Hindawi
Scientific Programming
Volume 2020, Article ID 8839979, 8 pages
https://doi.org/10.1155/2020/8839979

Research Article

Scale Adaptive Feature Pyramid Networks for 2D Object Detection

Lifei He,[1] Ming Jiang,[1] Ryutarou Ohbuchi ,[2] Takahiko Furuya,[2] Min Zhang,[1] and Pengfei Li[1]

[1]*School of Computer Science, Hangzhou Dianzi University, Hangzhou 310000, China*
[2]*Department of Computer Science and Engineering, University of Yamanashi, 4-3-11 Takeda, Kofu-shi, Yamanashi-ken 400-8511, Japan*

Correspondence should be addressed to Ryutarou Ohbuchi; ohbuchi@yamanashi.ac.jp

Received 2 August 2020; Accepted 10 December 2020; Published 22 December 2020

Academic Editor: Pengwei Wang

Object detection is one of the core tasks in computer vision. Object detection algorithms often have difficulty detecting objects with diverse scales, especially those with smaller scales. To cope with this issue, Lin et al. proposed feature pyramid networks (FPNs), which aim for a feature pyramid with higher semantic content at every scale level. The FPN consists of a bottom-up pyramid and a top-down pyramid. The bottom-up pyramid is induced by a convolutional neural network as its layers of feature maps. The top-down pyramid is formed by progressive up-sampling of a highly semantic yet low-resolution feature map at the top of the bottom-up pyramid. At each up-sampling step, feature maps of the bottom-up pyramid are fused with the top-down pyramid to produce highly semantic yet high-resolution feature maps in the top-down pyramid. Despite significant improvement, the FPN still misses small-scale objects. To further improve the detection of small-scale objects, this paper proposes scale adaptive feature pyramid networks (SAFPNs). The SAFPN employs weights chosen adaptively to each input image in fusing feature maps of the bottom-up pyramid and top-down pyramid. Scale adaptive weights are computed by using a scale attention module built into the feature map fusion computation. The scale attention module is trained end-to-end to adapt to the scale of objects contained in images of the training dataset. Experimental evaluation, using both the 2-stage detector faster R-CNN and 1-stage detector RetinaNet, demonstrated the proposed approach's effectiveness.

1. Introduction

Object detection is one of the most important problems in computer vision. In recent years, object detection accuracy has improved greatly by using deep convolution neural network (CNN) [1]. Object detection algorithms based on CNN are commonly classified into two-stage detector and one-stage detector. Representative two-stage detectors include R-CNN [2], SPP-net [3], fast R-CNN [4], and faster R-CNN [5]. Correspondingly, representative single-stage detectors include SSD [6], several incarnations of YOLO ([7–9]), and RetinaNet [10].

Most of these methods use a DCNN such as VGG [1] or ResNet [12] as their "backbone" network to extract features to detect object area, or bounding box, and to classify object in the bounding box. Some object detection algorithms, such as faster R-CNN, uses a feature map at a fixed resolution level for the task. Others, such as RetinaNet, use a hierarchy of feature maps having different resolutions. The single resolution feature map of the faster R-CNN (e.g., C_5 in Figure 1(a)) has high semantic content induced by later CNN layers of the backbone network. However, its resolution is low, so a small object in the input image may be missed (in Figure 1, thin and thick borders of the feature maps indicate their low and high semantic content, respectively). One could use earlier layers of the backbone CNN as feature maps for object area detection and classification. However, these earlier layers of feature maps (e.g., C_1 or C_2 in Figure 1(a)) have lower semantic content while having higher resolution. This leads to inaccurate object area (bounding box) and inaccurate classification labels of objects in the area.

In recent years, the multilayer feature map has been proposed to deal with this issue, significantly improving the accuracy of small object detection (e.g., SSD, FPN, and RetinaNet). For example, SSD combines a multiscale set of

