Multipatch Feature Pyramid Network for Weakly Supervised Object Detection in Optical Remote Sensing Images

Pourya Shamsolmoali, *Member, IEEE*, Jocelyn Chanussot, *Fellow, IEEE*, Masoumeh Zareapoor, Huiyu Zhou, and Jie Yang

***Abstract*— Object detection is a challenging task in remote sensing because objects only occupy a few pixels in the images, and the models are required to simultaneously learn object locations and detection. Even though the established approaches well perform for the objects of regular sizes, they achieve weak performance when analyzing small ones or getting stuck in the local minima (e.g., false object parts). Two possible issues stand in their way. First, the existing methods struggle to perform stably on the detection of small objects because of the complicated background. Second, most of the standard methods used handcrafted features and do not work well on the detection of objects parts that are missing. We here address the above issues and propose a new architecture with a multipatch feature pyramid network (MPFP-Net). Different from the current models that, during training, only pursue the most discriminative patches, in MPFP-Net, the patches are divided into class-affiliated subsets, in which the patches are related, and based on the primary loss function, a sequence of smooth loss functions is determined for the subsets to improve the model for collecting small object parts. To enhance the feature representation for patch selection, we introduce an effective method to regularize the residual values and make the fusion transition layers strictly norm-preserving. The network contains bottom-up and crosswise connections to fuse the features of different scales to achieve better accuracy compared to several state-of-the-art object detection models. Also, the developed architecture is more efficient than the baselines.**

***Index Terms*— Feature fusion, multiple patch learning (MPL), multiscale object detection, remote sensing images (RSIs).**

I. Introduction

HIGH-RESOLUTION remote sensing images (RSIs) are now available to facilitate a wide variety of applications, such as traffic management [1] and environment monitoring [2]. Recently, the application of object detection in RSIs has been extended from rural development to other areas, such as urban inspection [3], and the RSIs intensely increase in both quantity and quality. Cheng and Han [3] presented a comprehensive survey and compared the performance of different machine learning methods for RSI interpretation. Machine learning and deep learning-based models have been widely used for RSI object detection and classification [2], [4], [5]; however, supervised learning models require a large scale of annotated datasets to support satisfactory detection. Furthermore, RSI annotation needs to be undertaken by trained professionals. In contrast, weakly supervised learning is another technique to augment datasets for object detection [6], [7]. One of the popular weakly supervised object detection methods is the combination of multiple instance learning (MIL) and deep neural networks [8]. Despite that, their method is related to object parts rather than full body, which is due to the nonconvexity of the loss functions.

In practice, standard machine learning models have two main stages for object detection: first feature extraction and then classification. In [9], the extracted features were used as input, followed by a support vector machine to classify the predicted targets. Standard machine learning approaches are influenced by the quality of the handcrafted and light learning-based features. Despite their promising results, standard machine learning systems fail to provide robust outcomes in challenging circumstances, for example, changes in the visual appearance of objects and complex background clutters [3].

In recent years, object detection is progressing due to the advance of convolution neural networks (CNNs). CNNs are able to learn features' representation using a large size of data [10]. An enhanced CNN introduces a process called selective search [11] by adapting segmentation as a selective search strategy for improving detection accuracy with higher speeds. In comparison with the analysis of natural images, the main difficulty in RSI object detection is the sizes of objects. For small objects, because of low resolution, extracting significant and discriminative features is difficult; therefore, recent works are driven toward exploring solutions of extracting discriminative features [12]. For example, a rotational R-CNN was proposed by Guo *et al.* [13] for supervised object detection. In addition, learning feature representations is a major problem in image processing tasks, and detecting multiscale objects is challenging. To overcome this problem, pyramidal feature

Manuscript received April 5, 2021; revised July 19, 2021 and August 16, 2021; accepted August 17, 2021. Date of publication August 30, 2021; date of current version January 31, 2022. This work was supported in part by NSFC under Grant 61876107 and Grant U1803261, in part by the National Key Program of China under Grant 2019YFB1311503, and in part by the Committee of Science and Technology, Shanghai, under Grant 19510711200. *(Corresponding author: Jie Yang.)*

Pourya Shamsolmoali, Masoumeh Zareapoor, and Jie Yang are with the Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: pshams@sjtu.edu.cn; mzarea@sjtu.edu.cn; jieyang@sjtu.edu.cn).

Jocelyn Chanussot is with the GIPSA-lab, Université Grenoble Alpes, CNRS, Grenoble INP, 38000 Grenoble, France, and also with the Faculty of Electrical and Computer Engineering, University of Iceland, 101 Reykjavik, Iceland (e-mail: jocelyn.chanussot@grenoble-inp.fr).

Huiyu Zhou is with the School of Informatics, University of Leicester, Leicester LE1 7RH, U.K. (e-mail: hz143@leicester.ac.uk).

Digital Object Identifier 10.1109/TGRS.2021.3106442

representations have been introduced to represent an image through multiscale features in object detectors [14]–[16]. The feature pyramid network (FPN) [14] is among the best representative approaches for producing pyramidal feature representations of objects. Typically, pyramid models adopt a backbone network that is built for image segmentation or classification and create feature pyramids (FPs) by successively merging two or three consecutive layers in the backbone network with top-down and adjacent connections. High-level features have lower resolutions, but they are semantically strong and can be upscaled and merged with higher resolution features to create more discriminative features. Scholars who are working on object detection in RSIs have observed the powerful ability of FPN and applied this method to their works. Li *et al.* [17] proposed a feature-based method for identifying ships in RSIs. The authors proposed a region-based network to detect ships from the generated feature maps. In spite of its object detection performance, its computation is inefficient. In [18], a deep network based upon the two-stream pyramid module was proposed for detecting multiscale salient objects in RSIs. Their network has demonstrated promising performance on detecting objects of a regular size but failed to perform accurately on small-size or incomplete objects. Yang *et al.* [19] proposed a detector based on an encoder–decoder FPN for detecting clouds from RSIs. The network is simple and efficient but not effective in the crowd environments. Although FPN is effective and simple, it may not be an efficient architectural design. Recently, Ghiasi *et al.* [16] and Li *et al.* [17] added extra bottom-up and skip connection pathways onto FPN to enhance feature representations. Nonetheless, these methods only take into account one of the three dimensions (image size, depth, and width). However, by analyzing more dimensions and adopting fusion methods [20], [21], we can train a network to achieve better performance and efficiency. Shamsolmoali *et al.* [22] recently introduced a new architecture named SPN that improves the performance of FPN by extracting effective features from all the layers of the network for describing different scales' objects. This architecture is introduced to learn from the multilevel feature maps and improve the semantics of the features.

To tackle the above problems, in this article, we introduce a scalable architecture to build effective pyramidal representations, named multiple patch FPN (MPFP-Net). More specifically, the proposed model mainly consists of two components. First, we proposed a multiple patch learning (MPL) scheme to deal with the nonconvexity and lack of full object representation. MPL treats input images as patches. During training, MPL learns patch subsets, which have mutual correlation. Patch subsets with accurate trailer parameters can activate semantic regions to describe a full object. Second, we propose a new pyramid network with cross-scale connections for producing multiscale representative features. An additional advantage of the modular pyramidal architecture is the capability of efficient multiscale object detection. There are three contributions made in this article.

