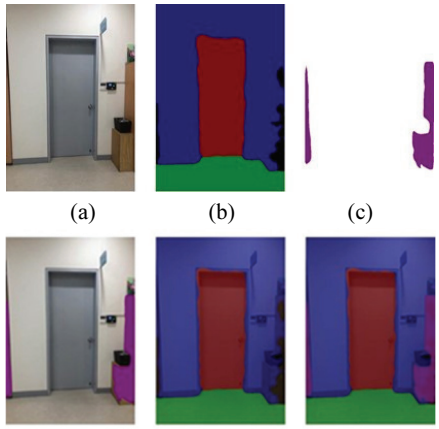
Table 1. Experimental environment

	Specification
Operating system	Ubuntu 2018 LTS
CPU	Intel i9 9900k
GPU	RTX 2080-ti
RAM	32 G
Language	Python 3.6.0
IDE	PyCharm 2021.1.1

4.2 Composition of Resultant Images

Two training models were created as a result of training using the method proposed herein: one segmented the images with the floor, wall, and door in indoor images, and the other segmented the images with the locker. The image segmentation training model identified only the previously trained objects in the image and painted them in a unique color, whereas the remaining background was black.

Fig. 7(a) shows an image that does not exist in the training dataset, and Fig. 7(b) shows the result of the segmentation, showing the floor, wall, and door in different colors using the method proposed herein. Fig. 7(c) shows the result of segmentation showing the locker in a unique color using the proposed method, in which the background is removed. Fig. 7(d) shows the result of segmenting only the locker in the image, and Fig. 7(e) shows the result of segmenting the floor, door, and wall in the image. Fig. 7(f) shows the result of overlapping the entire image created by the proposed method with the conventional image.



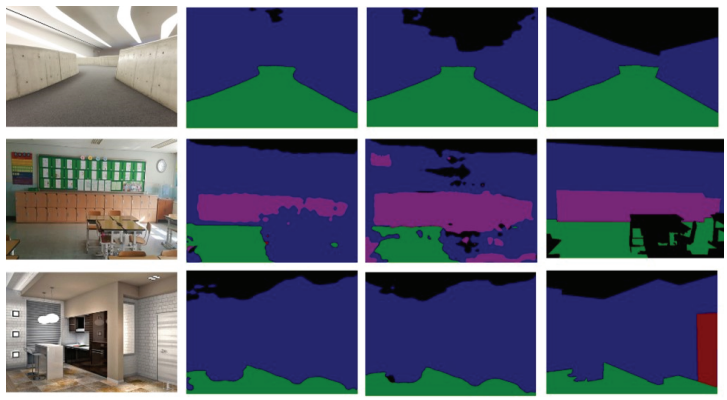
(a) (b) (c)

(d) (e) (f)

Fig. 7. Process of generating resultant images. (a) An indoor image as input of the model. (b) An input image is segmented with wall (blue), floor (green), and door (red). (c) An input image is segmented with locker (purple). (d) Superimposing the area identified as the locker in the image (a). (e) Superimposition of the results of segmenting floors, doors, and walls from the input image. (f) Overlapping the entire image created by the proposed method from the input image.

4.3 Result Visualization

The results of inference using the previous and proposed methods were compared. Fig. 8(a) shows an image that does not exist in the training dataset, and Fig. 8(b) shows the result of segmentation using the conventional method. Fig. 8(c) shows the segmentation results obtained using the proposed method, and Fig. 8(d) shows the ground-truth image of Fig. 8(a). The overall results do not differ much visually, although the time required for the training process is considerably reduced compared with that taken by conventional methods. The intersection-over-union (IoU) value compares common errors in image classification and image segmentation methods.



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

Fig. 8. Comparisons of the two methods. (a) Indoor images not included in the training dataset. (b) The outcome of the conventional method that all datasets are trained simultaneously. (c) The outcome of the proposed method that some classes are trained seperately. (d) The ground-truth of input images.

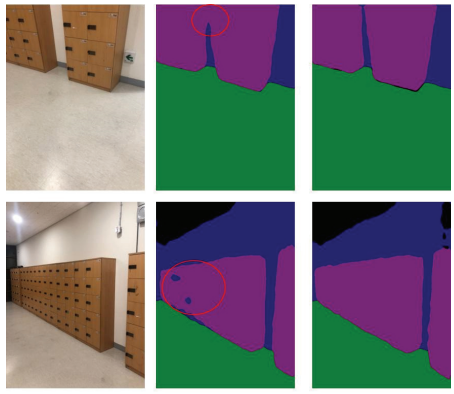
Fig. 9 shows the method of computation of the IoU, which shows the extent to which the predicted result matches the ground-truth image by calculating the ratio of the intersection and union of the two images. Table 2 compares the IoU values from the inference results of the conventional and proposed methods.

Fig. 9. Method of calculation of the IOU.

Table 2. IoU values from different training methods

Object	IOU (%)	
	Previous method	Proposed method
Wall	82.96	83.77
Floor	97.79	97.92
Door	76.03	74.95
Locker	91.66	92.43

Fig. 10 shows the result of segmentation for lockers, which have significantly fewer images than the other classes in the datasets. Fig. 10(a) shows an input image, and Fig. 10(b) shows the result of training all the objects simultaneously without considering the effect of the number of images per class on a dataset. Fig. 10(c) shows the results of training separately from the other classes. Consequently, if objects of different classes simultaneously participate in machine learning, the difference in their proportions in the entire dataset can affect training accuracy. A separate training method can be a solution for increasing the accuracy of training for objects in a class.