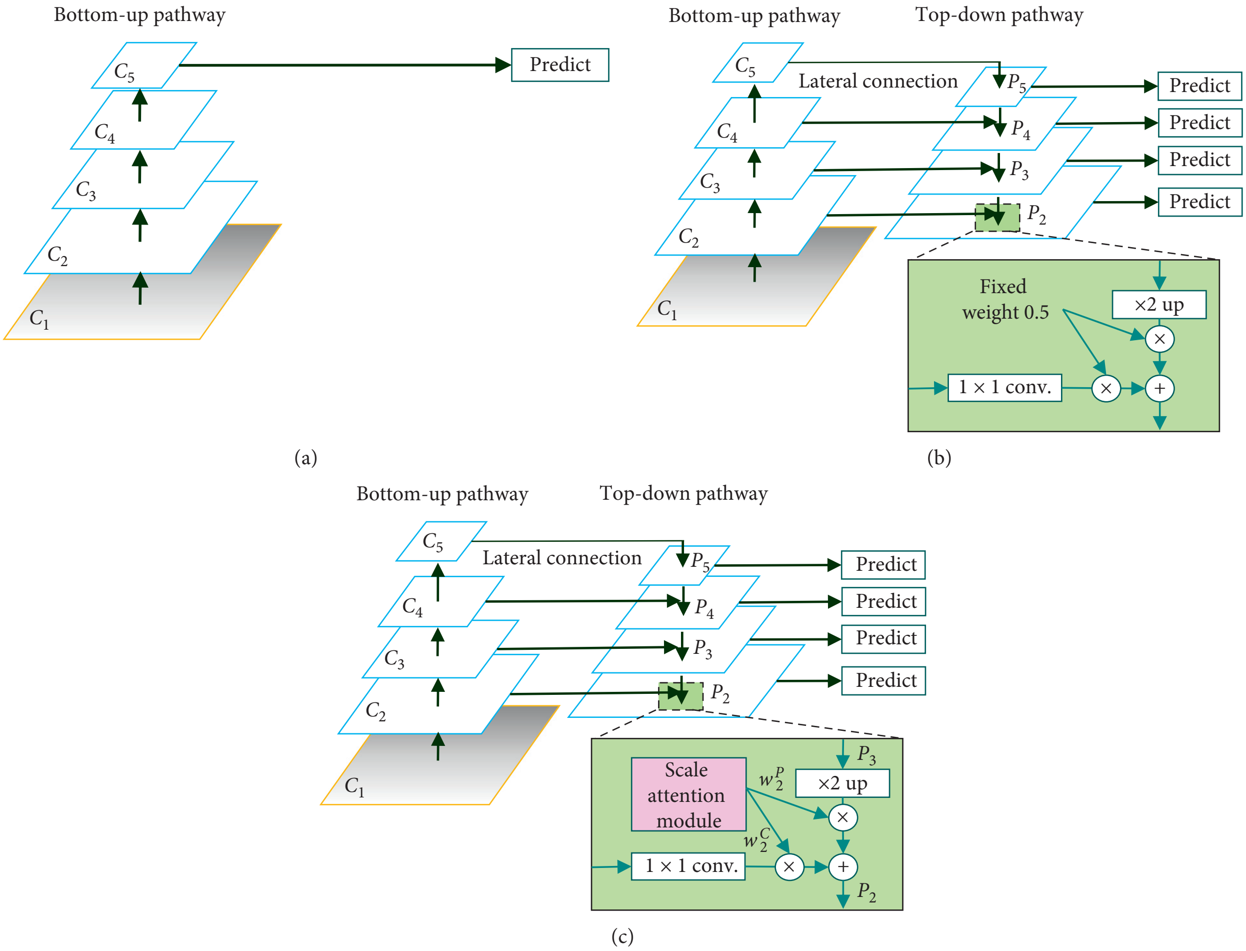


Figure 1: Feature maps used for region proposal and object class prediction in the faster R-CNN [5] (a), feature pyramid networks (FPNs) [11] (b), and the proposed scale adaptive FPN (SAFPN) (c). (a) The faster R-CNN [5] predicts object bounding box based on a highly semantic yet low-resolution feature map. Due to low resolution of the feature map, mall objects tend to be missed. (b) The feature pyramid network (FPN) [11] predicts object bounding box based on multiresolution, highly semantic feature maps formed by fusion feature maps of top-down and bottom-up pathways. Adjacent scale feature maps are integrated by using fixed weights 0.5. RetinaNet [10] also uses this FPN. (c) The proposed scale adaptive FPN (SAFPN) integrates feature maps by using weights computed, at each resolution level, by using the scale attention module (SAM) to suit scales of objects in each input image.

feature maps to localize and identify objects. However, its ability to detect smaller objects is still not satisfactory. It is also computationally expensive due in part to its choice of the base network, VGG.

Lin et al. [11] added the feature pyramid network (FPN) to faster R-CNN for efficient yet accurate detection of objects having varying scales. The FPN, illustrated in Figure 1(b), consists of a pair of multiscale pyramids of feature maps. The bottom-up pathway is a pyramid induced by the backbone CNN, starting from the highest resolution yet least semantic input image. The bottom-up pyramid produces a highly semantic yet lowest resolution feature map (C_5 in Figure 1(b)) at the top of the pyramid. The top-down pyramid starts with the low resolution yet highly semantic image handed over from the bottom-up pyramid (P_5 in Figure 1(b)) and is generated by repeated up-sampling. The top-down pyramid feature maps are enriched by fusing information passed on from feature maps of the bottom-up pathway via the lateral connections. The feature maps of the top-down pyramid, which simultaneously have high semantic content as well as high resolution, are used for object area detection and classification.

Several object detection network architectures appeared since incorporated FPN. For example, a single-stage detector RetinaNet combines the FPN with a modern backbone network and a loss function called focal loss to achieve high speed as well as high accuracy. The accuracy of RetinaNet is comparable to the two-stage detector faster R-CNN. Focal loss is introduced to the RetinaNet training to alleviate positive/negative sample imbalance.

In the FPN, integration of information from the bottom-up pathway via lateral connection and information from the top-down pathway is done by using fixed-weight summation, as illustrated in Figure 1(b). However, the optimal ratio of the fusion depends on the size of objects in the input image. Obviously, the sizes of objects vary from image to image, and distribution in size of objects in images depends on the dataset. Fixed weights used in the FPN to integrate bottom-up and top-down feature maps, which amounts to an even weight average of the two, are most likely suboptimal.

