Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 21(1): 17-23, Mar. 2023 Regular paper

Comparison of the Effect of Interpolation on the Mask R-CNN Model

Young-Pill Ahn[1], Kwang Baek Kim[2], and Hyun-Jun Park[3]*, *Member*, *KIICE*

[1]Department of Computer Science, Chungbuk National University, Cheongju 28644, Korea
[2]Department of Artificial Intelligence, Silla University, Busan 46958, Korea
[3]Department of Artificial Intelligence Software, Cheongju University, Cheongju 28503, Korea

Abstract

Recently, several high-performance instance segmentation models have used the Mask R-CNN model as a baseline, which reached a historical peak in instance segmentation in 2017. There are numerous derived models using the Mask R-CNN model, and if the performance of Mask R-CNN is improved, the performance of the derived models is also anticipated to improve. The Mask R-CNN uses interpolation to adjust the image size, and the input differs depending on the interpolation method. Therefore, in this study, the performance change of Mask R-CNN was compared when various interpolation methods were applied to the transform layer to improve the performance of Mask R-CNN. To train and evaluate the models, this study utilized the PennFudan and Balloon datasets and the AP metric was used to evaluate model performance. As a result of the experiment, the derived Mask R-CNN model showed the best performance when bicubic interpolation was used in the transform layer.

Index Terms: Bicubic interpolation, Instance segmentation, Mask R-CNN, Transform layer

I. INTRODUCTION

Interest in deep learning vision started with classification. In 2012, AlexNet [2], which builds a deep CNN [3], drew widespread attention and many researchers have attempted to develop deeper models since [4,5]. In 2016, ResNet [6], a very deep model, demonstrated remarkable performance and excellent results for image classification.

After the remarkable performance in image classification, many researchers have begun to focus on object localization with classification. Most object detection models use the image feature extraction module by transferring learning from a model with high classification performance [7]. Therefore, the performance of the classification model also affects the object detection performance. The R-CNN model [8] and improved R-CNN models [9,10] led to successful results. In addition to two-stage models, such as R-CNN series models, one-stage models, such as YOLO [11,12], SSD [13,14], and RetinaNet [15], have also shown good results.

Object detection can recognize whether objects belong to the same class but not whether they are the same instance. Real-world problems require the ability to distinguish not only classes, but also instances. Therefore, by adding instance information to object detection, research has been conducted on instance segmentation that can recognize instances.

Mask R-CNN [1], a model for instance segmentation, outperformed the Faster R-CNN [10] model, which in turn performed best in the two-stage model [8-10]. Since then, several models derived from Mask R-CNN have been proposed [24-26]. Currently, the majority of the latest models use the Mask R-CNN structure as a baseline and improve their performance in various ways, such as using an improved

Received 21 October 2022, **Revised** 28 December 2022, **Accepted** 30 December 2022
***Corresponding Author** Hyun Jun Park (E-mail: hyunjun@cju.ac.kr)
Department of Artificial Intelligence Software, Cheongju University, Cheongju 28503, Republic of Korea

Open Access https://doi.org/10.56977/jicce.2023.21.1.17 print ISSN: 2234-8255 online ISSN: 2234-8883

backbone [27-30] or data augmentation [31]. Therefore, the performance of the Mask R-CNN derivative models is affected by the Mask R-CNN performance.

Mask R-CNN [1] is a family of R-CNN models [8] and is directly derived from Faster R-CNN [10] models. The backbone of Mask R-CNN is a Feature Pyramid Network (FPN) [32] structure.

The process and characteristics of Mask R-CNN are as follows. Features of the image are extracted by the FPN [32], and the features are passed to the RPN (Region Proposal Network). The RPN generates regions of interest (RoIs), which are boxes of various sizes containing object location information. For subsequent training, the extracted features are scaled to the same size using the RoIs. In Faster R-CNN [10], information is lost owing to quantization because the feature size is changed using the RoIPool method. On the other hand, Mask R-CNN prevents this information loss by scaling the feature size using the RoIAlign method. After the RoIAlign process, the network is divided into three branches, namely: classification, bounding box regression, and mask. The classification and bounding box regression branches had the same structure as the Fast R-CNN. Mask R-CNN adds mask branches in parallel to improve the speed and predict pixel-level segmentation masks and utilizes a modified FCN [21] structure. Using this structure, instance segmentation can be performed both quickly and accurately.

