JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 16, No. 1, March 2022, pp. 14-24

Improvement in Object Detection Using Multi-Scale RoI Pooling and Feature Pyramid Network

Seungtae Nam
Department of Artificial Intelligence, Sungkyunkwan University, Suwon, Korea
stnamjef@skku.edu

Daeho Lee*
Department of Software Convergence, Kyung Hee University, Yongin, Korea
nize@khu.ac.kr

Abstract
The feature pyramid network (FPN) enhances the localization accuracy and detection performance of small objects using multiple scales of the features. FPN adopts lateral connections and a top-down pathway to make low-level features semantically more meaningful. However, it uses only single-scale features to pool regions of interest (RoIs) when detecting objects. In this study, we showed that single-scale RoI pooling may not be the best solution for accurate localization and proposed multi-scale RoI pooling to improve the minor drawbacks of the FPN. The proposed method pools RoIs from three feature levels and concatenates the pooled features to detect objects. Thus, the FPN with multi-scale RoI pooling, called FPN+, detects objects by taking into account all information scattered across three feature levels. FPN+ improved the FPN by 2.81 and 1.1 points in COCO-style average precision (AP) when tested on PASCAL VOC 2007 test and COCO 2017 validation datasets, respectively.

Category: Human-Computer Interaction

Keywords: Region of interest pooling; Feature pyramid network; Object detection; Deep learning

I. INTRODUCTION

Object detectors based on convolutional neural networks (CNNs) typically use high-level features to detect objects [1-4]. Since these features are translation-invariant and are capable of representing high-level semantics, CNN-based detectors outperform conventional detectors that use feature extraction algorithms such as scale-invariant feature transform (SIFT) [5] and histogram of oriented gradients (HOG) [6]. However, CNN-based detectors struggle to detect small objects due to the low resolution of high-level features. In particular, spatial information of the small objects is significantly lost as down-samplings are repeated in the CNN.

The feature pyramid network (FPN) [7] enhances the localization accuracy and detection performance of small objects using multi-scale features. It is important to use low-level features to detect small objects because these features are less down-sampled and thus, preserve detailed spatial information. However, low-level features are semantically less meaningful than high-level features. The FPN adopts lateral connections and a top-down pathway to enrich the visual semantics of low-level features. The lateral connections compress the features

Open Access http://dx.doi.org/10.5626/JCSE.2022.16.1.14 http://jcse.kiise.org

Received 11 October 2021; **Accepted** 04 January 2022
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

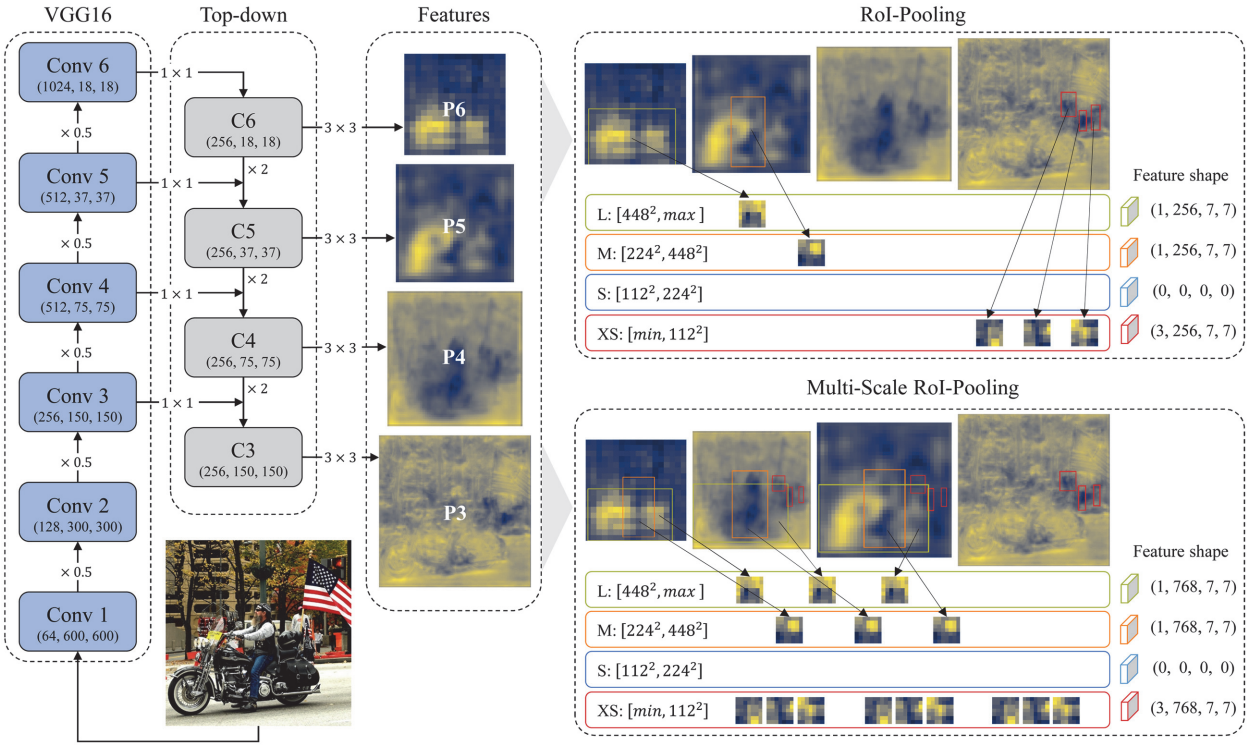

Fig. 1. The overall architecture of FPN and FPN+.

by 1×1 convolution. The top-down pathway enlarges the high-level features by bilinear interpolation and adds them to the low-level features passed through the lateral connections, as shown in Fig. 1. The FPN achieved a COCO-style average precision (AP) of 33.9% with these two structures, which exceeded that of the Faster R-CNN by 2.3 points [7].

The FPN uses four different feature scales to generate region proposals, but it uses only single-scale features to pool the regions of interest (RoIs) when detecting objects. As shown in the upper right of Fig. 1, the large-sized RoIs are pooled from the small-scale features, whereas the small-sized RoIs are pooled from the large-scale features. This seems plausible given that the low-level features preserve detailed spatial information and have semantics enriched by the top-down pathway. However, semantic gaps between the features at different levels may remain.