In this paper, to improve both object localization and classification accuracy, we propose integrating feature maps of the bottom-up pathway and top-down pathway of the FPN using scale adaptive weights computed per input image. In the proposed approach, called scale adaptive FPN (SAFPN), the weights for feature map integration are learned from the images in the training dataset and computed per input image at inference time. The SAFPN may be viewed as an attention mechanism over image object scales and semantic levels. We evaluate our proposed approach's effectiveness by applying the proposed SAFPN to two object detection architectures, one a two-stage detector faster R-CNN and the other a one-stage detector RetinaNet.

Our contributions can be summarized as follows:

(1) Proposal of the scale adaptive feature pyramid network (SAFPN): a novel method to fuse feature maps of the bottom-up pyramid and top-down pyramid of the FPN based on input image. The weights for fusion are computed per input image at each resolution level by the scale attention module (SAM). The SAM is trained end-to-end along with the other parts of the object detection network.

(2) Experimental evaluation of the SAFPN: experimental evaluation of the proposed SAFPN using two representative object detection networks, a one-stage detector and a two-stage detector. The evaluation using PASCAL VOC 2007 and PASCAL VOC 2007 + 2012 as training datasets showed that the proposed SAFPN significantly improves object detection accuracies in both types of detectors.

The rest of this paper is organized as follows. In the next section, we review relevant work. In Section 3, we describe the proposed method, followed by its experimental evaluation in Section 4. Conclusion and future work will be presented in Section 5.

2. Related Work

2.1. Object Detection. We review some of the previous methods of object detection in this section. In 2013, Girshick et al. proposed the R-CNN [2], which employs a region proposal. While it uses the CNN, the R-CNN's overall architecture is influenced by a traditional shallow object detection approach. A set of object region proposals are extracted by the classical selective search [11] approach. Then, each region proposal is fed to a CNN after the region is rescaled to a prescribed size to extract features for the region. The R-CNN used Alexnet by Kryzhevsky et al. [13] trained on the ImageNet dataset for feature extraction. The feature extracted by the CNN is passed on to a linear support vector machine (SVM) to determine the object class.

In 2014, He et al. proposed spatial pyramid pooling networks (SPP-nets) [3] to handle objects in images having arbitrary size/scale. The SPP layer is located after the last convolution layer, right before the fully connected layers of a standard CNN pipeline. The SPP-net reduces the costs of detecting large objects using pyramidal pooling.

In 2015, Girshick proposed the fast R-CNN [4], which is an improvement over the R-CNN and SPP-net. The fast R-CNN employs single CNN pipeline for both object bounding box regression and object classification. The CNN is trained simultaneously for both box regression and object classification objectives. This and other improvements brought significant speedup over the predecessor R-CNN. Note that region of interest (ROI) pooling in the fast R-CNN can be considered a special SPP case.

Also in 2015, shortly after the fast R-CNN [5], Ren et al. proposed the faster R-CNN. The faster R-CNN is the first DNN-based object detector trained end-to-end. It is also the first DNN-based object detector to perform at near real-time speed. The most important innovation of the faster R-CNN is its region proposal network (PRN) that proposes bounding boxes having high objectness. By sharing most of its processing with the main object detection network, the faster R-CNN is much more efficient than the fast R-CNN. The faster R-CNN uses a single resolution feature map for region proposal and object classification (Figure 1(a)). As the feature map of a latter layer of a standard CNN is highly semantic yet of low resolution, bounding box region regression and object classification accuracy are limited.

Lin et al. proposed, in 2017, the feature pyramid network (FPN) [11] to faster R-CNN. The idea of the feature pyramid had been popular during pre-DNN due to its ability to perform multiscale processing of images, e.g., for object recognition. However, it went out of favor in the early years of the DNN for its computational cost. The FPN couples a bottom-up pyramid inherent in a CNN with a top-down pyramid that performs up-sampling and deconvolution. Semantic information trickles down the top-down pathway from the small (low resolution) yet highly semantic feature map to the high-resolution and highly semantic feature map. The two pyramids are connected by lateral connections to pass on high-resolution information to the top-down network from the bottom-up network. The FPN has been adopted by other object detection networks, most notably a one-stage detector RetinaNet [10] by Lin et al.

Traditionally, a two-stage detector held an advantage in accuracy over a one-stage detector. However, RetinaNet, despite being a one-stage detector, achieved accuracy comparable to the two-stage detector faster R-CNN. RetinaNet combines the FPN with a new loss function called focal loss with an improved backbone network. Focal loss alleviates issues associated with an unbalanced number of samples between the object region (foreground) and its background.

The method proposed in this paper improves upon the FPN by adding a scale-space attention mechanism to adaptively compute, for each input image, weights for feature map integration. Given an input image, a trained scale attention module in the top-down pathway adaptively weights feature maps based on scales of objects contained in the input image. For example, given an image containing smaller objects, higher-resolution feature maps are emphasized more. The scale attention module is trained along with the other parts of the object detection network using the training image dataset.