In the early days of deep learning, researchers focused on improving the performance through neural network model structures. However, many researchers have recently recognized the importance of data and are attempting to improve AI performance through data preprocessing.

Before the image, which is the input data, is processed in the backbone of Mask R-CNN, it is necessary to resize the image to a fixed size.

In several cases, bilinear interpolation is used to create images of the same size. Here, the Mask R-CNN resizes the input image using bilinear interpolation. In this study, the layer in which Mask R-CNN resizes the input image is called the transform layer.

Therefore, this study compares the overall performance change of Mask R-CNN when various interpolation methods are used in the transform layer. To do this, bicubic (which uses more pixels and has greater complexity than the bilinear), nearest-neighbor (uses fewer pixels), and bilinear interpolation were applied to transform the layer and compare the performance of Mask R-CNN.

The contribution of this study is to compare the performance change of the model according to the interpolation method and to determine which interpolation method can be used to achieve the best performance.

II. Transform Layer in Mask R-CNN

The transform layer of Mask R-CNN is shown in Fig. 1. In Mask R-CNN, the input image is passed through the transform layer before being fed to the backbone. The purpose of the transform layer is normalization and resizing.

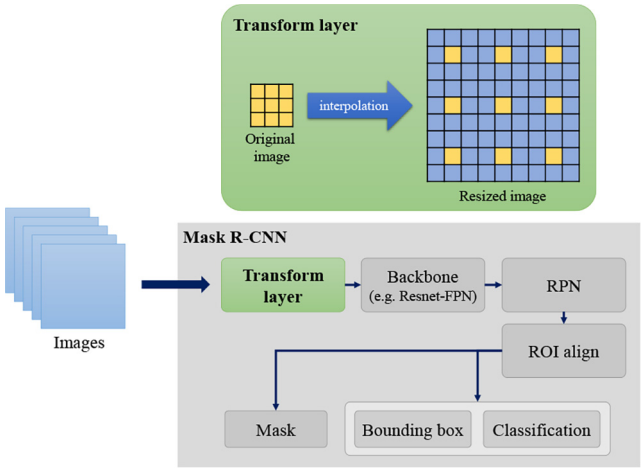


Fig. 1. Interpolation in transform layer and Mask R-CNN structure.

A. Image Data Preprocessing

In the transform layer, input images are resized as follows:

- Normalization
- Size determination
- Interpolation
- Padding
- Batching

1) Size Determination and Interpolation

The process from normalization to interpolation is represented in Fig. 2.

Algorithm 1 presents the detailed procedure for determining the image size in a range of configured values. In Algorithm 1, the variables *min* and *max* are set by a user and represent the ranges of the image height and width. The scale variable was determined based on the minimum size, maximum size, height, and width. The saved scale variable was then multiplied by the height and width of each image. The multiplied height and width (described as w×scale and h×scale in Algorithm 1) are the last numbers for determining the output size of the interpolation.

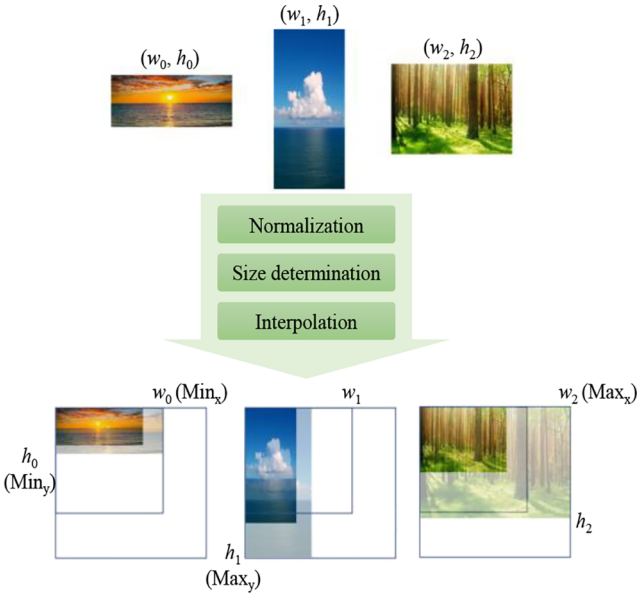


Fig. 2. The process from normalization to interpolation.