1) Our proposed MPL strategy is built on a CNN to accurately describe an object such that the network performs even robustly with different backbone models, such as ResNet [23] and SPN [22]. By adopting SPN as the backbone of the proposed model, the detection accuracy and speed of MPFP-Net are better than those of the other state-of-the-art models.
2) We incorporate scalewise image fusion within the FPN architecture. The cross-scale connections are used to adapt the number of the channels and the size of the feature maps to maintain the norm of the gradients. Also, we propose a computationally efficient method to extract and normalize a multilevel feature response to the corresponding level. By using a layer for extracting multiscale feature maps, the network weights are updated once per iteration, which considerably improves the training efficiency of the proposed model.
3) We fully evaluate several recent deep CNNs for object detection in RSIs, and the overall performance is reported.

The rest of this article is structured as follows. Section II shows a brief review of the existing object detection methods on RSIs. In Section III, we describe our proposed architecture in detail. In Section IV, we report the experimental results on the RSI datasets. Section V contains the ablation study. We conclude this article in Section VI.

II. Related Work

In the last ten years, object detection in RSIs has achieved significant progress. In this section, we will discuss the available object detection technologies for natural and RS images. In particular, we focus on the discussion on weakly supervised models for object detection.

A. Object Detection in Nature Scene Image

Deep CNNs significantly improve the performance of object detection models in recent years. Girshick *et al.* [24] proposed a detector named R-CNN. R-CNN has a high detection rate, but its speed is limited. With the intention of increasing the accuracy and speed of object detection, Fast R-CNN [25] was proposed to use bounding-box regression with end-to-end training. Later, Faster R-CNN [26] was introduced to combine object proposals and identification within a unified network with impressive efficiency. Generally, there are two approaches to deal with the scale-variation problem. The first approach is to use featurewise image pyramids to generate semantical multiscalar features. Features from the images of different scales return individual predictions to generate the final detection. With regard to localization precision and detection accuracy, features from the images of different sizes outweigh the features that are only based on the images of a single size [13]. The second approach is to detect objects in the FP extracted from different layers in the network, while, as an input, only a single-scale image is applied. This methodology requires a smaller memory space than the first one. In addition, the standard FP unit can be modified and inserted into the state-of-the-art CNN-based detectors. Lin *et al.* [14] proposed an FPN with double-path connections to seek more representative features. Ding *et al.* [27] proposed a weakly supervised pyramid CNN for object detection. The model

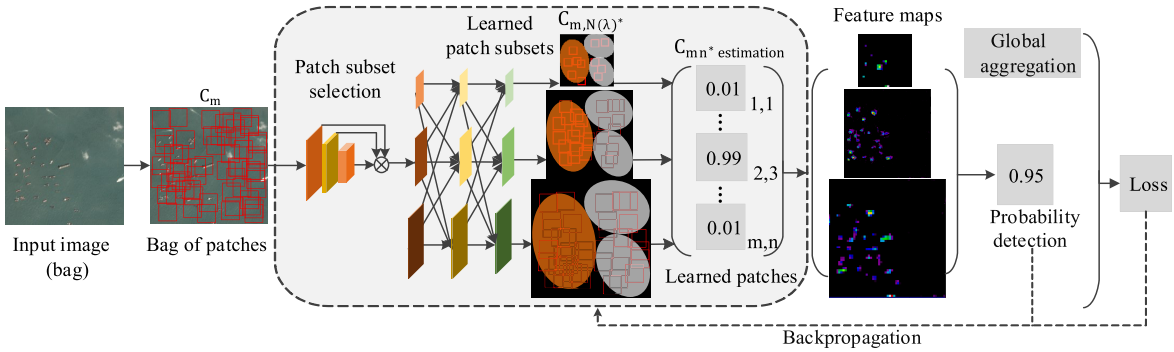

Fig. 1. Overall architecture of the proposed MPFP-Net; it contains patch detection, subsets' selection, feature extraction, and instance classification.

consists of a hierarchical configuration with both top-down and bottom-up connections to learn high-level semantics and low-level features.

B. Object Detection in RSI

Compared to object detection in natural scene images, object detection in RSIs has additional challenges. This topic has been intensively studied over the last few years. Conventional object detection models have ranking systems to categorize objects and background [28]. For example, Wu *et al.* [29], [30] proposed an efficient detection framework based on rotation-invariant feature aggregation and adopted a learning method to acquire significant and meaningful features for small object detection.

With deep learning development, there has been a significant improvement in object detection. Cheng *et al.* [31] proposed an equivariant CNN method by introducing a regularization method for object detection in RSIs. Dong *et al.* [32] proposed a deep transfer learning method on the basis of R-CNN to minimize the loss of tiny objects in RSIs. Moreover, transfer learning can be used to label RSIs by annotating both object positions and classes. Deng *et al.* [33] proposed multiscale object detection models in RSIs using inspection modules to train the subnetworks. The proposed model has significant detection performance on multiscale objects but is not efficient due to the depth of the network. Different conditions and objects variations in RSIs bring various challenges to object detection in such images. Consequently, it is hard to obtain satisfactory results by deploying the available object detection models. Furthermore, due to the deficit of training data and the complexity of the network architecture to handle various objects with multiple scales and complex backgrounds, we introduce a novel multiscale deep learning model for object detection in RSIs.

C. Weakly Supervised Object Detection

MIL is a weakly supervising learning method that, in the training phase, treats each image as a "bag" and constantly picks high-scored instances from the bags. It performs similar to an expectation–maximization algorithm for estimating instances and training detectors concurrently. On the other hand, such a model, due to lack of local minima in the nonconvex loss functions, does not have a smooth training process [34]. To mitigate the nonconvexity issue, bundling was used as one of the preprocessing approaches to simplify the selection of relevant instances [35]. A box-level supervision method was presented to decrease the solution space throughout a recurrent constraint network [7]. In [36], MIL-Net was proposed in which the convolution operator and filters are used as detectors. Nevertheless, MIL-Net's loss functions stay nonconvex. To deal with this problem, spatial regularization is used [36] within the cascaded convolutional networks. Current approaches mainly use selective regions (patches) as ground truth to gradually improve the classifiers [37]. Existing methods that use spatial regularization and gradual refinement are successful at enhancing object detection. Even so, it is a lack of a systematic approach to deal with the local minimum problem.

III. Proposed Model and Methodology

In this section, we present the details of our proposed model that consists of two main components: global multiple patches (image regions) learning and a variation of multiscale FPNs with a novel fusion scheme to robustly detect an object region to further improve the detection performance. Fig. 1 shows the architecture of our proposed model.