2.2. Attention Mechanisms. Various forms of attention mechanisms have been used in biological systems, i.e., human beings, as well as in recent neural networks. The human visual system is immediately attracted to salient locations in the visual field. This behavior indicates that the human visual system assigns different importance to an image's different locations to perform the task at hand. Spatial attention mechanism has been applied in computer vision. For example, in [14], Pinheiro and Collobert proposed a CNN trained to put higher weight on pixels important in classifying an image. The CNN learns to perform segmentation tasks based on the per-image class label in a weakly supervised setting.

Another well-known example of attention is the adaptive weighting of channels in feature maps of neural networks. Fu et al. in [15] employed channel attention to selectively emphasize channels in feature maps, as well as spatial attention, for scene segmentation.

This paper applies the idea of the attention to scale space to adaptively weight multiple-scale feature maps of the bottom-up pyramid and top-down pyramid of the feature pyramid networks for fusion.

3. Methods

3.1. Overview. Our proposed method, scale adaptive feature pyramid networks (SAFPNs), is an improvement over the original FPN. Feature pyramid networks (FPNs) [11], depicted in Figure 1(b), try to create a multiresolution feature pyramid in which feature maps at all resolution levels are highly semantic and, at the same time, have a high resolution for accurate object localization and classification. This is achieved by a top-down pathway using up-sampling and lateral connections linking the top-down pathway with the bottom-up pathway inherent in a CNN. The FPN uses fixed, equal weights to integrate information coming from the two pathways. Fixed weight used for the integration, however, is not optimal for every image.

Our SAFPN, illustrated in Figure 1(c), computes weights for the integration adaptively, per scale and per each input image, so that the feature map at a scale concordant with the scale of object in the input image is weighted more than the others. For example, for an image containing smaller objects, a higher-resolution feature map would be weighted more than the other feature maps. For an image containing larger-scale objects, on the other hand, the lower resolution feature map would be weighted more than the others. The adaptive weighting is learned by the scale attention module (SAM), which is trained end-to-end with the other parts of the object detection network. The proposed approach, SAFPN, is versatile in which it applies to many different object detection architectures, including both 1-stage and 2-stage networks for object detection. We will later evaluate the SAFPN on both the faster R-CNN [5], a 2-stage method, and RetinaNet [10], a 1-stage method.

3.2. Adaptive Multiscale Feature Integration. The scale adaptive feature pyramid network (SAFPN) is illustrated in Figure 1(c). The original FPN, illustrated in Figure 1(b), uses a fixed weight of 0.5 to weigh both feature maps C_i of the bottom-up pathway and P_i of the top-down pathway. Our SAFPN employs the scale attention module to compute weights for integration adaptively for each input image. This is done by the SAM observing the strengths of responses of feature maps at various scales in the bottom-up pathway induced by a backbone CNN.

Let us assume the SAFPN to contain five scale levels, $C = \{C_1, C_2, C_3, C_4, C_5\}$, in its bottom-up pathway induced by convolutional layers of a CNN. A feature map C_i has height H, width W, and depth D. Of these multiple levels of feature maps in the pyramid, C_1 is the highest resolution yet least semantic feature map, while C_5 is the lowest resolution yet most semantic feature map. Similarly, let $P = \{P_2, P_3, P_4, P_5\}$ be the set of feature maps generated by the top-down pathway formed by up-sampling. As shown in Figure 1(c), the lowest resolution feature map at the top of the top-down pathway, P_5, is obtained by reducing the number of channels of C_5 using 1×1 convolution. Other feature maps P_{i-1} at i^{th} level, where $i = 2 \sim 4$ is computed by using the following equation, except for P_5:

$$P_{i-1} = w_{i-1}^{C} \cdot C_{i-1} + w_{i-1}^{P} \cdot P_i, \tag{1}$$

where C_i and P_i, i^{th} feature map in C and P, are weighted by the (scalar) weights w_i^C and $w_i^P = (1 - w_i^C)$, respectively. The weights are determined based on the strengths of responses of the feature maps $C = \{C_2, C_3, C_4, C_5\}$ in the bottom-up pathway and are normalized using the Softmax function as equation (2) and Figure 2:

$$w_i^C = \frac{\exp(\|C_i\|)}{\sum_{i=2}^{n} \exp(\|C_i\|)}. \tag{2}$$

The strength of response of the feature map C_i is computed as its L^p-norm of all the values in the feature map using equation (3). In the experiments below, we tried several different values of p; L^p-norm corresponds to $L1$-norm, $L2$-norm, and $L\infty$-norm if $p = 1$, $p = 2$, or $p = \infty$, respectively. We also tried $L0.5$-norm and square of $L2$-norm:

$$\|C_i\| = \left(\sum_{h=1}^{H} \sum_{w=1}^{W} \sum_{d=1}^{D} |c_{h,w,d}|^p \right)^{1/p}. \tag{3}$$

Note that the feature maps of resolution levels $i \in \{2, 3, 4, 5\}$ are used, and the highest resolution feature map C_1 and its counterpart P_1 are not used in our implementation. C_1 and P_1 are not used in part due to their large memory footprint.