We experimentally verified two significant hypotheses: (1) the top-down pathway makes low-level features semantically more meaningful, and (2) using large-scale features has an advantage over using small-scale features when detecting small objects. We visualized different feature levels using a gradient-weighted class activation map (Grad-CAM) [8]. We found that the top-down pathway did not enrich the visual semantics of the low-level features as much as those of the high-level features. Additionally, using the large-scale features was not always advantageous for detecting small objects over using the small-scale features.

Based on our findings, we proposed an improved FPN, called FPN+, which has multi-scale RoI pooling to overcome the drawbacks of the FPN, as shown in the lower right of Fig. 1. This method pools RoIs from three feature levels and concatenates the pooled features to detect objects. Thus, FPN+ detects objects by taking into account all information scattered across three feature levels. The AP of FPN+ was increased by 2.81 and 1.1 points when tested with the PASCAL VOC 2007 test and COCO 2017 validation datasets, respectively.

II. RELATED WORK

A. Multi-scale Object Detectors

The single shot detector (SSD) [9] adds several convolutional layers to VGG16 [10] and detects objects using multi-scale features extracted from these layers. SSD512 achieved an AP of 26.8% in the COCO 2015 test dataset, which improved the Faster R-CNN by 2.6 points [9]. However, the model cannot robustly detect small objects. The AP over small (AP_S), AP over medium (AP_M), and AP over large (AP_L) objects were 9.0%, 28.9%, and 41.9%, respectively [9]. This was because the SSD only uses small-scale features. Since the spatial information dissipates as the scale of the features decreases, the model should have additionally used large-scale features to improve the detection performance for small objects.

The spatial dependent pooling network (SDPNet) [11] also uses multi-scale features to improve the detection performance for small objects. The SDPNet assumes that the visual semantics of objects could emerge at different feature levels depending upon the size of objects. For example, the visual semantics of small objects could emerge in the low-level features, whereas those of the large objects could emerge in the high-level features. Therefore, RoI pooling is performed on multiple levels of features depending upon the size of the objects, and the pooled features are passed through scale-dependent classifiers. The SDPNet achieved a PASCAL-style AP (AP@0.5) of 68.2% in the PASCAL VOC 2007 test dataset, which improved the Fast R-CNN by 1.3 points [11].

Cai et al. [12] proposed a multi-scale CNN (MS-CNN) and pointed out that there were inconsistencies between the sizes of objects and the filter receptive fields. For example, if the size of an object was smaller than the receptive field, unnecessary contextual information in the receptive field may lead to false detections. MS-CNN uses multi-scale features to compensate for the inconsistencies and attaches independent classifiers to handle each feature level. All of the features are passed through a deconvolutional layer to restore the spatial information before RoI pooling.

Hyper network (HyperNet) [13] uses additional low-level features to localize objects more accurately. HyperNet assumes that high-level features do not provide sufficient spatial information due to their coarseness. The model uses three feature levels (low-level, intermediate-level, and high-level features) by concatenating them into a single feature block. Since the features have different scales, the concatenating process inevitably requires up-sampling or down-sampling. The high-level features are up-sampled through a deconvolutional layer, and the low-level features are down-sampled through a max-pooling layer. Additional convolutional layers are attached to the top of the different levels of the features to extract semantically more meaningful features and compress all the features into a uniform space [13]. HyperNet achieved an AP@0.5 of 76.3% in the PASCAL VOC 2007 test dataset, which improved the Faster R-CNN by 3.1 points [13].

In summary, the SDPNet, MS-CNN, and HyperNet use additional low-level features with sufficient spatial information to localize objects more accurately. However, these methods did not overcome the limitation that the visual semantics weaken as the feature levels decrease. In contrast, SSD used additional high-level features, but it did not prevent the loss of spatial information.

B. Visualization of Convolutional Neural Networks

Simonyan et al. [14] proposed a method to visualize the CNN by calculating the gradient of an output vector with respect to the input image. This allowed them to observe the part of the input image that contributed the most to classifying the image into a particular class. Zeiler and Fergus [15] presented a method to visualize the features extracted from each layer of the CNN by mapping them into the input image space. These two methods are similar in that they use backpropagation to visualize the CNN. However, they differ in how they handle backpropagation through rectifier linear units (ReLUs). The former sets the gradient to zero when a bottom input signal is negative as

$$R^l(i,j) = (f^l(i,j) > 0) \cdot R^{l+1}(i,j), \tag{1}$$

where $f^l(i,j)$ denotes the activation at location (i,j) of the feature maps in the l-th layer, and $R^{l+1}(i,j) = \partial f^{out} / \partial f^{l+1}(i,j)$. This method is the same as vanilla backpropagation. The latter sets it to zero when the top gradient signal is negative as

$$R^l(i,j) = (R^{l+1}(i,j) > 0) \cdot R^{l+1}(i,j). \tag{2}$$

Springenberg et al. [16] combined these two methods and set the gradient to zero when the bottom input signal or the top gradient signal was negative as

$$R^l(i,j) = (f^l(i,j) > 0) \cdot (R^{l+1}(i,j) > 0) \cdot R^{l+1}(i,j). \tag{3}$$

Zhou et al. [17] presented a method to generate a class activation map (CAM) using global average pooling (GAP) [18]. For a given activation of the last convolutional layer, GAP is defined by

$$F^k = \frac{1}{Z} \sum_i \sum_j f^k(i,j), \tag{4}$$

where k denotes the channel index.

As shown in the upper left of Fig. 2, GAP averages the activations of each channel and produces a vector of length k. By computing the product of this vector and the corresponding weights of the last fully connected layer, a class c score Y^c is calculated by

$$Y^c = \sum_k w_k^c F^k. \tag{5}$$

The rest of the training process is the same as training a traditional backpropagation model. CAM [17] is defined by a weighted linear sum of the feature maps as

$$\mathrm{CAM}^c(i,j) = \sum_k w_k^c f^k(i,j), \tag{6}$$

where w_k^c is the weight connecting f^k and Y^c.