Algorithm 1. The detailed procedure of determining image size

Input: *image*, *min*, *max*
Output: resized image
1. $i_{max} \leftarrow$ max (w, h)
2. $i_{min} \leftarrow$ min (w, h)
3. $scale \leftarrow min / i_{min}$
4. if ($scale \times i_{max}$) > max then
5. $scale \leftarrow max / i_{max}$
6. end if
7. $image \leftarrow$ interpolation ($image$, $w \times scale$, $h \times scale$)
8. **return** resized image

2) Padding and Batching

The height and width of the resized images by interpolation were between the variable minimum size and maximum size. However, it cannot be said that all these images are of the same size. Therefore, before feeding these images to the backbone module, they need to be unified into one size. This process is illustrated in Fig. 3 and Algorithm 2.

First, it saves the best image size in the images that enter the transform layer (function max_by_axis). After determining the maximum height and width, the images are quantized by the unit size set by the user. The maximum image sizes (mH and mW) from the max_by_axis function are not the same as the quantized unit numbers, but are instead rounded up. In the rounding-up case, zero-padding is used to fill the blanks, after which images are unified in one size, and the dimensions are batch size × channel number × height × width. It is then ready to be moved to the backbone.

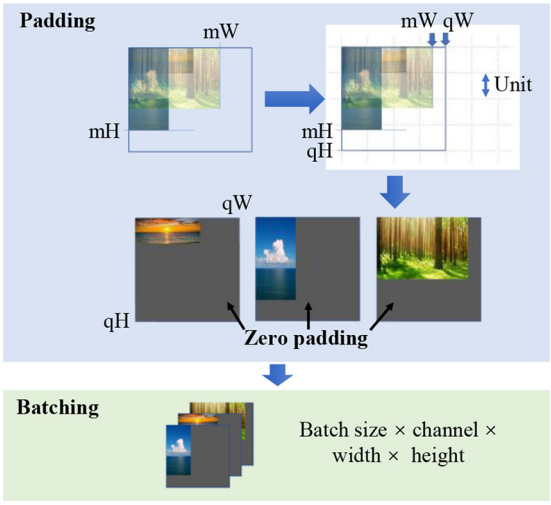


Fig. 3. The overall process of padding and batching.

Algorithm 2. The detailed process of padding

Input: resized images
Output: unified-size images
1. mW, mH ← max_by_axis (images)
2. qW, qH ← quantization (mW, mH, Unit)
3. images ← fill_zero (images, qW, qH)
4. **return** unified-size images

B. Diverse Interpolations for Transform Layer

Better input results in better models; thus, many researchers have tried to improve input by data preprocessing.

The inputs of the Mask R-CNN are scaled images. In the transform layer, the input images are scaled, and the images are passed to the backbone of the Mask R-CNN. Therefore, to improve the performance of Mask R-CNN, it is necessary to improve the input image generation process.

Given that Mask R-CNN uses interpolation to scale the images, the input images are determined by an interpolation method. Image interpolation refers to the calculation of pixel values at missing locations due to scaling. Usually, in image processing, nearest-neighbor, bilinear, and bicubic interpolations are used to scale an image. Therefore, this study used these three interpolations (nearest-neighbor, bilinear, bicubic) to compare the performance of Mask R-CNN. The characteristics of each method are as follows:

Nearest-neighbor interpolation is the simplest method and selects and interpolates the values of the nearest (closest) pixels. However, this method does not consider the values of neighboring pixels.