A. Multiple Patch Global Learning

In MPL, images are treated as bags and the image's regions as patches. In MPL, the labels are only allocated to bags of patches. Here, $C_m \in C$ shows the mth bag, and C represents all the bags (training images). $y_m \in Y$, while $Y = [1, -1]$ shows the bag's label, and C_m shows the possibility that the bag owns positive patches. $y_m = 1$ signifies a positive bag, which always holds a single patch. On the other hand, $y_m = -1$ signifies a negative bag, while the entire patches are negative. Let C_{mn} and y_{mn} indicate the patches and labels in bag C_m, in which $n \in \{1, 2, \ldots, N\}$, and N is the number of the patches. The MIL methods have two steps for object detection: patch selection (patch controller) as the initial step and object estimation [7], [38]. In the patch selection phase, a patch controller $f(C_{mn}, \omega_f)$ calculates the object score of all the patches to obtain a patch from C_m

$$C_{mn^*} = \text{argmax}_n f(C_{mn}, \omega_f) \tag{1}$$

in which ω_f denotes the parameters of the patch selector and n^* denotes the highest scored patch. With certain patches, a detector $d_z(C_{mn}, \omega_d)$ with parameter ω_d is trained, in which $z \in Y$. ω_d represents the parameters of the object detector. In standard MIL methods [7], [36], the above processes are combined, and $f(C_{mn}, \omega_f)$ and $d(C_{mn}, \omega_d)$ are fully integrated as follows:

$$\text{Loss}(C, \omega) = \sum_m \text{Loss}_f(C_m, \omega_f) + \text{Loss}_d(C_m, C_{mn}, \omega_d) \tag{2}$$

in which the standard loss of patch selection is determined as

$$\text{Loss}_f(C_m, \omega_f) = \max\Big(0, 1 - y_m \max_n f(C_{mn}, \omega_f)\Big) \tag{3}$$

and the loss of detector estimation is determined as

$$\text{Loss}_d(C_m, C_{mn^*}, \omega_d) = -\sum_z \sum_j \delta_{z,y_{mn}} \log d_z(C_{mn}, \omega_d) \tag{4}$$

in which y_{mn} is determined based on the metric introduced in [39]

$$y_{mn} = \begin{cases} +1, & \text{if IoU } (C_{mn}, C_{mn^*}) \geq 0.5 \\ -1, & \text{if IoU } (C_{mn}, C_{mn^*}) < 0.5 \end{cases} \tag{5}$$

for $m = n$ or $m \neq n$ the delta function $\delta_{mn} = 1$ or 0, respectively

$$\delta(mn) = \lim_{\sigma \to 0} \frac{1}{\sigma\sqrt{2\pi}}. \tag{6}$$

To enhance the performance of the patch selector, we introduce a novel optimization technique. This approach splits the patches into subsets while handling the nonconvexity of the loss function shown in (2). Our model includes a sequence of smooth loss functions from the beginning point $(\omega^0, 0)$ to the resulting point $(\omega^*, 1)$, where ω^0 is the result of $\text{Loss}(C, \omega, \Gamma)$ where $\Gamma = 0$, and ω^* is used, where $\Gamma = 1$. Thus, we determine a range of Γ, $0 = \Gamma 0 < \Gamma 1 < \cdots < \Gamma\tau = 1$ and, accordingly, improve (2) to a persistent loss function, as follows:

$$\omega^* = \arg\min_{\omega_f, \omega_d} \sum_m \text{Loss}_f(C_m, C_{m,N(\Gamma)}, \omega_f) + \text{Loss}_d(C_m, C_{m,N(\Gamma)}, \omega_d) \tag{7}$$

in which $C_{m,N(\Gamma)}$ represents the patch subset and $N(\Gamma)$ the index of $C_{m,N(\Gamma)}$, controlled by Γ, which is a continuous parameter. $\text{Loss}_f(C_m, C_{m,N(\Gamma)}, \omega_f)$ is the persistent loss function of patch selection, and $\text{Loss}_d(C_m, C_{m,N(\Gamma)}, \omega_d)$ is the persistent loss function of the detector. To learn the patch selector, a bag (image) is divided into patch subclasses. In each subclass, the objects are spatially related and may have overlapping with each other, and each class contains partially similar objects. Each subclass at least contains a single bag (image) $C_{m,N} \cup C_{m,\acute{N}} = C_m$ and $C_{m,N} \cap C_{m,\acute{N}} = \emptyset$ for $\forall N \neq \acute{N}$. The overall patches in an image (bag) are arranged by their object scores $f(C_{m,n}, \omega_f)$, and the following two processes are carried out: 1) create a patch for a subclass based on the highest object score and 2) identify the patches that are overlapped with the highest scored patch C_{m,n^*} and then include them in the subclasses. The successive patch selection is made with $0 \leq \lambda \leq 1$ in which the loss function is formulated as follows:

$$\text{Loss}_f(C_m, C_{m,N(\lambda)}, \omega_f) = \max\left(0, 1 - y_m \max_{N(\lambda)} f(C_{m,N(\lambda)}, \omega_f)\right) \tag{8}$$

where $f(C_{m,N(\lambda)}, \omega_f)$, the score of patch subclass $C_{m,N(\lambda)}$, is determined as follows:

$$f(C_{m,N(\lambda)}, \omega_f) = \frac{1}{|C_{m,N(\lambda)}|} \sum_n f(C_{mn}, \omega_f) \tag{9}$$

in which $|C_{m,N(\lambda)}|$ indicates the total number of the patches in subclass $C_{m,N(\lambda)}$ and $C_{mn} \in C_{m,N(\lambda)}$. In the time of model learning, all the patches in subclass $C_{m,N(\Gamma)}$ equally engage to adjusting the network parameters. When $\lambda = 0$, each bag C_m only has one subset that contains all the patches. Given $\lambda = 1$, C_m is divided into several subsets, where each subset bears at least a single patch, and consequently, (8) is not satisfied. Referring to (9), the score of a patch, in general, is similar to the mean score of the patches in that subset. Therefore, our defined loss function [see (7)], is convex and grows linearly compared to the standard MIL [see (3)]. In supervised learning, for the subclass $C_{m,N(\lambda)}$, the maximum average score is taken for object detection. If bounding-box annotation is not feasible, the patch selector may not be accurate, and the selected subclass may contain the background or only part of the objects. To fully outline valid patches and learn to detect objects, the patches are divided into positive and negative categories with parameter λ. Let $C_{m,N(\lambda)^*}$ be the learned patch subclass and the patch of the highest score in $C_{m,N(\lambda)^*}$ be C_{mn^*}. Patches are separated into positive and negative categories on the basis of their relations, as follows:

$$y_{mn} = \begin{cases} +1, & \text{if IoU } (C_{mn}, C_{mn^*}) \geq 1 - \lambda/2 \\ -1, & \text{if IoU } (C_{mn}, C_{mn^*}) < \lambda/2 \end{cases} \tag{10}$$

in which IoU stands for the intersection of union between two consequent patches. Equation (10) states that patches, where their IoU are above the threshold, $(1 - \lambda/2)$ are positive, and the patches with C_{mn^*} less than $\lambda/2$ are negative. As stated in (10), gradually, the other patches are identified as positive or negative, and the detector $d_z(C_{mn}, \omega_d)$, in accordance with these patches, predicts the objects with the following loss function:

$$\text{Loss}_d(C_m, C_{m,N(\lambda)}, \omega_d) = \sum_z \sum_n \delta_{z,y_{mn}} \log d_z(C_{mn}, \omega_d). \tag{11}$$

B. Feature Pyramid Network for Patch Estimation

Before applying classification to the whole image, we aim to generate image regions (patches) by randomly extracting patches from the images. In the next step, the collected patches go through the proposed FPN to estimate the probability of the objects. Consider an image X with a dimension $H \times W$ pixels. If the sliding window has $K \times K$ pixels and d denotes the stride, then X is divided to create the patches $\{c_{i,j}\}$, in which $i \in \{1, 2, 3, \ldots, \lceil (H - K)/d \rceil + 1\}$ and $j \in \{1, 2, 3, \ldots, \lceil (W - K)/d \rceil + 1\}$ represent the vertical and horizontal