CAM shows the part of the feature maps that contributes to the activation of a neuron corresponding to class c. However, it is only applicable to a particular kind of CNN architecture performing global average pooling over

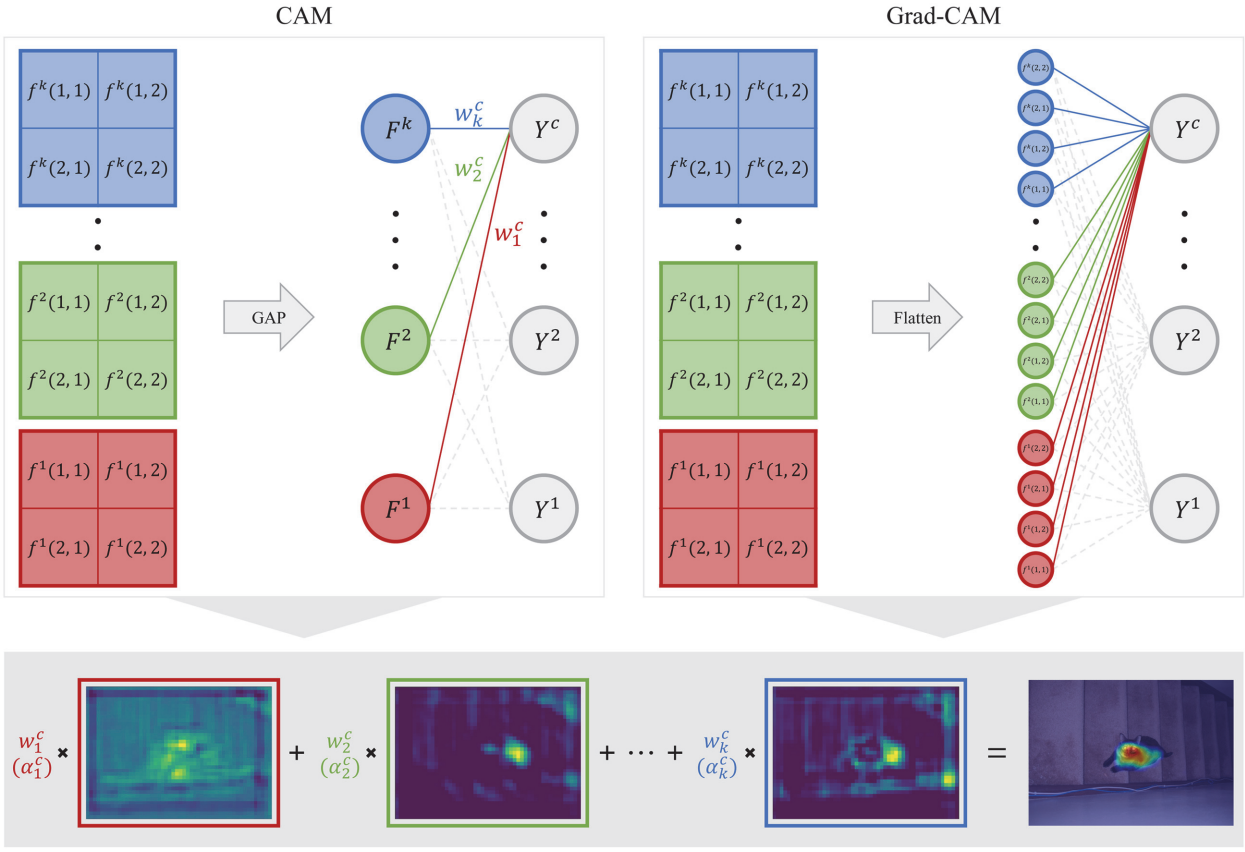

Fig. 2. Comparison of CAM and Grad-CAM.

convolutional maps immediately prior to prediction [8].

Selvaraju et al. [8] replaced the weight w_k^c with α_k^c, which is the gradient of the class score with respect to the feature map. They also mathematically proved that the two expressions were identical. Therefore, Grad-CAM could be generated without adding GAP or learning new weights. Grad-CAM is defined as

$$\text{Grad-CAM}^c(i,j) = \text{ReLU}\left(\sum_k \alpha_k^c f^k(i,j)\right), \qquad (7)$$

where $\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial f^k(i,j)}$.

III. DRAWBACKS OF FPN

Lin et al. [7] showed the effects of the top-down pathway on the detection performance of the FPN. The baseline model achieved an AP of 33.9% in the COCO minival dataset, whereas the model without the top-down pathway achieved an AP of 24.9% [7]. The results showed that removing the top-down pathway significantly reduced the detection performance. In other words, the top-down pathway enriched the visual semantics of the low-level features.

We designed an experiment to further verify that all feature levels had rich semantics when the top-down pathway was adopted. We assumed that if all of the features had rich semantics, then the detection performance should be improved as the features enlarge. Additionally, we aimed to determine whether the large-scale features had an advantage over the small ones in terms of the detection performance on the small-sized objects, and vice versa.

We implemented seven variants of the FPN. Lv6, shown in Table 1, used the features extracted from the sixth

Table 1. Detection results of the models with and without the top-down pathway

Model	AP	AP@0.5	AP@0.75	AP_S	AP_M	AP_L
Lv6	30.91	63.08	25.98	0.93	12.26	36.41
Lv5	35.60	68.01	33.01	4.82	20.18	39.76
Lv4	**37.48**	**68.64**	36.29	**9.66**	**22.62**	**40.41**
Lv3	36.84	65.39	**36.86**	8.61	19.11	40.26
Lv5-	**34.68**	**67.86**	**31.28**	5.01	**20.65**	**37.83**
Lv4-	31.27	62.19	27.64	**6.29**	18.30	32.79
Lv3-	17.79	40.07	13.20	3.44	5.06	19.98

The highest AP of each column is shown in bold.