Bilinear interpolation is well known, intuitive, easy to implement, and interpolates two variables using repeated lin-

ear interpolation. It is performed using linear interpolation, first in one direction and then again in the other direction. Each step is linear, but the overall interpolation is quadratic. Bilinear interpolation is a basic resizing technique used for computer vision and image processing. However, it has a disadvantage in that it uses a small number of pixels to fill blank pixels.

Bicubic interpolation is an extension of cubic interpolation for interpolating data points in two-dimensions. The interpolated values are more accurate (natural) than the values obtained by bilinear interpolation or nearest-neighbor interpolation. Bicubic interpolation can be performed using Lagrange polynomial, cubic spline, or cubic convolution algorithms. In image resizing, bicubic interpolation is often the preferred choice over bilinear or nearest-neighbor interpolation when speed is not an issue because images with bicubic interpolation are smoother and less noisy.

C. Training Stage

Normalization was used for data pre-processing. Here [0.485, 0.456, 0.406] was set for each channel as the mean value, and [0.229, 0.224, 0.225] was set as the standard deviation. These values were calculated the ImageNet [35] dataset.

For the total loss calculation, the sum of the bounding box, classification, and mask losses were used [1]. In classification loss, cross-entropy [36] is used to calculate the values from softmax. In the mask loss, binary cross-entropy is used for each instance value.

The layers in the model were initialized by Kaiming He initialization [37], excluding the backbone. For the optimization, a learning rate of 0.005, a momentum of 0.9, and a weight decay of 0.0005 was used. A batch size of four units per GPU was used. One GPU was used for training; therefore, the total batch size was four. Resnet-50-FPN, where 50 indicates the number of layers [6, 32] and a pretrained backbone [7] on ImageNet [35] were also used. Table 1 lists the Mask R-CNN settings.

Table 1. Model setting

Data preprocessing	normalization
Loss function	bounding box loss + classification loss + mask loss
Weight initialization	Kaiming He [37]
Learning rate	0.005
Momentum	0.9
Weight decay	0.0005
Batch size	4/GPU

III. EXPERIMENT

The model performance was evaluated by changing the interpolation of the Mask R-CNN transform layer module to bicubic, nearest, or bilinear.

Various metrics can be used to evaluate object detection [38]. In this study, well-known mean Average Precision (mAP) metrics were used.

There are bounding box AP and mask AP for evaluating the bounding box and mask, respectively. Each metric has AP, AP50, and AP75 with different thresholds (AP50: threshold >50, AP75: threshold >75, AP: average from AP50 to AP95 in increments of five).

A. Experimental Environment and Dataset

A system comprising of an NVIDIA RTX 2080Ti (GPU memory:11016MiB) GPU and 32 GB RAM was used to run the evaluations and the Pytorch framework [34] was used to implement the models.