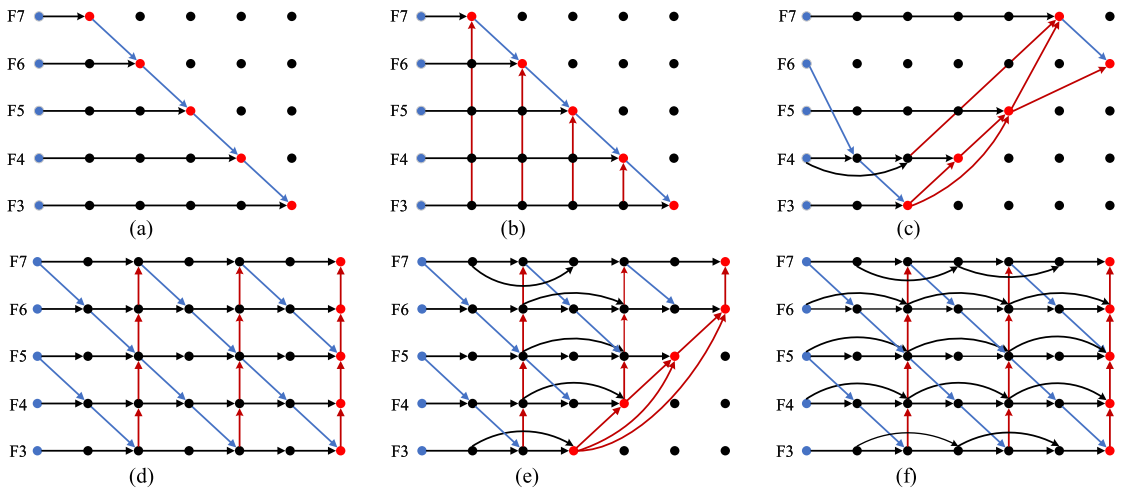

Fig. 2. Proposed network design where the top-down paths are shown in blue and bottom-up paths are shown in red. (a) FPN [14] shows a top-down path for fusing multiscale features (F3–F7). (b) PANet [15] added additional bottom-up paths to the FPN. (c) NAS-FPN [16] adopted neural architecture search to identify a feature network. (d)–(f) Different architectures of ESS-FPN learned by multiscale feature fusion. (d) ESS-FPN/39.3 AP. (e) ESS-FPN/41.7 AP. (f) Architecture of ESS-FPN used in the experiments (ESS-FPN/42.6 AP).

axes, respectively. These indexes report the information of the patches. Each bag (image) contains different numbers of patches, and the patches can be resized to match the network's input size. For the patch-level estimation, we propose a novel FPN model that contains two main components: a backbone network for constructive feature extraction and a scalewise feature fusion module to efficiently integrate high-level semantic features with low-level ones. The proposed model aims to optimize feature fusion with a better inherent way to improve object detection in RSIs. In the following sub-section, we discuss the multiscale feature fusion problem and propose a novel model with multiple cross-scale connections and weighted feature fusion. Our efficient scalewise FP architecture is named ESS-FPN.

1) Multiscale Feature Fusion: The goal of multiscale feature fusion is to combine different scales of features. $\vec{F}^{\text{in}} = (F_{l_1}^{\text{in}}, F_{l_2}^{\text{in}}, \ldots)$ represent the multiscale feature list, in which $F_{l_i}^{\text{in}}$ indicates the l_ith feature. Our objective is to transform f to effectively combine multiscale features and generate a list of features as the output: $\vec{F}^{\text{out}} = f(\vec{F}^{\text{in}})$. Fig. 2(a) represents the methodology of the standard top-down FPN [14]. This model gets the input features of level (3-7) $\vec{F}^{\text{in}} = (F_3^{\text{in}}, \ldots, F_7^{\text{in}})$, while F_i^{in} signifies a feature with $1/2^i$ resolution of the input images. As an example, if the input resolution is of 1024×1024, then F_3^{in} indicates the third-level feature $(1024/2^3 = 128)$ with the resolution of 128×128, where F_7^{in} shows the seventh-level feature with the resolution of 8×8. The standard FPN combines the features in a hierarchical fashion

$$\begin{cases} F_7^{\text{out}} = \text{conv}\left(F_7^{\text{in}}\right) \\ F_6^{\text{out}} = \text{conv}\left(F_6^{\text{in}} + \text{rescale}\left(F_7^{\text{out}}\right)\right) \\ \ldots \\ F_3^{\text{out}} = \text{conv}\left(F_3^{\text{in}} + \text{rescale}\left(F_4^{\text{out}}\right)\right) \end{cases} \tag{12}$$

in which rescaling is often a down/up scaling operation to match the image resolution and Conv denotes a convolutional operation for the feature processing.

2) Crosswise Feature Fusion: The standard top-down FPN cannot widely reuse the features that are extracted in the previous layers as it only contains single-path information flows. Several approaches have been proposed for handling this problem. In PANet [15], the authors proposed to use both the top-down and reversed paths in the network architecture, as presented in Fig. 2(b). NAS-FPN [16] introduced an FPN, based on neural network search, to develop a scalewise network architecture; however, it is not efficient to search for the best path, and it is challenging to modify the architecture, as indicated in Fig. 2(c). After having evaluated the accuracy and efficiency of these three networks, NAS-FPN achieves higher accuracy than FPN and PANet but with expensive computation. To enhance the performance and efficiency of our model, the nodes that contain single input edges are eliminated. If a particular node has a single input edge, then its contribution to the feature network is subtle and can be used in a fusion scheme. This has resulted in a multipath network.

Unlike PANet and NAS-FPN that only have top-down and bottom-up paths, in our bottom-up and crosswise model, each path is treated as a single feature network layer. The same layer repeats several times in various directions, and at the final stage, the features at different scales are fused to generate high-level feature fusion. To fuse a number of input features with various resolutions, a simple yet effective method reconfigures them to be of equal resolutions and then merges them. PAN [15] proposed global upscaling for improving pixel localization. Standard feature fusion methods equally treat the overall input features. Nevertheless, as the features of the input images have various resolutions, they do not share equal contributions to the output feature. To deal with this problem, we propose that each input has an additional weight during the feature fusion. In light of this idea, the following three weighted fusion approaches are considered.