4. Results and Discussion

4.1. Experimental Settings. To evaluate the proposed scale adaptive feature pyramid network (SAFPN), we conduct a set of experiments over the following five variations of networks:

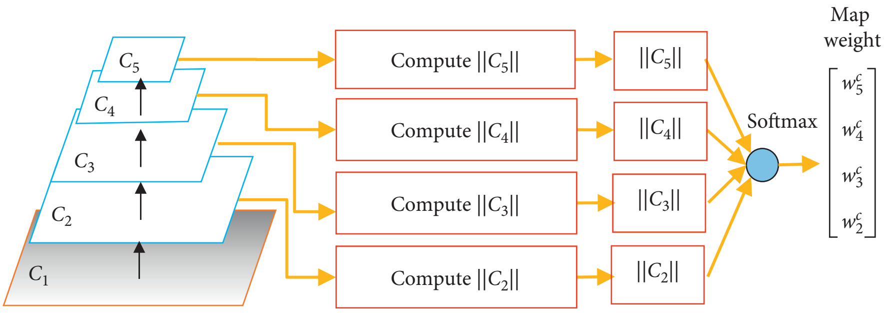


FIGURE 2: Computing weights for feature map integration based on strengths of responses of the feature maps at various scales.

(1) FRCNN: original faster R-CNN [5] that uses a single resolution feature map (i.e., no FPN).

(2) FRCNN-FPN: faster R-CNN added with the multiresolution feature map FPN [11]. The original FPN [11] and RetinaNet [10] use fixed weights $(w_i^C, w_i^P) = (0.5, 0.5)$ for integration. We also experimented with different values of fixed weights as listed below to see how it affects object detection accuracy:

$$w_i^C = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}. \tag{4}$$

(3) FRCNN-SAFPN: faster R-CNN added with the proposed SAFPN. The pair of weights (w_i^C, w_i^P) for feature map integration is determined adaptively.

(4) RetinaNet: original RetinaNet [10] with the (fixed-weight) FPN.

(5) RetinaNet SAFPN: RetinaNet [10] whose FPN is replaced with the SAFPN.

In the list above, two networks using the proposed approach, the faster R-CNN SAFPN (3) and the RetinaNet SAFPN (5), are compared against others, which are (1), (2), and (4). Recall that the original RetinaNet (4) includes the FPN by birth, but its weights for feature map integration are fixed at unity.

Training is done by using combinations of Pascal VOC 2007 and Pascal VOC 2012 datasets. We train the networks using either

(1) Pascal VOC 2007 trainval dataset or

(2) Pascal VOC 2007 + 2012 joint trainval dataset

Evaluation is done by using Pascal VOC 2007 test set (Pascal VOC 2012 test set is not openly available and thus cannot be used). We use mean average precision (mAP) as index of accuracy.

We added our SAFPN to the faster R-CNN and RetinaNet reimplementations by a group called UCAS-Det downloaded from [16, 17]. UCAS-Det makes available these networks with several different backbone networks, e.g., ResNet-50 and ResNet-101. For both the faster R-CNN and for the RetinaNet, we chose ResNet_v1_101 [12] pretrained by using the ILSVRC-2012-CLS image classification dataset as the backbone. Both networks are then trained by using either the Pascal VOC2007 trainval or Pascal VOC 2007 + 2012 joint trainval dataset. Pixel resolution of images in these databases varies, but the majority of images are either 500 × 375 (landscape) or 375 × 500 (portrait).

The training used the Adam [18] optimizer with momentum 0.9 and minibatch size 1. The small minibatch size of 1 is due to memory limitation (5GByte) of the GPU, Nvidia Tesla K20, which we used. Training is done for 100,000 epochs for the Pascal VOC 2007 trainval dataset and 150,000 epochs for the Pascal VOC 2007 + 2012 joint trainval dataset. The learning rate is manually scheduled, starting at 1×10^{-3} and reduced to 1×10^{-4} at 50,000th epoch and to 1×10^{-5} at 70,000th epoch.

Table 1 shows the number of images and number of objects in Pascal VOC 2007 and Pascal VOC 2012 datasets. Both of these two datasets consist of 20 categories.

4.2. Experimental Result.

4.2. Experimental Result. Tables 2 and 3 compare accuracies in mAP [%] of the five cases listed above. It also compared $L1$-norm, $L\infty$-norm, and $L2$-norm for computing strength of response for a feature map C_i in determining weights for feature map fusion.

The results in both Tables 2 and 3 show that for both the 2-stage network faster R-CNN [5] and for 1-stage network RetinaNet [10], the proposed SAFPN produces the highest accuracy among those compared.

In Table 2, networks are trained using the Pascal VOC 2007 train dataset. In the table, the FRCNN-SAFPN produced mAP of 78.3%, which is significantly better than 74.6% of the original FRCNN (without FPN) and 76.1% of the FRCNN-FPN that uses fixed weight $(w_i^C, w_i^P) = (0.5, 0.5)$. In case of RetinaNet, mAP improved from 73.3% using the original FPN to 74.2% using the proposed SAFPN. Of various norms tried for the pooling of responses, square of $L2$-norm, $(L2)^2$, shows the highest accuracy, very closely followed by $L2$-norm.