The PennFudan and Balloon datasets [39,40], both of

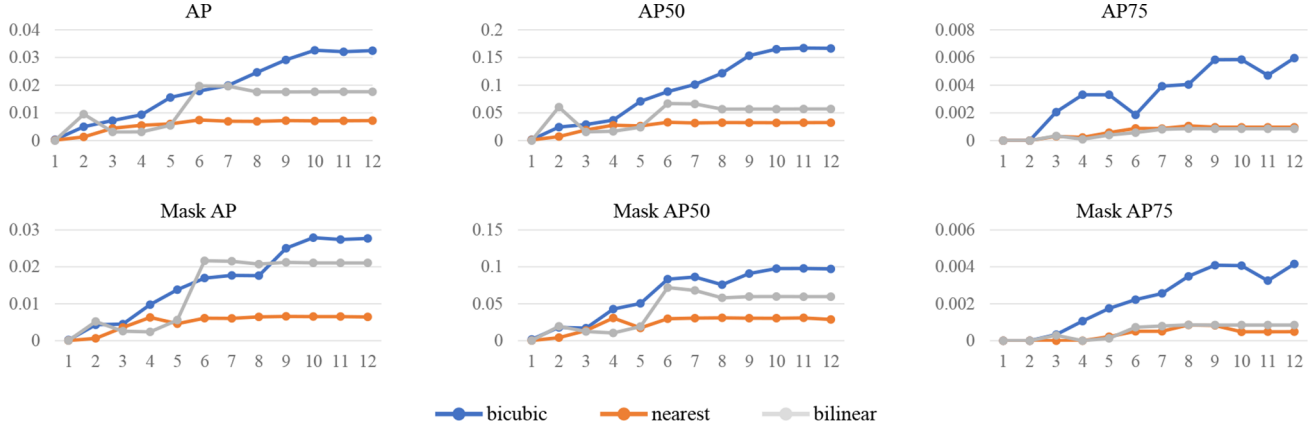


Fig. 4. Bounding box AP on PennFudan (X axis: epochs).

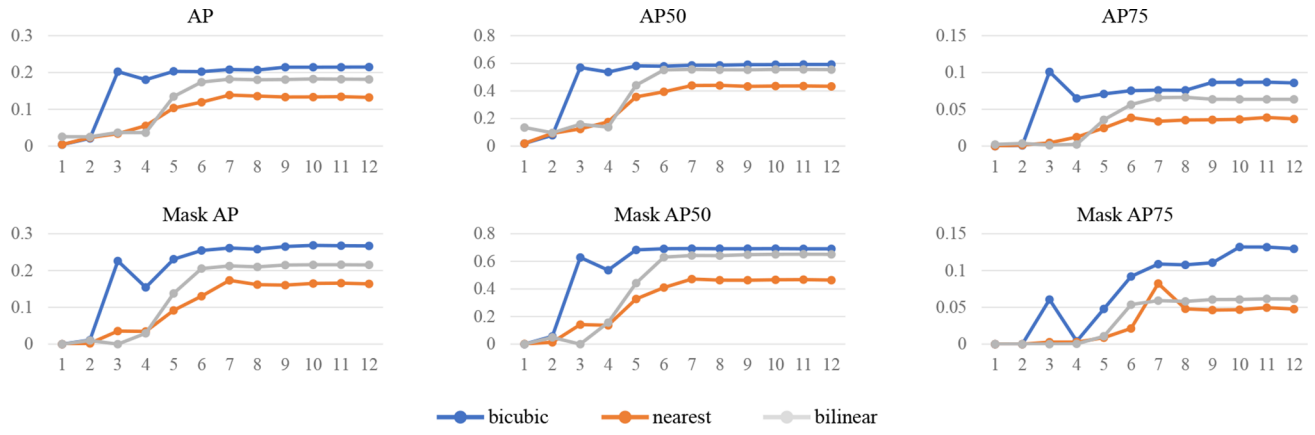


Fig. 5. Experiment results on Balloon (X axis: epochs).

which provide annotations for instance segmentation, were utilized in this study. Since these two datasets are relatively small, it is good to quickly experiment with the effect of the interpolation method. The PennFudan dataset contains only a pedestrian class. When learning, the total number of classes was two, including the background. The total number of images was 100, with 50 used as the test set. Similarly, the balloon dataset contains only one balloon class. When learning, the total number of classes was two, including the background. The total number of images was 62, with 13 used as the test dataset.

C. Experimental Results and Analysis

Experimental results on PennFudan and Balloon are shown in Fig. 4, Fig. 5, Table 2, and Table 3.

Table 2. Bounding box AP and mask AP on PennFudan dataset (epoch 11)

method	AP	AP50	AP75	MaskAP	MaskAP50	Mask75
bicubic	0.0324	0.1665	0.0059	0.0276	0.0972	0.0041
Bilinear	0.0176	0.0571	0.0009	0.0210	0.0596	0.0008
nearest	0.0072	0.0326	0.0010	0.0064	0.0287	0.0004

Table 3. Bounding box AP and mask AP on balloon dataset (epoch=11)

method	AP	AP50	AP75	MaskAP	MaskAP50	Mask75
bicubic	0.2149	0.5916	0.0858	0.2668	0.6909	0.1296
Bilinear	0.1815	0.5545	0.0635	0.2152	0.6507	0.0613
nearest	0.1324	0.4334	0.0369	0.1637	0.4640	0.0475

The bicubic model outperformed the bilinear and nearest models for the six metrics in both datasets. In many cases, the bilinear model outperformed the nearest model. In the Balloon dataset AP75, the difference between the bilinear model and the closest model was very small. For AP75 with a high threshold, there was no significant difference between the bilinear and nearest interpolation.