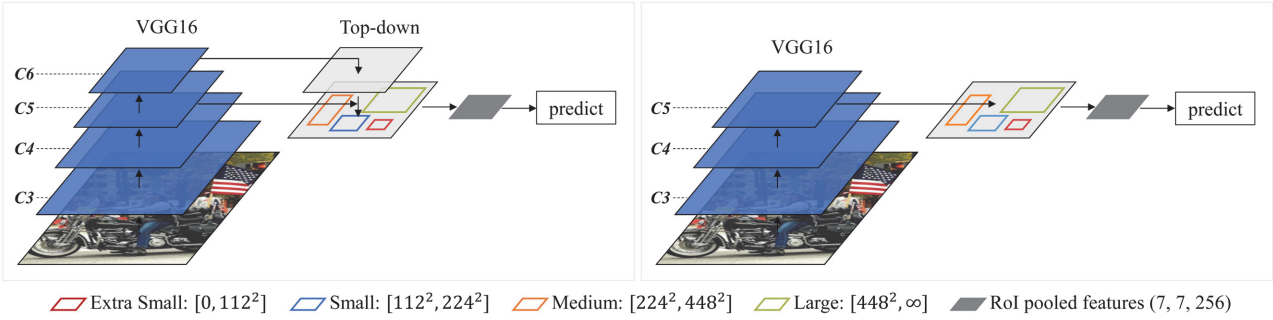

Fig. 3. Examples of the two models used in the experiment. Left: Lv5. Right: Lv5 without the top-down pathway.

convolutional layer and then passed through the lateral connection. Lv5 used a combination of two features from the fifth and sixth layers as shown on the left of Fig. 3. The scale of the latter was doubled before combining with the former. Lv4 and Lv3 used all of the features combined from the sixth layer to the layer corresponding to its name. The last three models (Lv5-, Lv4-, and Lv3-) were the same as the aforementioned models except that the top-down pathway was removed as shown on the right of Fig. 3.

A. Implementation Details

The models used in this experiment adopted the pre-trained VGG16 as a backbone network. The two fully connected layers, fc6 and fc7, were converted into convolutional layers [9, 19] to reduce the number of parameters. Anchor boxes outside an image were included for training, as it was in the original FPN [7]. The input image was re-scaled such that its shorter side had 600 pixels. The anchor scales of the region proposal network (RPN) [3] were $[64^2, 128^2, 256^2, 512^2]$, and the aspect ratios were [0.5, 1, 2]. The batch size, weight decay, and momentum were 1, 0.0005, and 0.9, respectively. The learning rate was 0.001 for the first 50k images, and 0.0001 for the next 25k. Lastly, the PASCAL VOC 2007 trainval dataset was used for training.

B. Single-Scale RoI Pooling

As shown in Table 1, the AP and AP@0.5 (the AP with an IoU threshold of 0.5) were increased when the top-down pathway was applied. Specifically, the discrepancies in AP and AP@0.5 between the models with the top-down pathway and those without the top-down pathway increased as the feature levels decreased. This suggests that the top-down pathway can fill a large semantic gap effectively. We also observed this effect in the Grad-CAM of each model shown in Fig. 4. The features of Lv3-, the model without the top-down pathway, showed characteristics of the low-level features (e.g., edges). In contrast, the features of Lv3 were semantically more meaningful and similar to the high-level features.

Although the top-down pathway adds semantics to the low-level features, it does not add rich semantics to all feature levels. As Table 1 shows, the AP and AP@0.5 were progressively improved as the feature scale increased up to level 4. However, they decreased by 0.64 and 2.71 points, respectively, when the scale exceeded level 4. This suggests that the semantic gaps between Lv4 and Lv3 features are not entirely filled by the top-down pathway. Therefore, pooling small RoIs only from Lv3 features could degrade the detection performance.

As shown in Table 1, the AP of Lv3 was decreased by 1.05 points compared to that of Lv4, which indicates that

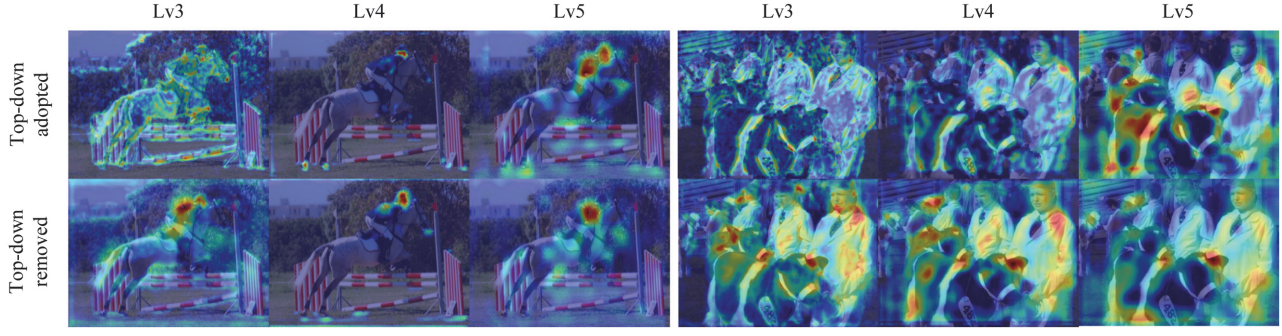

Fig. 4. Grad-CAM of the models used in the experiment. Row 1: the models with the top-down pathway removed. Row 2: the models with the top-down pathway adopted.

the large-scale features were not necessarily advantageous for detecting small objects. This also stems from the semantic gap between the two features. Specifically, the semantic loss was greater than the spatial information gain when the level was decreased (i.e., when the scale was increased) from 4 to 3. Meanwhile, the AP_L of Lv6 was decreased by 3.35 points compared to that of Lv5, which suggests that small-scale features are not necessarily advantageous for detecting large objects.

In summary, the top-down pathway contributed to improving the detection performance by enriching the visual semantics of the low-level features. However, the top-down pathway did not enrich the visual semantics of the low-level features as much as those of the high-level features. In other words, the semantic gap between the features remained. Lastly, large-scale features were not necessarily advantageous for detecting small objects, and vice versa. Therefore, assigning single-scale features to the RoI depending upon their area could inhibit the model from improving the detection performance.