In Table 3, networks are trained using the Pascal VOC 2007 + 2012 joint trainval dataset that contains more training samples than Pascal VOC 2007 train only. The FRCNN-SAFPN produced mAP of 79.9% in the table, which is significantly better than 76.4% of FRCNN and 78.4% of FRCNN-FPN. Tendency for RetinaNet is similar; accuracy in mAP improved from 76.3% to 78.5% by swapping the FPN with SAFPN. Again, the square of $L2$-norm, $(L2)^2$, does the best among the norms we tried in pooling a feature map's response.

Table 1: Pascal VOC 2007 and Pascal VOC 2012 datatsets.

Splits		Pascal VOC 2007	Pascal VOC 2012
Train	# images	2,501	5,717
	# objects	6,301	13,609
Val	# images	2,510	5,823
	# objects	6,307	13,841
Trainval	# images	5,011	11,540
	# objects	12,608	27,450
Test	# images	4,952	—
	# objects	12,032	—

Table 2: Accuracies in mAP [%] of the faster R-CNN [5] without the FPN, with the (original) FPN, and the proposed SAFPN.

	W/O FPN	FPN	SAFPN				
			$L0.5$	$L1$	$L2$	$(L2)^2$	$L\infty$
Faster R-CNN	74.6	76.1	76.1	78.1	**78.3**	**78.3**	75.3
RetinaNet		73.3	72.0	**74.2**	74.1	**74.2**	73.1

Accuracies for RetinaNet [10] are also shown with the FPN and SAFPN. Training is done using the Pascal VOC 2007 train dataset.

Table 3: Accuracies in mAP [%] of the faster R-CNN [5] without FPN, with (original) FPN, and proposed SAFPN.

	W/O FPN	FPN	SAFPN				
			$L0.5$	$L1$	$L2$	$(L2)^2$	$L\infty$
Faster R-CNN	76.4	78.4	78.9	79.5	79.5	**79.9**	76.5
RetinaNet		77.6	76.1	77.2	78.3	**78.5**	76.4

Accuracies for RetinaNet [10] are also shown with the FPN and SAFPN. Training is done using the Pascal VOC 2007 + 2012 joint trainval dataset.

Figures 3 and 4 plot, for the FPN, the effects that fixed values (w_i^C, w_i^P) have on accuracy. It is compared against the cases using the SAFPN, which appear as horizontal lines. We tried values of (w_i^C, w_i^P) as listed below and plotted accuracy against it:

$$w_i^C = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}. \quad (5)$$

Accuracy varies depending on (w_i^C, w_i^P), but the proposed adaptive SAFPN performs equal or better than the hand-tuned fixed-weight integration.

In the case of hand-tuned integration, for both the RetinaNet and faster R-CNN, weighting the bottom-up pyramid feature maps C_i more than top-down pyramid feature maps P_i seems to produce higher accuracy.

Figure 5 shows examples of object detection using the FRCNN-FPN and FRCNN-SAFPN. As indicated in Figure 5, the FRCNN-FPN uses (w_i^C, w_i^P) = (0.5, 0.5) for every resolution level of its FPN. The SAFPN, on the other hand, uses adaptively computed values of w_i^C determined from each input image for each of the four resolution levels. Using fixed weights, a small cow in the background (A), as well as a small cow close to the center (C) is not detected. Using the SAFPN, however, small-sized objects are detected, as in (C) and (D). Note that, for the images having small objects, i.e., (B) and (D), the SAFPN weighs higher-resolution feature maps at

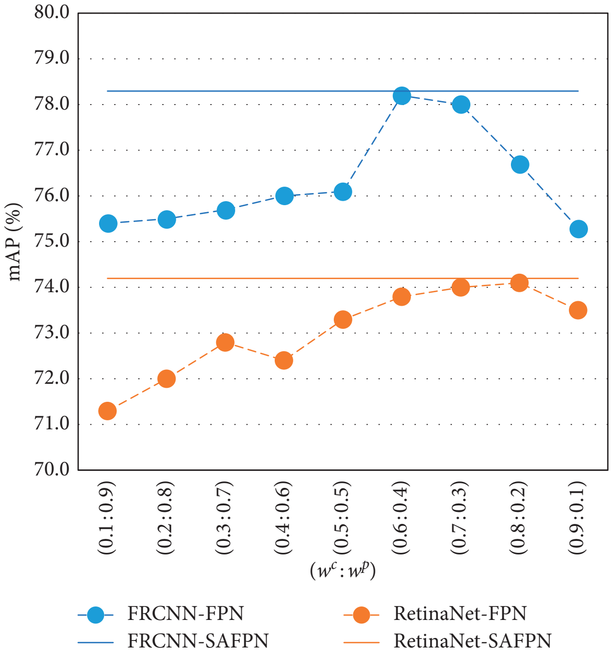


Figure 3: Fixed weight FPN using various ($w^c : w^p$) ratios (dashed lines) versus the SAFPN using square of $L2$-norm (solid lines) trained using the Pascal VOC 2007 train dataset. Note that the original FPN [11] uses ($w^c : w^p$) = (0.5 : 0.5). The FRCNN-FPN original (0.5 : 0.5 fixed weights for every scale).