(a) (b) (c)

Fig. 10. Effect on training when significantly fewer classes are added to datasets and trained together. (a) Indoor image as input of the model. (b) Among the data sets, there are remarkably few images, including a locker, so a part of the locker in the image is not accurately segmented. (c) The problem shown in (b) solved by training the image which contains the locker separately.

5. Conclusion

Various techniques have been developed for identifying objects in an image through a neural network; however, the existing methods attempted to increase the accuracy of a dataset with fixed classes. Conventionally, if the number of classes in a dataset increases, the dataset is reconfigured and the entire dataset is trained again. This process is not necessary for classes that have already been trained; consequently, significant amounts of time and resources are consumed. All the datasets in a group can affect the learning accuracy of a partial class. This is because all the datasets cannot be trained simultaneously because of the limitation of the GPU capacity. During training, the neural network is trained using separate subsets, which are separated from the dataset by batch size. Consequently, the neural network cannot be properly trained owing to errors in a small number of classes in datasets. A new approach that efficiently reuses the previous training model instead of discarding the results is proposed. The testing showed that the subsets of datasets can be learned separately to solve the aforementioned problem.

In the future, we will investigate a new method that can eliminate unnecessary duplications in the inference process. If training is conducted using the proposed method, several learning models can be created. During the generation of the resultant image, the inference process should be repeated as many times as the number of training models there are. As the number of training models increases, the number of inferences correspondingly increases. This problem can be solved by stacking multiple neural networks, which is the same as the neural networks used in training separate datasets. The weights of the newly constructed neural network are obtained from several trained models.

Segmenting a small object in an image remains a difficult problem. In the label data, the background is labeled 0. If the target object is very small in the image, almost all parts are marked as zero values. In this situation, the neural network tends to output the value of all pixels to zero to minimize the error value. These problems can be solved by modifying the method used to calculate the errors. We know which pixels constitute the background in the target image. Therefore, during the training of the neural network, the accuracy can be improved by reducing the weight of errors calculated from the background pixels.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 1106-1114, 2012.

[2] H. C. Li, S. S. Li, W. S. Hu, J. H. Feng, W. W. Sun, and Q. Du, "Recurrent feedback convolutional neural network for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, article no. 5504405, 2021. https://doi.org/10.1109/LGRS.2021.3064349

[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*, vol. 28, pp. 91-99, 2015.

[4] JSpin, "Object detection," 2019 [online]. Available: https://nuggy875.tistory.com/20.

[5] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1520-1528.

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

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

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Computer Vision – ECCV 2016*. Cham, Switzerland: Springer, 2016, pp. 630-645.

[9] J. Fu, J. Liu, Y. Wang, J. Zhou, C. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *IEEE Transactions on Image Processing*, 2019. https://doi.org/10.1109/TIP.2019.2895460

[10] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2019, pp. 3146-3154.

[11] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2019, pp. 7519-7528.

[12] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: real-time instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, South Korea, 2019, pp. 9156-9165.

[13] Y. Lee and J. Park, "CenterMask: real-time anchor-free instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, WA, 2020, pp. 13903-13912.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham, Switzerland: Springer, 2015, pp. 234-241.

[15] M. Majurski, P. Manescu, S. Padi, N. Schaub, N. Hotaling, C. Simon, and P. Bajcsy, "Cell image segmentation using generative adversarial networks, transfer learning, and augmentations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA, 2019, pp. 1114-1122.

[16] N. Tajbakhsh, L. Jeyaseelan, Q. Li, J. N. Chiang, Z. Wu, and X. Ding, "Embracing imperfect datasets: a review of deep learning solutions for medical image segmentation," *Medical Image Analysis*, vol. 63, article no. 101693, 2020. https://doi.org/10.1016/j.media.2020.101693

[17] A. Hatamizadeh, A. Hoogi, D. Sengupta, W. Lu, B. Wilcox, D. Rubin, and D. Terzopoulos, "Deep active lesion segmentation," in *Machine Learning in Medical Imaging*. Cham, Switzerland: Springer, 2019, pp. 98-105

[18] D. Seichter, M. Kohler, B. Lewandowski, T. Wengefeld, and H. M. Gross, "Efficient RGB-D semantic segmentation for indoor scene analysis," in *Proceedings of 2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, pp. 13525-13531.

[19] L. Gao, Y. Zhang, F. Zou, J. Shao, and J. Lai, "Unsupervised urban scene segmentation via domain adaptation," *Neurocomputing*, vol. 406, pp. 295-301, 2020.

Chengnan Lu **https://orcid.org/0000-0002-4623-0077**

He received the B.S. degree in computer science and technology from Yanbian University of Science and Technology of China in 2018 and the M.S. degree from the Department of Media, Soongsil University in 2020. His research interests include machine learning and data science.

Jinho Park **https://orcid.org/0000-0002-8694-2976**

He received the B.S. and M.S. degrees in applied mathematics, in 1999 and 2001, respectively, and the Ph.D. degree in computer science from the Korea Advanced Institute of Science and Technology, in 2007. He is currently an associate professor with the Global School of Media, Soongsil University, South Korea. His research interests include augmented reality and machine learning