C. Overfitting to Certain Features

FPN computes the level of the features as

$$k = \left\lfloor k_0 + \log_2(\sqrt{wh}/224) \right\rfloor, \quad (8)$$

where w and h are the horizontal and vertical lengths of the RoI, and k_0 is the level of features that should be assigned to the RoI with an area of 224^2. We set $k_0 = 5$. If we calculate the level k for all possible areas of the RoI, we can obtain four ranges of the RoI area to which the level k features should be assigned. Specifically, level 3 features are assigned to the RoI with an area $[0, 112^2)$, level 4 to $[112^2, 224^2)$, level 5 to $[224^2, 448^2)$, and level 6 to $[448^2, \infty]$. The problem was that each range had a different number of objects depending upon the dataset. For example, the PASCAL VOC 2007 trainval dataset had more objects in the second range and the third range than in the rest of the ranges, as shown in Table 2. The COCO 2017 training dataset was more imbalanced in terms of the size of the objects. It had more than 60% level 3 objects, but less than 10% level 6 objects. If the dataset has a significantly higher number of small objects than the large ones, the FPN will generate an excessive number of small RoIs. Then the model could be overfitted to large-scale features since it assigns large-scale features to small RoIs.

Furthermore, the number of objects in each range changes as the input image is re-scaled. For example, if we increase the scale of the input image such that its shorter side has 800 pixels, the number of large objects increases, and that of small objects decreases, as shown in Table 2. The imbalance in size could be handled by researchers prior to training the model. However, considering that this is a time-consuming job, we need a better way to use each feature more evenly.

IV. PROPOSED METHOD

We proposed FPN+, which is an FPN with multi-scale RoI pooling. Multi-scale pooling pools RoIs from multiple feature levels and concatenates the pooled features into a single block. Since the FPN only uses single-level features, three additional levels of features are available for RoI pooling. Therefore, we conducted experiments to determine the number of features most effective for multi-scale RoI pooling. We implemented three variants of FPN+. The first variant used two levels of features, the second variant used three levels of features, and the third variant used four levels of features. All models were trained using the PASCAL VOC 2007 trainval dataset.

As shown in Table 3, the model using three feature levels scored the highest on all metrics. It is noteworthy that the AP_S was decreased by 4.52 points when using

Table 2. Classification of objects by area

Input size (px)	Dataset	$[0,112^2)$	$[112^2, 224^2)$	$[224^2, 448^2)$	$[448^2, \infty)$
600	VOC 2007 trainval	24.28	27.04	31.83	16.85
	COCO 2017 train	61.72	19.58	13.55	5.16
800	VOC 2007 trainval	15.13	24.27	30.73	29.88
	COCO 2017 train	52.84	21.25	16.05	9.86

Table 3. Experimental results to determine the number of features for multi-scale RoI pooling

Number of features	AP	AP@0.5	AP@0.75	AP_S	AP_M	AP_L
2	38.28	70.09	36.71	14.26	23.61	40.43
3	**39.09**	**71.18**	**37.89**	**16.21**	**24.65**	**41.25**
4	38.64	70.52	37.66	11.69	23.96	41.21

The highest AP of each column is shown in bold.

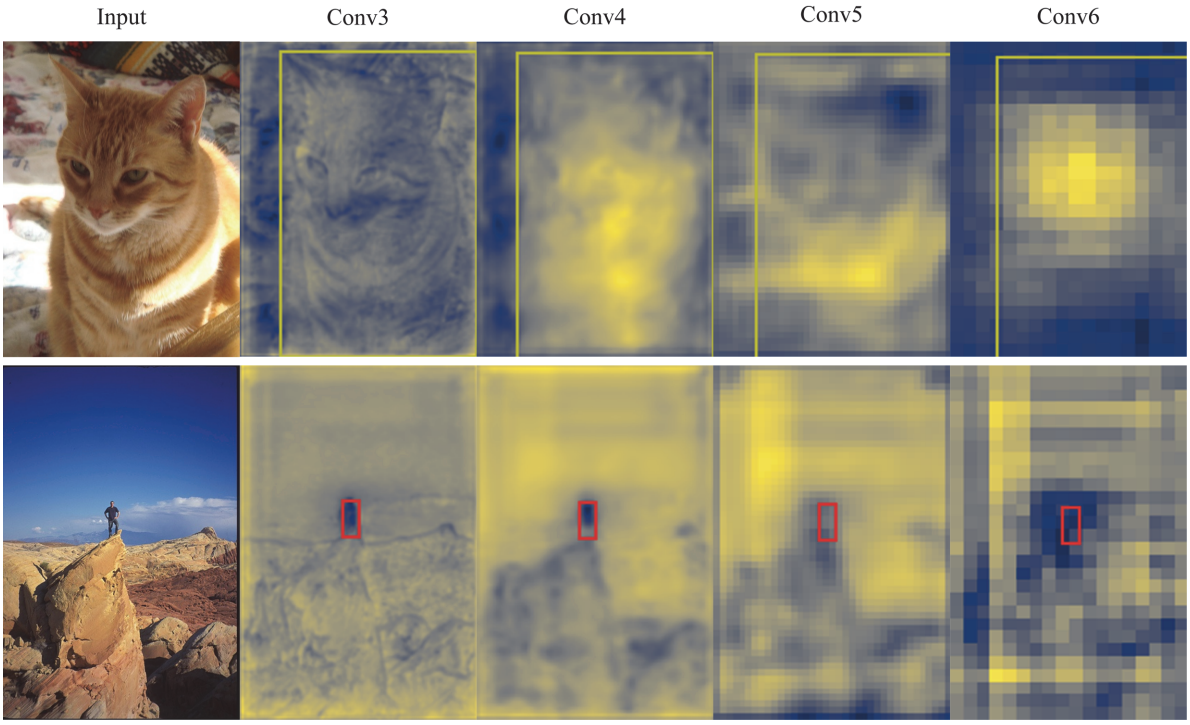

Fig. 5. Feature maps of FPN. Top: an example of a large-sized object. Bottom: an example of a small-sized object.

four levels of features instead of three. The addition of level 6 features did not improve the detection of small objects. This is because level 6 features may not contain the spatial information of small objects. In other words, since receptive fields become larger as the feature levels increase [20], the regions corresponding to the small RoIs in the Conv6 features may contain unnecessary contextual information, as shown in the bottom row of Fig. 5.

An example of multi-scale RoI pooling is shown in Fig. 1, where five RoIs with sizes of [50^2, 50^2, 100^2, 300^2, 500^2], and feature levels were computed as [3, 3, 3, 5, 6] by Eq. (8). Adding two adjacent levels to each of them gave final levels of [[3, 4, 5], [3, 4, 5], [3, 4, 5], [4, 5, 6], [4, 5, 6]]. RoI pooling was performed on each of the three features. As a result, an output with a size of 256×7×7 was obtained from each feature, and concatenating them gave a single block of size 768×7×7. The size of the final output was 6×768×7×7 since there were six RoIs.