a) Weighted feature fusion: As demonstrated in [40], a learnable weight can improve the accuracy with very small computation costs. Nevertheless, due to the unbounded scalar

weight, it may lead to unstable training. For this reason, weight normalization is generally used to limit the range of values for weights. As discussed earlier, the outputs of the top layers are much similar to the ground truth. Using the same weights for all the locations compels the network to learn better fusion weights, which, unfortunately, misses the contributions from the low-level features. This prevents low-level features from delivering adequate edge information, which is useful to identify object boundaries. Thus, before feature fusion, we deal with the scale variation of different level responses via normalization of their scales. Thereby, a robust weight learner can avoid scale variation and better learn fusion weights. A systematic multilevel feature extractor with a normalizer is here proposed to extract and normalize multilevel responses to the same scale [see Fig. 2(e) and (f)]. In particular, the feature normalization unit in the module is in charge of feature map normalization in each layer. In the proposed dynamic fusion model, two different schemes for predicting adaptive fusion weights are devised to determine the location fusion weights and ensure location invariance. The model equally treats all the feature maps, and global fusion weights are learned subsequently. Next, the model adjusts multilevel cross outputs A_{cross} of size $H \times W$ to acquire a fused output $A_{\text{fuse}} = f(A_{\text{cross}})$ by combining multilevel and multiscale outcomes. The fusion process is given as follows:

$$A^i_{\text{fuse}} = w^i_1 A^i_{\text{cross6}} + w^i_2 A^i_{\text{cross5}} + w^i_3 A^i_{\text{cross4}} + w^i_4 A^i_{\text{cross3}} \quad (13)$$

in which $(w^i_1, w^i_2, \ldots, w^i_5)$ are the parameters of the kth 1×1 convolution layer, denoting the fusion weights for the ith levels. The above equation can be generalized to the following form:

$$A_{\text{fuse}} = f(A_{\text{cross}}; W) \quad (14)$$

in which f summarizes the operation of (13) and $W = (w^i_1, w^i_2, \ldots, w^i_5)$ represents the fusion weights. In addition, an adaptive weight learner is proposed to learn the fusion weights based on the feature condition as follows:

$$A_{\text{fuse}} = f(A_{\text{cross}}; \Phi(x)) \quad (15)$$

where x indicates the feature map. The above equations illustrate the key differences between our designed adaptive weight fusion model and the current static weight fusion model. The fusion weight W is strongly based on the feature map x, i.e., $W = \Phi(x)$. Each input feature map x will engender different parameters $\Phi(x)$ and, consequently, adjust this adaptive weight learner. Thus, the model can quickly adjust to the input image and satisfactorily learn multilevel outcome fusion weights in an end-to-end mode. $\Phi(x)$ represents two fusion weight strategies, location-invariant weight learners, and location-adaptive weight. The location-invariant weight learner totally learns $5k$ fusion weights that are generated from each location of the fused feature maps

$$\Phi(x) = \left(w^i_1, w^i_2, w^i_3, w^i_4, w^i_5\right), \quad i \in [1, k]. \quad (16)$$

On the other hand, the location-adaptive weight learner generates $5k$ fusion weights in accordance with different spatial locations, which, in sum, leads to $5k \times H \times W$ weighting parameters

$$\Phi(x) = (W_{s,t})$$
$$W_{s,t} = \left(\left(w^i_1\right)_{s,t}, \left(w^i_2\right)_{s,t}, \left(w^i_3\right)_{s,t}, \left(w^i_4\right)_{s,t}, \left(w^i_5\right)_{s,t}\right) \quad (17)$$

where $s \in [1, H]$, $t \in [1, W]$ and $i \in [1, k]$. In this model, the location-invariant weight learner generates global fusion weights, whereas the location-adaptive weight learner aggregates the generated fusion weights for each location based on the spatial variations.

b) Softmax fusion: If multiclass objects exist, we apply softmax to all the weights; therefore, all the weights are normalized to the range between 0 and 1, indicating the importance of each input. Then, the weighted feature maps are aggregated across all the scales to build the fused feature maps. Here, $f^s_{l,c}$ denotes multiscale features where l, c, and s are the indexes of locations, channels, and scales, respectively. In our model the attention module F is consist of $F(f^1_l, \ldots, f^S_l)$, and the fused feature map is formulated as

$$\hat{f}(l, c) = \sum_{s=1}^{S} \hat{w}^s_{l,c} . f^s_{l,c}. \quad (18)$$

Among all the previous works [38] where sigmoid is used for selecting features from different scales $w^s_l = (\exp(h^s_l))/(\sum_{t=1}^{s} \exp(h^t_l))$, in our model, softmax weight is used as $\hat{w}^s_{l,c} = 1/(1 + e^{-h^s_{l,c}})$. Since the proposed feature fusion model is performed across all the layers instead of only the final layer [15], we realized that softmax has a better performance in fusing multiscale features across different layers. However, having softmax causes additional costs. To reduce the extra costs, we introduce an instant fusion approach.

c) Instant fusion: In this approach, softmax is not used; therefore, the process is faster. In this operation, the total value of each weight falls between 0 and 1. The instant fusion is formulated as

$$I_f = \sum_i \frac{w_i}{\theta + \sum_j w_j} . I_i \quad (19)$$

in which $w_i \geq 0$ is obtained using the standard Relu activation function, and to prevent numerical instability, a small value $\theta = 0.001$ is used. Our experiments illustrate that our instant fusion approach obtains equal results as the sigmoid-based fusion, but runs up to 25% faster. In the proposed model, we combine the proposed multidirectional connections and the instant fusion to obtain highly semantic features. Here, we demonstrate the details of the feature fusion in the fifth layer, as illustrated in Fig. 2(f)

$$F^{\text{td}}_5 = \text{Conv}\left(\frac{w_1 . F^{\text{in}}_5 + w_2 . \text{rescale}\left(F^{\text{in}}_6\right)}{w_1 + w_2 + \theta}\right) \quad (20)$$

$$F^{\text{out}}_5 = \text{Conv}\left(\frac{\acute{w}_1 . F^{\text{in}}_5 + \acute{w}_2 . F^{\text{td}}_5 + \acute{w}_3 . \text{rescale}\left(F^{\text{out}}_4\right)}{\acute{w}_1 + \acute{w}_2 + \acute{w}_3 + \theta}\right) \quad (21)$$

in which F^{td}_5 is the transitional feature at the fifth level of the cross route and F^{out}_5 is the result of the fifth layer on the upward route. All the rest of the features are constructed similarly. It is worthy to mention that multiscale divisible

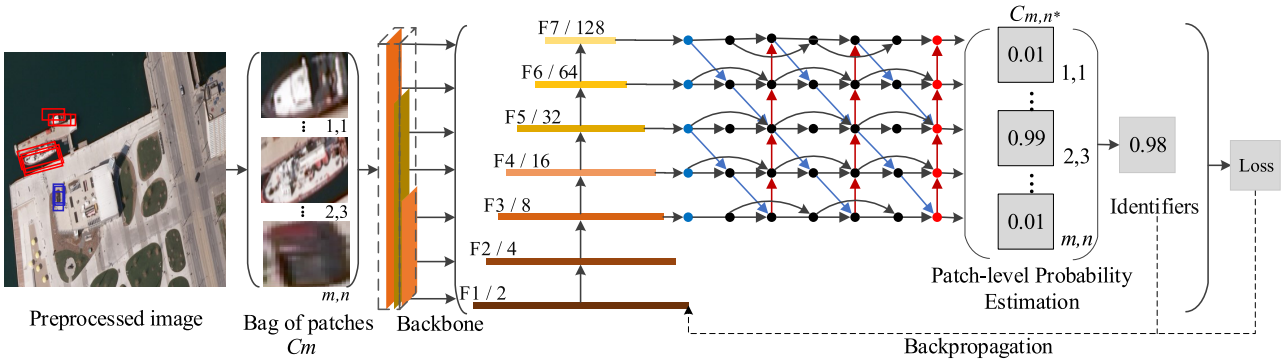

Fig. 3. Proposed network contains: patch detection. It adopts ResNet-50 + SPN [22] as the backbone and ESS-FPN as the multiscale feature network and the class prediction network.

convolutions are adjusted to the network to improve the efficiency.