To obtain excellent results, all models must be able to extract good features. Bicubic interpolation creates more accurate input images than the other interpolation methods, enabling better features to be extracted. In other words, using the bicubic interpolation method means that better features can be extracted from better input images, thereby improving the performance of the model.

Therefore, it is better to use bicubic interpolation to obtain the best results from a model that uses interpolation. However, the bicubic interpolation method has the disadvantage of requiring a long time owing to the large number of calculations required compared to other interpolation methods.

IV. CONCLUSIONS

Mask R-CNN is widely known as a baseline for instance segmentation. The interpolation performed by the Mask R-CNN transform module affects the performance of the Mask R-CNN. Therefore, this study tested the performance change of the Mask R-CNN when various interpolation methods were used.

Three interpolation methods (bilinear, nearest, and bicubic) were used in the experiment, and the Penn-Fudan and Balloon datasets were used. The results of the experiment showed that the double cubic method was the best among all tested cases and in most cases, the bilinear method yielded better results than the nearest-neighbor method.

When studying a Mask R-CNN-based derived model, a performance improvement can be expected if the interpolation method is changed from bilinear to bicubic in the transform layer.

ACKNOWLEDGMENTS

This work was supported by the National Research Foun-

dation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2022-00166722).

REFERENCES

[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, Venice, Italy, pp. 2961-2969, 2017. DOI: 10.1109/ICCV.2017.322.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 60, no. 6, pp. 84-89, May. 2012. DOI: 10.1145/3065386.

[3] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[4] C. Szegedy, W. Liu, Y. Jia, and P. Sermanet, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, pp. 1-9, 2015. DOI: 10.1109/cvpr.2015.7298594.

[5] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, [Online]. Available: https://arxiv.org/abs/1409.1556.

[6] 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, USA, pp. 770-778, 2016. DOI: 10.1109/cvpr.2016.90.

[7] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus, USA, pp. 1717-1724, 2014. DOI: 10.1109/cvpr.2014.222.

[8] R. 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*, Columbus, USA, pp. 580-587, 2014. DOI: 10.1109/cvpr.2014.81.

[9] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, pp. 1440-1448, 2015. DOI: 10.1109/iccv.2015.169.

[10] 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.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, pp. 779-788, 2016. DOI: 10.1109/cvpr.2016.91.

[12] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, pp. 6517-6525, 2017. DOI: 10.1109/cvpr.2017.690.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Amsterdam, Netherlands, pp. 21-37, 2016. DOI: 10.1007/978-3-319-46448-0_2.

[14] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, Dssd: Deconvolutional single shot detector, 2017, [online]. Available: https://arxiv.org/abs/1701.06659.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, Venice, Italy, pp. 2980-2988, 2017. DOI: 10.1109/iccv.2017.324.

[16] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *European conference on computer vision*, Zurich, Switzerland, pp. 297-312, 2014. DOI: 10.1007/978-3-319-10584-0_20.

[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, pp. 447-456, 2015. DOI: 10.1109/cvpr.2015.7298642.

[18] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, USA, pp. 3992-4000, 2015. DOI: 10.1109/cvpr.2015.7299025.

[19] J. R. R. Uijlings, K. E. A. Van De Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154-171, Apr. 2013. DOI: 10.1007/s11263-013-0620-5.

[20] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus, USA, pp. 328-335, 2014. DOI: 10.1109/CVPR.2014.49.

[21] 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, USA, pp. 3431-3440, 2015. DOI: 10.1109/cvpr.2015.7298965.

[22] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *European Conference on Computer Vision*, Amsterdam, Netherlands, pp. 534-549, 2016. DOI: 10.1007/978-3-319-46466-4_32.