V. EXPERIMENTAL RESULTS

A. Implementation Details

We tested three models for this experiment: the Faster R-CNN, FPN, and FPN+. FPN+ is a model in which multi-scale RoI pooling is applied. The Faster R-CNN was tested in the same environment as the original method [3] except for the anchor scale. We used four anchor scales [64^2, 128^2, 256^2, 512^2] to ensure a fair comparison with the other two models. The input size, batch size, weight decay, and momentum were 600, 1, 0.0005, and 0.9, respectively. The learning rate was 0.001 for the first 50k images and 0.0001 for the next 25k for the PASCAL VOC dataset; and 0.001 for the first 351k images and 0.0001 for the next 234k for the COCO dataset. We excluded 1,021 images with no annotations from 118,287 images in the COCO 2017 training dataset. For the same reason, 48 images were excluded from 5,000 images in the validation dataset

B. Comparison of Faster R-CNN and FPN

As shown in Table 4, the FPN was 0.93 points worse than the Faster R-CNN in AP@0.5 when the models were trained on the PASCAL VOC 2007 trainval dataset. However, the FPN exceeded the Faster R-CNN by 1.18 and 3.22 points in the AP and AP@0.75, respectively. This suggests that the FPN localized objects more accurately than the Faster R-CNN. The most noticeable difference between the two models was the detection performance for small objects. The FPN AP_S improved by 13.47 points compared to the Faster R-CNN, because the FPN uses low-level features. Meanwhile, the Faster R-CNN AP_L was better than that of the FPN by 0.74 points, which indicates that the performance of the former was slightly better than that of the latter for large objects. We observed similar results when the models were trained on the COCO 2017 training dataset, as shown in Table 5. The precision of the FPN was higher

Table 4. Detection results of the Faster R-CNN, FPN, and FPN+ in the PASCAL VOC 2007 test dataset

Dataset	Model	AP	AP@0.5	AP@0.75	AP_S	AP_M	AP_L
VOC 07 trainval	Faster-R-CNN	35.10	69.85	30.71	4.59	20.77	38.62
	FPN	36.28	68.92	33.93	**18.06**	22.99	37.88
	FPN+	**39.09**	**71.18**	**37.89**	16.21	**24.65**	**41.25**
VOC 07+12 trainval	Faster-R-CNN	42.73	75.12	42.49	7.79	26.47	46.50
	FPN	43.69	75.43	44.79	**18.32**	28.99	45.49
	FPN+	**45.35**	**76.03**	**47.30**	18.04	**30.69**	**47.51**

The highest AP of each column is shown in bold.

Table 5. Detection results of the Faster R-CNN, FPN, and FPN+ in the COCO 2017 validation dataset

Model	AP	AP@0.5	AP@0.75	AP_S	AP_M	AP_L
Faster-R-CNN	19.4	38.1	17.99	5.0	21.4	32.1
FPN	22.4	40.5	22.9	10.9	25.9	30.0
FPN+	**23.5**	**41.6**	**24.0**	**11.0**	**26.5**	**32.5**

than that of the Faster R-CNN except for AP_L. The FPN AP@0.75 improved by 4.91 points compared to that of the Faster R-CNN.

C. Comparison of FPN and FPN+

FPN+ was 2.81 and 3.96 points better than the FPN in AP and AP@0.75, respectively, as shown in Table 4. However, FPN+ scored less than the FPN by 1.85 points in APs. We obtained similar results when the models were trained on the combined datasets of PASCAL VOC 2007 and 2012 trainval. The APs of FPN+ was 0.24 points lower than that of the FPN. The additional contextual information of high-level features may not be helpful for detecting small objects. In contrast, FPN+ scored the highest in AP_L among the three models. This suggests that the additional local information on low-level features is helpful for detecting large objects. As shown in the top row of Fig. 5, the regions corresponding to the RoI in Conv3 and Conv4 features contained detailed information. Additionally, multi-scale RoI pooling allowed more feature levels to contribute to the detection of large objects, as shown in Fig. 6. Training the models on the COCO 2017 training dataset gave similar results. FPN+ surpassed the others in all metrics. The detection results of FPN+ are shown in Fig. 7.

We replaced the backbone network of the FPN and FPN+ with ResNet101 and trained them on the PASCAL VOC 2007 trainval dataset. This test showed that the effectiveness of multi-scale RoI pooling is not limited to

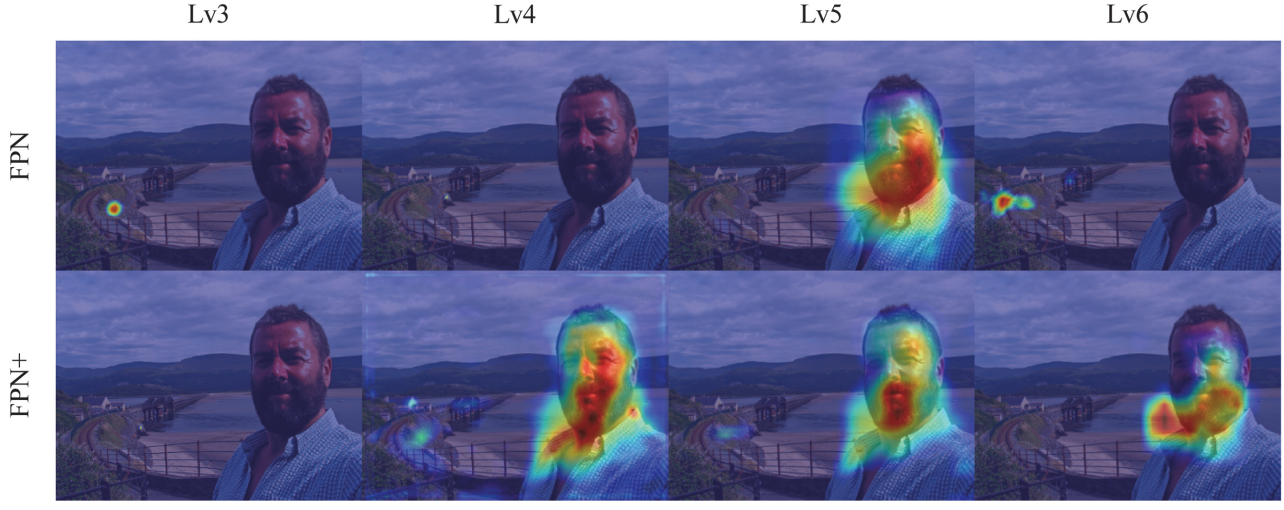

Fig. 6. Grad-CAM of FPN and FPN+.