3) ESS-FPN Architecture: Fig. 3 presents the architecture of ESS-FPN. SPN [22] is used as the backbone. The proposed ESS-FPN acts as the network for multiscale feature extraction that receives the features from the third to seventh layers of the backbone and, constantly, executes top-down, bottom-up, and crosswise feature fusion. To improve both accuracy and efficiency, the current approaches scale up a baseline by adopting larger backbone networks, using a larger dataset, or adding more FPN layers [19], [23], [41]. Such models are generally unproductive as they can only process one or few scaling dimensions (i.e., depth, width, and input resolution). We propose an efficient scaling method for object detection, which adopts an extensive multiplex factor ψ to rescale the entire dimensions of the backbone network. Different from other image classification models [42], object detectors have various scaling dimensions; therefore, searching over all the dimensions is considerably expensive. Thus, we scale up the dimensions by using a heuristic method. We increase the ESS-FPN's depth (layers) D_{essfpn} and the width (channels) W_{essfpn} to improve the proposed detector's efficiency. More precisely, grid search is applied on the bases of {1.15, 1.2, 1.25, 1.3, 1.35, 1.4}, and we choose the premier value 1.3 as the adjustment factor for the width. Conventionally, the width and depth of ESS-FPN are determined as follows:

$$D_{\text{essfpn}} = 3 + \psi, \quad W_{\text{essfpn}} = \frac{2^7}{1.3^{\psi}}. \tag{22}$$

We use feature levels 3–7 in ESS-FPN; therefore, the input resolution should be dividable by $2^6 = 64$, and the following equation is used:

$$\text{Res}_{\text{input}} = \frac{512 + \psi}{64} \tag{23}$$

where, with various ψ, we have performed a wide range of evaluations in order to find the best parameter from $S_0(\psi = 0)$ to $S_7(\psi = 7)$, as listed in Table I, in which S_7 has a higher resolution than S_6. For each processed image patch, the ultimate sigmoid layer will generate a probability distribution that will be used for further patch selection. Compared to the other FPNs [14], [16], in the proposed MPFP-Net inspired by SPN [22], the U-shape network is adopted to build a pyramid network. In this model, the upward path gets the outputs of multiple layers as its reference sets.

TABLE I

SCALING CONFIGURATION FOR ESS-FPN ($S_0 - S_6$) – Φ IS THE COEFFICIENT PARAMETER THAT ADJUSTS THE OTHER SCALING DIMENSIONS

	Input size	W_{essfpn}	D_{essfpn}
$S_0(\psi = 0)$	512	32	3
$S_1(\psi = 1)$	640	64	4
$S_2(\psi = 2)$	768	88	5
$S_3(\psi = 3)$	896	112	6
$S_4(\psi = 4)$	1024	160	7
$S_5(\psi = 5)$	1280	224	7
$S_6(\psi = 6)$	1280	288	8

Furthermore, to enhance the performance and preserve the features' smoothness, 1 × 1 convolution layers are inserted after each up-scaling process. In total, five stacks of U-shape networks are used for building the multiscale features. In our model, after the patch-level estimation, visual domain aggregation is applied to connecting all the detected patch-level probabilities to a detected image map. In the present patch, visual features are first aggregated using a pooling layer and then transformed to a semantic domain.

IV. EXPERIMENTS

In this section, first, we discuss the details of the benchmark datasets for object detection in RSIs and then describe the evaluation metrics, training procedure, and implementation details of our proposed model. Next, we compare the performance of the proposed MPFP-Net with that of several state-of-the-art approaches. MPFP-Net is implemented in PyTorch, and all the experiments are conducted on a workstation equipped with Tesla P40 GPU. To validate that the proposed MPFP-Net can learn effective features for object detection with different appearance variations and scales, the activation values of the classification convolution layers, including scales and level dimensions, are shown in Fig. 4. The input image contains four harbors, two ships, and two cars. The sizes of harbors, ships, and cars are different.

It is worth mentioning that: 1) compared with the smaller harbors, the larger harbors have the larger activation value at the feature map of a large scale, similar to the larger

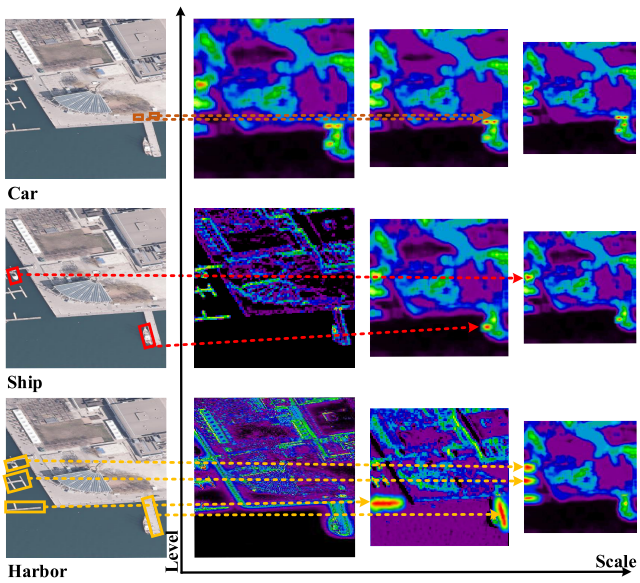

Fig. 4. Examplar activation values of multiscale multilevel features. Best viewed in color.

ship and the smaller one and 2) the smaller harbors and the smaller ship have higher activation values at the feature maps of the same scale. This sample illustrates that MPFP-Net learns effective features to deal with different scales and appearance-complexity across object patches. MPFP-Net is evaluated on three public datasets: NWPU VHR-10 [32], LEVIR [43], and DOTA [44].

A. Dataset

1) NWPU VHR-10: For evaluating the performance of the proposed MPFP-Net model, we use the challenging ten class Northwestern Polytechnical University Very High-Spatial Resolution (NWPU VHR-10) dataset [32]. This dataset contains 650 VHR optical RSIs, in which 565 images were obtained from Google Earth, where each image has the size of 1000 × 1000 pixels with the resolution ranging from 0.5 ∼ 2 m, and 85 pansharpened infrared images with 0.08-m resolution. The dataset includes ten manually annotated classes.

2) LEVIR: This dataset contains 22k high-resolution Google Earth images, where each image has the size of 800 × 600 pixels and a resolution of 0.2 ∼ 1.0 m/pixel [43]. The dataset contains three annotated classes with small objects of 30 × 30 pixels and minimum objects of 10 × 10 pixels.

3) DOTA: It is a large RSIs dataset [44] used for object detection, which comprises 2806 images with different size ranges (800 × 800–4000 × 4000 pixels) and 188 282 instances of 15 categories of objects: plane, baseball diamond (BD), bridge, ground field track (GFT), harbor and helicopter (HC), small vehicle (SV), large vehicle (LV), tennis court (TC), basketball court (BC), storage tank (ST), soccer ball field (SBF), roundabout (RA), and swimming pool (SP). Each image is labeled with an arbitrary quadrilateral.