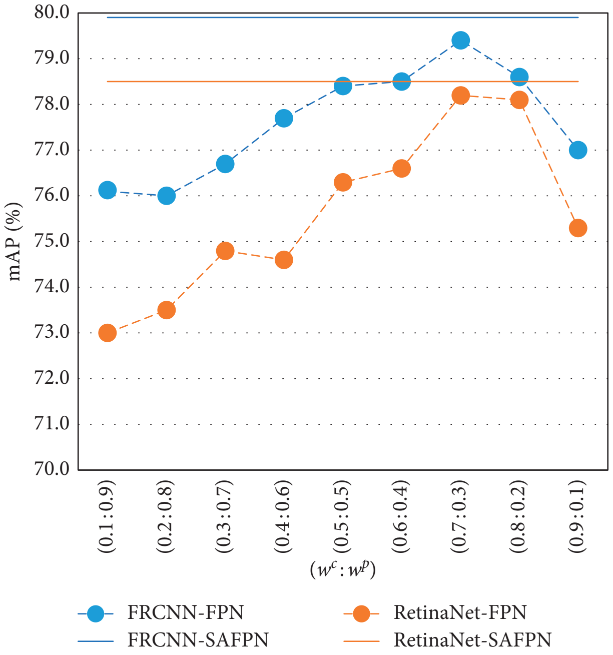
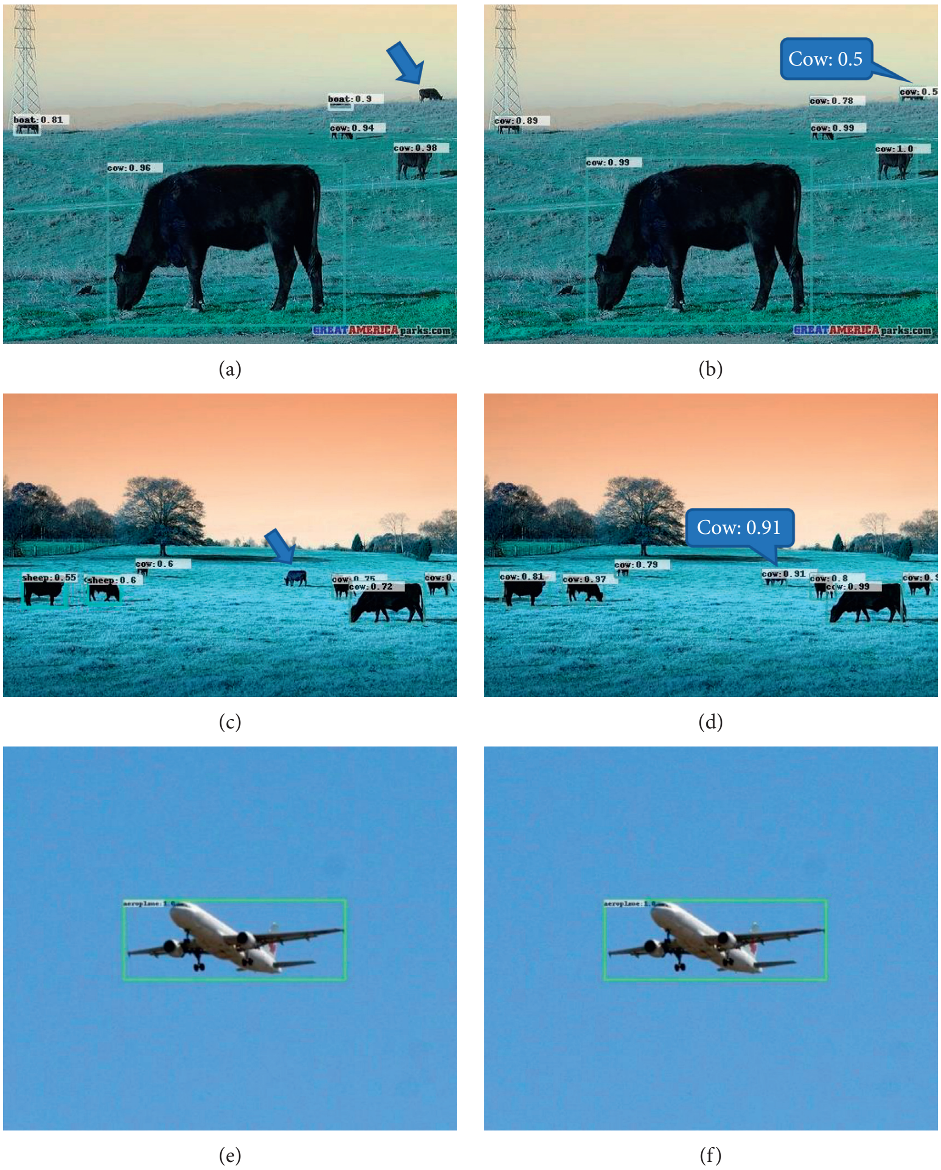


Figure 4: Fixed weight FPN using various ($w^c : w^p$) ratios (dashed lines) versus the SAFPN using square of $L2$-norm (solid lines) trained using the Pascal VOC 2007 + 2012 train dataset. Note that the original FPN [11] uses ($w^c : w^p$) = (0.5 : 0.5). The FRCNN-SAFPN adaptive weights.