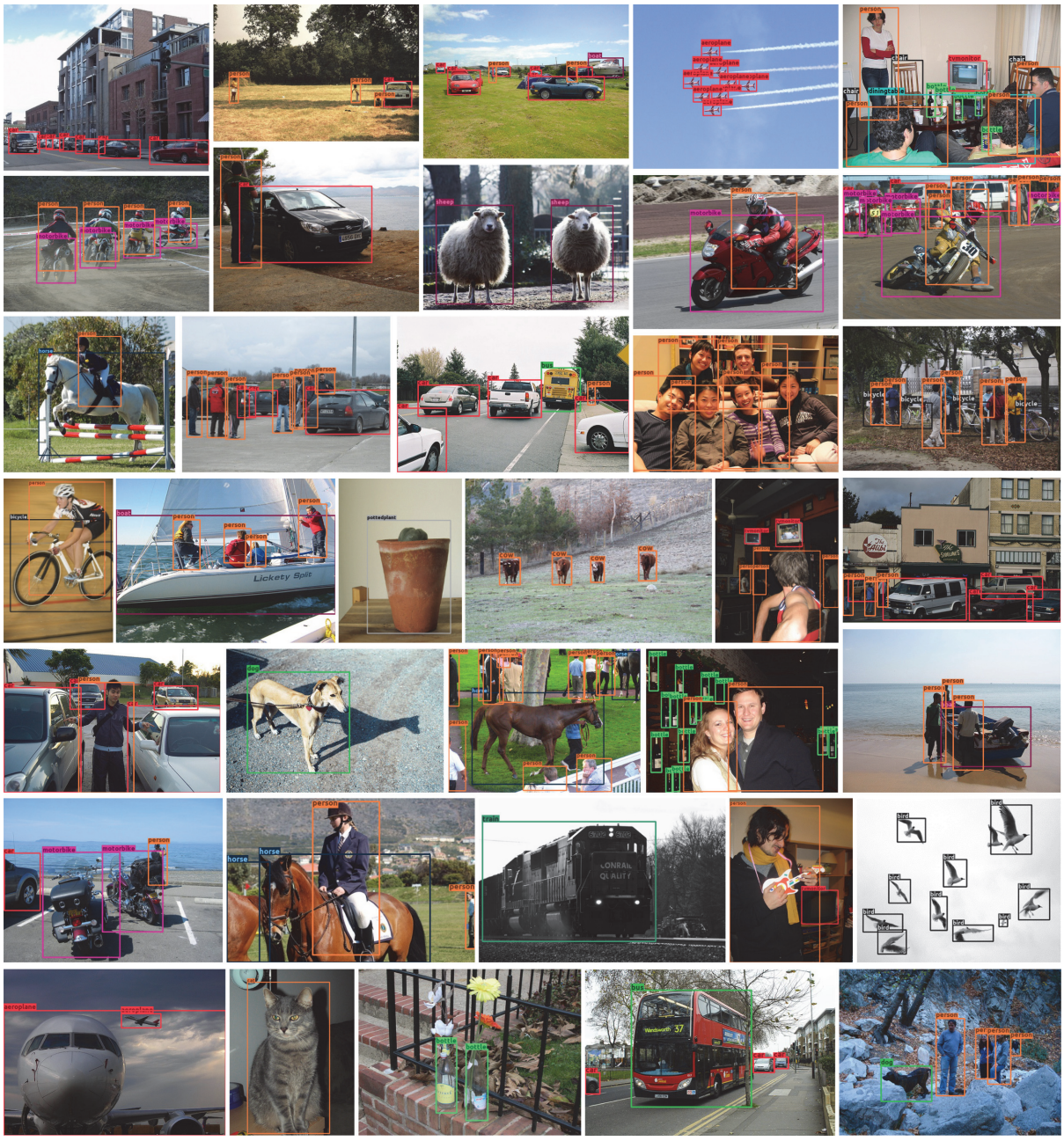

Fig. 7. Examples of detection results on PASCAL VOC 2007 test dataset using FPN+. The backbone network of the model is VGG16, the score threshold is 06, and the model was trained on a combined dataset of VOC 2007 and 2012 datasets.

Table 6. Detection results of the FPN and FPN+ (ResNet101 as a backbone network) in the PASCAL VOC 2007 test dataset

Model	AP	AP@0.5	AP@0.75	AP_S	AP_M	AP_L
FPN	42.09	73.52	43.80	**19.46**	49.78	77.45
FPN+	**44.98**	**75.14**	**47.16**	19.26	**53.49**	**78.04**

a particular network. As shown in Table 6, FPN+ scored the highest in all metrics except APs.

D. Comparison of FPN+ with SDPNet and HyperNet

We compared FPN+ to the SDPNet and HyperNet. MS-CNN [12] was not included in the comparison because the authors did not present detection results for the PASCAL VOC dataset.

As shown in Table 7, FPN+ scored the highest in AP@0.5 among the four models when trained on the PASCAL VOC 2007 trainval dataset. However, HyperNet

Table 7. Detection results of the SDPNet, HyperNet, Faster R-CNN, FPN, and FPN+ in the PASCAL VOC 2007 test dataset

Dataset	SDPNet	HyperNet	Faster R-CNN	FPN	FPN+
VOC 07 trainval	69.4	-	69.8	68.9	**71.1**
VOC 07+12 trainval	-	**76.3**	75.1	75.4	76.0

The highest AP of each row is shown in bold. The detection results of SDPNet and HyperNet are cited from [11] and [13].

outperformed FPN+ by 0.3 points when the models were trained on the combined PASCAL VOC 2007 and 2012 trainval datasets. Although the two models were equally trained on the PASCAL VOC dataset, hyperparameters such as learning rate, epoch, and the threshold for non-maximum suppression were different. Given that the difference between the two models in AP@0.5 was marginal, the performance gap could be due to different hyperparameter settings.