B. Evaluation Metrics

The target detection results contain two components: bounding boxes (BBs) that enclose the detected targets and the labels. In general, IoU is used as the evaluation metrics in object detection, which denotes the ratio between the estimated object and the ground-truth BBs. The IoU is formulated as follows:

$$\text{IoU} = \frac{(S_{\text{BBs}} \cap S_{\text{GT}})}{(S_{\text{BBs}} \cup S_{\text{GT}})} \tag{24}$$

in which S_{BB} and S_{GT} are the predicted areas and the ground-truth boxes, respectively. Here, we use precision–recall curves and the average value of precision (AP) for a single-object class from recall 0 to 1 to assess the object detection framework, in which a higher AP represents better performance. While having multiclass objects, mAP is used to calculate the average AP of all the classes.

C. Training Procedures and Implementation Details

NWPU-VHR-10 and LEVIR labels are in a conventional axis-aligned BBs form, while DOTA objects' labels are in a quadrilateral form. For adapting the both settings, our proposed MPFP-Net uses both horizontal and oriented BBs (HBB, OBB) as ground truth, where HBB: $\{x_{\min}, y_{\min}, x_{\max}, y_{\max}\}$ and OBB: $\{x_{\text{center}}, y_{\text{center}}, w, h, \theta\}$, where w and h denote width and height, and θ is within $[0, 90°)$ to create ground truth for each object. In training, the OBB ground truth is produced by a group of rotated rectangles that properly overlap with the given quadrilateral labels. For the NWPU-VHR-10 and LEVIR datasets, the MPFP-Net just produces HBB results, as OBB ground truth does not exist in the datasets.

However, for the DOTA, the MPFP-Net produces both HBB and OBB outputs. In the training phase, for the LEVIR and NWPU VHR-10 datasets, we resize the original images to 512 × 512 pixels. For the NWPU VHR-10, the quantity of images is insufficient; to increase the training set, we perform rotation and mirroring. For the DOTA dataset, we split the images into 640 × 640 patches with 200 pixels overlap by the development toolkit. In our experiments, 75% of LEVIR is selected for training and the remaining 25% for testing. Also, we select 60% of NWPU-VHR for training, 10% for validation, and the rest for testing, and 60% of the DOTA dataset for training, 20% for validation, and the remaining for testing. We employ the ResNet-50 + SPN [22] as backbone. For the DOTA and LEVIR datasets, we train the model for 150k iterations with batch size 1 on two Tesla P40 GPUs, which took around 24 h. The initial learning rate is set to 1e-4 and is divided by 10 after every 30k iterations. The weight decay is set to 0.0005, the momentum is 0.9, and the batch normalization and the Swish activation (to enhance the backpropagation) are used after each convolution layer. Adam optimization [48] is used to speed up the training. On the other hand, for the NWPU dataset, we train the model with 30k iterations, and the initial learning rate was set to 1e-3 and changed to 1e-4 and 1e-5 at 10k and 20k iterations, respectively, which took around 3 h. For the NWPU dataset since the number of images is not enough, during training, we also use data augmentation, including rotation and random flip.

In Table II, we compare MPFP-Net with several object detection methods on LEVIR. Our proposed MPFP-Net

TABLE II

MPFP-NET AND THE OTHER MODELS PERFORMANCE ON THE LEVIR DATASET [43]. PARAMS. AND FLOPS (IN BILLIONS) REPRESENT THE NUMBER OF THE PARAMETERS AND MULTIPLY ADDS. LAT SIGNIFIES INFERENCE LATENCY WITH BATCH SIZE 1. THE MODELS THAT HAVE ALMOST EQUAL PERFORMANCE ARE GROUPED

Model	mAP	Params	Ratio	FLOPs	Ratio	GPU LAT(ms)	Speedup	CPU LAT(s)	Speedup
MPFP-Net-S_0 (512)	**62.39**	**4.1M**	**1×**	**3.8B**	**1×**	**19±2.1**	**1×**	**0.35±0.003**	**1×**
LARGE-RAM [43]	62.42	18.3M	4.3×	53B	13.6×	52	2.7×	3.9±0.034	9.6×
MPFP-Net-S_1 (640)	**65.07**	**7.6M**	**1×**	**8.1B**	**1×**	**23±0.8**	**1×**	**0.78±0.004**	**1×**
RIRBM [45]	64.31	67M	8.5×	368B	46.1×	279±0.3	12.1×	57±0.017	63.1×
MPFP-Net-S_2 (768)	**72.55**	**8.7M**	**1×**	**16.3B**	**1×**	**26±1.1**	**1×**	**1.3±0.003**	**1×**
SUCNN [46]	70.11	103M	11.4×	1028B	64.3×	362±0.4	13.8×	57±0.028	35.2×
MPFP-Net-S_3 (896)	**75.71**	**13.4M**	**1×**	**35B**	**1×**	**44.8±0.5**	**1×**	**2.7±0.005**	**1×**
HSF-Net [17]	75.33	152M	11.3×	2389B	68.4×	566±0.8	12.5×	98±0.043	32×
ResNet-50 + NAS-FPN (1280) [16]	73.75	59.6M	4.5×	350B	10×	267±0.4	5.8×	57±0.170	20.2×
ResNet-50 (1280) [23]	–	25.6M	–	125B	–	117±0.3	–	24±0.05	–
MPFP-Net-S_4 (1024)	**79.05**	**21.1M**	**1×**	**56B**	**1×**	**76±0.4**	**1×**	**5.1±0.002**	**1×**
Sig-NMS [32]	78.11	71.7M	3.8×	479B	8.3×	290±1.3	3.8×	60.14±0.007	11.9×
ResNet-50 + NAS-FPN (1280@84) [16]	74.28	72.6M	3.6×	546B	9.5×	327±0.1	4.3×	63.71±0.031	12.5×
MPFP-Net-S_5 (1280)	**83.42**	**34.2M**	**1×**	**142B**	**1×**	**145±2.1**	**1×**	**11.4±0.014**	**1×**
LV-Net [18]	82.34	114M	3.3×	1259B	8.8×	438±1.6	3.2×	84.31±0.034	7.5×
MPFP-Net-S_6 (1280)	**86.73**	**51.2M**	**1×**	**228B**	**1×**	**198±0.7**	**1×**	**17±0.002**	**1×**
HSP [47]	85.32	125M	2.6×	1428B	6.2×	416±1.8	2.25×	85±0.096	4.4×
MS-OPN [33]	85.21	101.6M	2.2×	1019B	4.4×	380±1.5	1.9×	72±0.081	4.2×