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

Figure 5: Examples of fixed weights using FPN (a) and adaptive weights using SAFPN (b). SAFPN gave large weight to high resolution feature map c_3 in (B) and (D), which helps detection of small objects. For image (F) containing a large object, SAFPN gave large weight to low resolution feature map c_5. (a) A "cow" far in the background is not detected $[w_c^2, w_c^3, w_c^4, w_c^5] = [0.5, 0.5, 0.5, 0.5]$. (b) A small "cow" far in the background is detected. Smaller scales (level 2 and 3) have higher weights for this image $[w_c^2, w_c^3, w_c^4, w_c^5] = [0.28, 0.34, 0.19, 0.16]$. (c) Small cow near the center of image is not detected $[w_c^2, w_c^3, w_c^4, w_c^5] = [0.28, 0.34, 0.19, 0.16]$. (d) Small "cow" near the center of the image is detected. Smaller scales (levels 2 and 3) have higher weights for this image $[w_c^2, w_c^3, w_c^4, w_c^5] = [0.31, 0.34, 0.19, 0.16]$. (e) This image contains easy-to-detect large object, an airplane $[w_c^2, w_c^3, w_c^4, w_c^5] = [0.5, 0.5, 0.5, 0.5]$. (f) With a large object, SAFPN weighs larger scale (i.e., lowest resolution) feature map at level 5 $[w_c^2, w_c^3, w_c^4, w_c^5] = [0.06, 0.08, 0.25, 0.61]$.

level 3 the most. In comparison, for the image (F) having a large airplane, the SAFPN weighs the lowest resolution feature map at level 5 the most.

5. Conclusion

In this paper, we tackled the issue of correctly detecting objects having varying scales, especially those having small scales, from images.

With a single low-resolution feature map, such as those found in the faster R-CNN [5], small objects localization and classification are difficult. The feature pyramid network (FPN) [11] significantly improved object detection accuracy by using pyramids of multiple resolution feature maps that provide high semantic content at multiple resolution levels. However, its ability to detect small-scale objects is still limited. We conjectured that the limitation is in part due to fixed weights used in the integration of feature maps between the bottom-up pyramid and the top-down pyramid.

This paper proposed an improvement to the FPN [11] called scale adaptive feature pyramid network (SAFPN) that adaptively determines weight for feature map fusion per input image for each scale. The weights are computed per input image for each scale based on responses of feature maps in the bottom-up feature pyramid by the scale attention module. While the SAFPN incurs an increase in computational cost, an increase in memory footprint is negligible.

We performed a set of experiments using both the 2-stage network faster R-CNN [5] and 1-stage network RetinaNet [10], both of which are modified with the SAFPN. The set of experiments has shown that the proposed SAFPN significantly improves object detection accuracy over the FPN. Accuracy measured in mean average precision (mAP) for the original faster R-CNN and faster R-CNN with the (fixed weight) FPN is 74.6% and 76.1%, respectively, when trained using the Pascal VOC 2007 dataset. The faster R-CNN with the proposed SAFPN improved mAP value to 78.3%. For RetinaNet, replacing its (fixed weight) FPN with the proposed SAFPN improved accuracy in mAP from 73.3% to 74.2%.

Future work includes combining the proposed scale-space attention mechanism with some form of spatial attention mechanism to improve accuracy.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Zhejiang Provincial Technical Plan Project (nos. 2020C03105 and 2021C01129).

References

[1] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, April 2014.

[2] 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 (CVPR)*, pp. 580–587, Columbus, OH, USA, June 2014.

[3] K. He, X. Zhang, X. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[4] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, CL, USA, Dec 2015.

[5] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99, Montreal, Canada, December 2015.

[6] W. Liu, D. Anguelov, D. Erhan et al., “SSD: single shot MultiBox detector,” in *Computer Vision—ECCV 2016*, pp. 21–37, Springer, Berlin, Germany, 2016.

[7] 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 CVPR*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[8] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE conference on CVPR*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[9] J. Redmon and A. Farhadi, “YOLOv3: an incremental improvement,” 2018, http://arxiv.org/abs/1804.02767.

[10] T. Y. Lin, P. Goyal, R. Girshick, and K. He, “Focal loss for dense object detection,” in *Proceedings of the IEEE ICCV*, pp. 2980–2988, Venice, Italy, 2017.

[11] 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 CVPR*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on CVPR*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems (NIPS 2012)*, vol. 25, no. 2, pp. 1097–1105, 2012.

[14] P. O. Pinheiro and R. Collobert, “From image-level to pixel-level labeling with convolutional networks,” in *Proceedings of the IEEE Conference on CVPR*, pp. 1713–1721, Boston, MA, USA, June 2015.

[15] J. Fu, J. Liu, H. Tian et al., “Dual attention network for scene segmentation,” in *Proceedings of the IEEE Conference on CVPR*, pp. 3146–3154, Long Beach, CA, USA, 2019.

[16] Y. Yang, “Faster-R-CNN_Tensorflow,” 2020, https://github.com/DetectionTeamUCAS/Faster-RCNN_Tensorflow.

[17] Y. Yang, “Focal loss for dense object detection,” 2020, https://github.com/DetectionTeamUCAS/RetinaNet_Tensorflow.

[18] D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” in *Proceedings of the ICLR*, San Diego, CA, USA, April 2015.