The biggest advantage of FPN+ is that it can localize objects accurately. All APs (average precisions) in Table 7 were calculated for an IOU threshold of 0.5. A bounding box with an IOU greater than 50% was considered a true positive. However, an IOU threshold of 0.5 was too loose to fit the ground truth object precisely. This is why COCO-style AP uses 10 IOU thresholds of 0.5 to 0.95. Averaging APs for 10 different IOU thresholds rewards the detectors with better localization. Therefore, if we calculate the AP for a higher threshold, FPN+ could outperform HyperNet.

E. Applicable Fields

The sizes of objects vary in high spatial resolution (HSR) images for remote sensing, and tiny size objects can be present, especially in aerial images [21]. The visual appearance may be diminished by poor weather and illumination conditions [22]. Therefore, the task of detecting HSR images is more challenging. FPN can widely be used in this field because it can accurately localize and detect objects of various sizes. For example, FPN is used for general object detection [23, 24], ship detection [25], land segmentation [26], and road segmentation [27].

VI. CONCLUSION

In this paper, we presented two minor drawbacks of the FPN. First, the FPN uses only single-scale features, which could hinder the model from improving localization accuracy. Second, if a training dataset was imbalanced in object size, the FPN could overfit to a certain feature level. We proposed multi-scale RoI pooling to overcome the drawbacks of the FPN. The experimental results showed that FPN+, the FPN with multi-scale RoI pooling, outperformed the FPN in AP when tested on PASCAL VOC 2007 test and COCO 2017 validation datasets. Specifically, the localization accuracy was greatly improved. Our study suggests that even though the FPN makes low-level features semantically more meaningful, it is still important to consider all information at different feature levels to detect objects accurately.

CONFLICT-OF-INTEREST STATEMENT

The authors declare that there is no conflict of interest.

ACKNOWLEDGMENTS

This research was supported by the Ministry of Science and ICT (MIST), Korea, under the National Program for Excellence in SW supervised by the Institute for Information & Communications Technology Promotion (IITP) (No. 2017-0-00093).

REFERENCES

1. 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, OH, 2014, pp. 580-587.
2. R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 1440-1448.
3. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.
4. 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, NV, 2016, pp. 779-788.
5. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
6. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of 2005 IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, 2005, pp. 886-893.
7. T. Y. Lin, P. Dollar, 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, HI, 2017, pp. 936-944.

8. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 618-626.
9. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Computer Vision – ECCV 2016*. Cham, Switzerland: Springer, 2016, pp. 21-37.
10. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014 [Online]. Available: https://arxiv.org/abs/1409.1556.
11. F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 2129-2137.
12. Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Computer Vision – ECCV 2016*. Cham, Switzerland: Springer, 2016, pp. 354-370.
13. T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: towards accurate region proposal generation and joint object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 845-853.
14. K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: visualising image classification models and saliency maps," 2014 [Online]. Available: https://arxiv.org/abs/1312.6034.
15. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV*. Cham, Switzerland: Springer, 2014, pp. 818-833.
16. J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: the all convolutional net," 2014 [Online]. Available: https://arxiv.org/abs/1412.6806.
17. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 2921-2929.
18. M. Lin, Q. Chen, and S. Yan, "Network in network," 2014 [Online]. Available: https://arxiv.org/abs/1312.4400.
19. L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFS," 2016 [Online]. Available: https://arxiv.org/abs/1412.7062.
20. W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 29, pp. 4898-4906, 2016.
21. H. Tayara and K. T. Chong, "Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network," *Sensors*, vol. 18, no. 10, article no. 3341, 2018. https://doi.org/10.3390/s18103341
22. X. Han, Y. Zhong, and L. Zhang, "An efficient and robust integrated geospatial object detection framework for high spatial resolution remote sensing imagery," *Remote Sensing*, vol. 9, no. 7, article no. 666, 2017. https://doi.org/10.3390/rs9070666
23. X. Zhang, K. Zhu, G. Chen, X. Tan, L. Zhang, F. Dai, P. Liao, and Y. Gong, "Geospatial object detection on high resolution remote sensing imagery based on double multi-scale feature pyramid network," *Remote Sensing*, vol. 11, no. 7, article no. 755, 2019.
24. Y. Li, Q. Huang, X. Pei, L. Jiao, and R. Shang, "RADet: refine feature pyramid network and multi-layer attention network for arbitrary-oriented object detection of remote sensing images," *Remote Sensing*, vol. 12, no. 3, article no. 389, 2020.
25. X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, and Z. Guo, "Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks," *Remote Sensing*, vol. 10, no. 1, article no. 132, 2018. https://doi.org/10.3390/rs10010132
26. S. Seferbekov, V. Iglovikov, A. Buslaev, and A. Shvets, "Feature pyramid network for multi-class land segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 272-275.
27. X. Gao, X. Sun, Y. Zhang, M. Yan, G. Xu, H. Sun, J. Jiao, and K. Fu, "An end-to-end neural network for road extraction from remote sensing imagery by multiple feature pyramid network," *IEEE Access*, vol. 6, pp. 39401-39414, 2018.

Seungtae Nam

Seungtae Nam received B.S. degrees in Software Convergence and Hospitality Management from Kyung Hee University, Korea, in 2021. He is pursuing M.S. degree in Artificial Intelligence at Sungkyunkwan University, Korea. His research interests include object detection, image classification, neural fields, meta learning.

Daeho Lee https://orcid.org/0000-0003-2313-7483

Daeho Lee received B.S., M.S., and Ph.D. degrees in Electronic Engineering from Kyung Hee University, Korea, in 1998, 2001, and 2005, respectively. He is a professor in the Department of Software Convergence, Kyung Hee University, Republic of Korea from July 2019. He was an assistant professor in Faculty of General Education at Kyung Hee University from September 2005 to February 2011, and was a professor in Humanitas College at Kyung Hee University from March 2011 to June 2019. His research interests include computer vision, image processing, pattern recognition, machine learning, deep learning, computer games, augmented reality, and human-computer interaction.