[23] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, pp. 2359-2367, 2017. DOI: 10.1109/cvpr.2017.472.

[24] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 6154-6162, 2018. DOI: 10.1109/CVPR.2018.00644.

[25] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, pp. 8759-8768, 2018. DOI: 10.1109/CVPR.2018.00913.

[26] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, pp. 4974-4983, 2019. DOI: 10.1109/cvpr.2019.00511.

[27] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S Lin, and B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, 2021, [online]. Available: https://arxiv.org/abs/2103.14030.

[28] Y. Liu, Y. Wang, S. Wang, T Liang, Q. Zhao, Z. Tang, and H. Ling, "Cbnet: A novel composite backbone network architecture for object detection," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 07, pp. 11653-11660, Apr. 2020. DOI: 10.1609/aaai.v34i07.6834.

[29] T. Liang, X. Chu, Y. Liu, Y. Wang, Z. Tang, W. Chu, J. Chen, and H. Ling, CBNetV2: A Composite Backbone Network Architecture for Object Detection, 2021, [online]. Available: https://arxiv.org/abs/2107.00420.

[30] X. Du, T-Y. Lin, P. Jin, G. Ghiasi, M. Tan, Y. Cui, Q. V. Le, and X.

Song, "SpineNet: Learning scale-permuted backbone for recognition and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, USA, pp. 11592-11601, 2020. DOI: 10.1109/cvpr42600.2020.01161.

[31] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple copy-paste is a strong data augmentation method for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nashville, USA, pp. 2918-2928, 2021. DOI: 10.1109/cvpr46437.2021.00294.

[32] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, pp. 2117-2125, 2017. DOI: 10.1109/cvpr.2017.106.

[33] G. Bradski and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc., 2008.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026-8037, 2019.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosia, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211-252, Apr. 2015. DOI: 10.1007/s11263-015-0816-y.

[36] P. T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19-67, Feb. 2005. DOI: 10.1007/s10479-005-5724-z.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, pp. 1026-1034, 2015. DOI: 10.1109/iccv.2015.123.

[38] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Niteroi, Brazil, pp. 237-242, 2020. DOI: 10.1109/iwssip48289.2020. 9145130.

[39] L. Wang, J. Shi, G. Song, and I. Shen, "Object detection combining recognition and segmentation," in *Asian conference on computer vision*, pp. 189-199, 2007. DOI: 10.1007/978-3-540-76386-4_17.

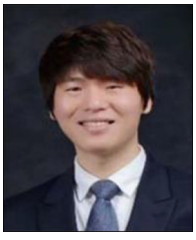
[40] W. Abdulla, Mask r-cnn for object detection and instance segmentation on keras and tensorflow, 2017, [online]. Available: https://github.com/matterport/Mask_RCNN.

Young Pill Ahn

Young Pill Ahn received his B.S. degree in Computer Science from Academic Credit Bank System, Korea, in 2021. He is currently pursuing a M.S. at Chungbuk National University, Korea. His major research interests include instance segmentation, super-resolution, and deep learning vision.

Kwang Baek Kim

Kwang Baek Kim received his M.S. and Ph.D. degrees from the Department of Computer Science, Pusan National University, Busan, Korea in 1993 and 1999, respectively. From 1997-2020, he was a professor at the Department of Computer Engineering, Silla University, Korea. From 2021 to the present, he is a professor at the Department of Artificial Intelligence, Silla University, Korea. He is currently an associate editor for Journal of Intelligence and Information Systems. His research interests include fuzzy clustering and applications, machine learning, and image processing.



Hyun Jun Park

Hyun Jun Park received his M.S. and Ph.D. degrees from the Department of Computer Engineering, Pusan National University, Busan, Korea, in 2009 and 2017, respectively. In 2017, he was a postdoctoral researcher at BK21PLUS, Creative Human Resource Development Program for IT Convergence, Pusan National University, Korea. From 2018 to the present, he has worked as an associate professor at the Department of Artificial Intelligence Software, Cheongju University, Korea. His research interests include computer vision, image processing, factory automation, neural network, and deep learning applications.