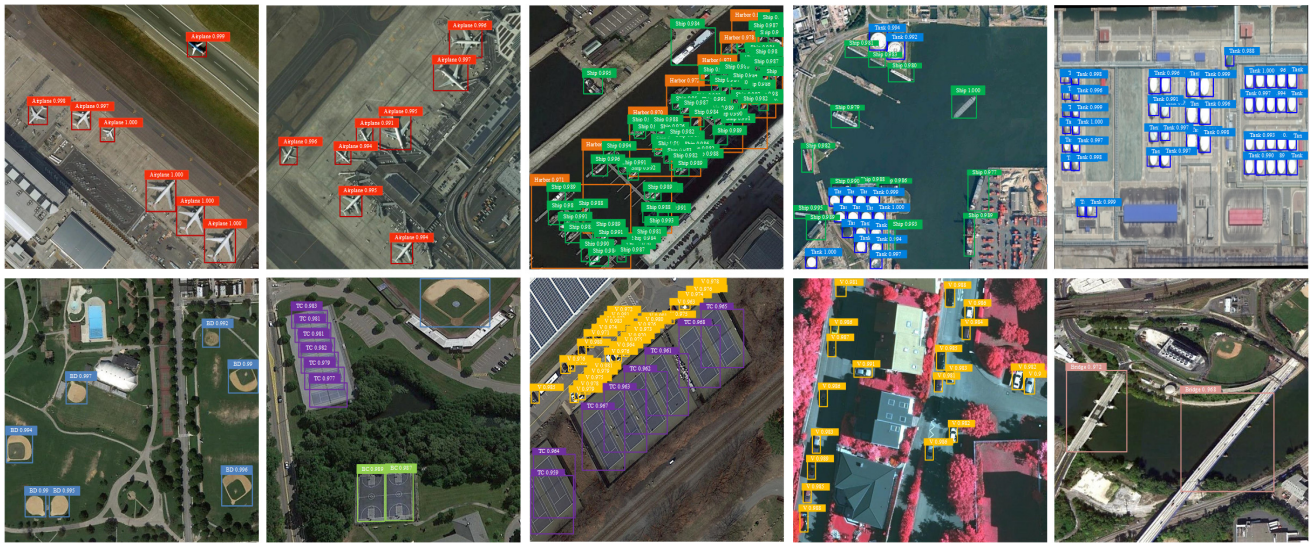

Fig. 5. Selected detection samples by MPFP-Net on NWPU VHR-10. (Zoom in for better vision.)

achieves better accuracy with lower computation costs compared to the other models across a broad range of resource constraints. The MPFP-Net-S_0 generates only 4.1M parameters on top of the backbone parameters and obtains almost the same accuracy as LARGE-RAM [43] with 13.6× fewer FLOPs. In comparison with RIRBM [45], the proposed MPFP-Net-S_1 obtains a better detection rate with up to 46× lower FLOPs and 8.5× less number of parameters. In our proposed model, by increasing the input size from 512 × 512 to 640 × 640, the result is improved with 3 mAP; on the other hand, the model's size and FLOPs are increased by 1.8% and 2.1%, respectively.

D. Comparison With Other State-of-the-Art Methods

Fig. 5 and Fig. 6 present some detected objects by our proposed model on the evaluation datasets.

1) Performance Evaluation on the NWPU VHR Dataset: As clearly seen in Fig. 5, the detected objects are belonging to various classes, for example, airplanes, ships, harbors, and oil tanks. The proposed MPFP-Net has superior performance in object detection, and even the objects that stay very close together are also properly detected. For the evaluation, other state-of-the-art RS detection approaches, such as Sig-NMS [32], HSF-Net [17], LV-Net [18], MS-OPN [33], and HSP [47], are applied to the NWPU VHR-10, and their object detection performance are compared with that of MPFP-Net. Table III lists the HBB detection results. As the results show, the MPFP-Net has superior performance in comparison with the other methods and achieves better results in the airplane and ship classes. The airplane and ship classes contain tiny targets, indicating that our proposed method performs satisfactorily in detecting small objects. Our proposed MPFP-Net retains better mAP with much fewer

Fig. 6. Example of detection results by the MPFP-Net on DOTA.

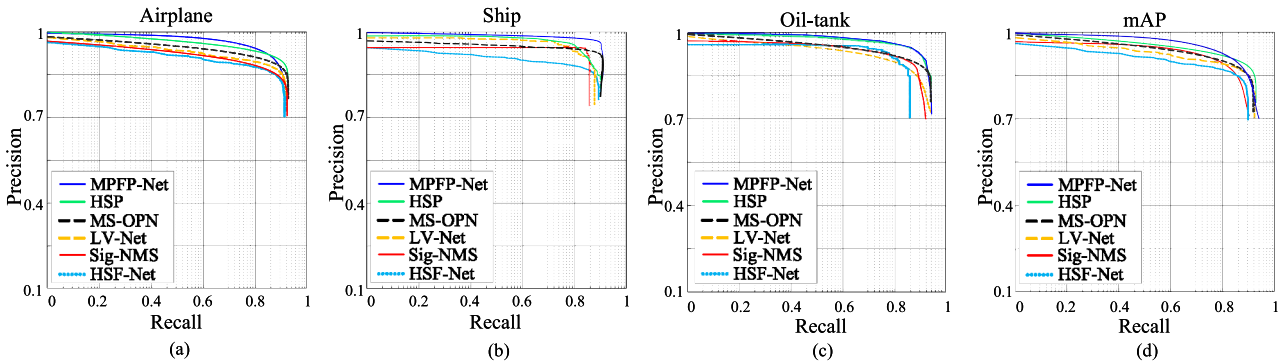

Fig. 7. Precision–recall comparisons between MPFP-Net and other methods on the LEVIR dataset, for the classes of (a) airplane, (b) ship, (c) oil-tank, and (d) mean AP.

TABLE III

PERFORMANCE COMPARISON FOR THE HBB TASK ON THE NWPU-VHR-10. THE BEST RESULTS ARE BOLD

	HSF-Net	Sig-NMS	LV-Net	MS-OPN	HSP	MPFP-Net
Airplane	88.64	90.84	93.40	99.74	99.80	**99.84**
Ship	80.54	80.58	81.80	89.02	92.47	**92.63**
ST	59.13	59.25	73.57	80.13	**96.99**	96.98
BD	88.35	90.89	98.35	98.10	**98.58**	98.49
TC	79.38	80.86	84.89	86.09	**90.39**	89.83
BC	89.87	90.94	88.86	**92.57**	91.49	91.96
GTF	96.41	**99.85**	96.87	98.65	99.08	99.73
Harbor	81.81	90.39	91.64	94.61	88.93	**94.82**
Bridge	65.59	67.86	82.91	**94.37**	87.11	92.30
Vehicle	78.89	78.16	79.87	82.23	89.09	**89.15**
mAP	80.84	82.93	87.19	91.53	93.40	**94.57**
Parameters	153M	71.7M	115M	103M	126M	**52.3M**
FLOPs	2389B	479B	1270B	1022B	1525B	**230B**

parameters and FLOPs compared to the other approaches on the NWPU-VHR-10 dataset.

2) Performance Evaluation on LEVIR: Detailed object detection comparisons between MPFP-Net and the other state-of-the-art models on the LEVIR dataset are reported in Table IV. It is clear that MPFP-Net shows the best detection results in all the classes. In particular, our model surpasses HSP and MS-OPN in airplane and ship classes,

TABLE IV

PERFORMANCE COMPARISON FOR THE HBB TASK ON THE LEVIR DATASET

	HSF-Net	Sig-NMS	LV-Net	MS-OPN	HSP	MPFP-Net
Airplane	81.04	86.83	82.58	85.09	86.92	**87.24**
Ship	77.29	79.42	80.57	83.76	83.75	**85.57**
Oil-tank	67.62	68.28	83.59	86.80	**87.42**	87.35
mAP	75.33	78.11	82.34	85.21	85.32	**86.73**
Parameters	152M	71.7M	114M	101.6M	125M	**51.2M**
FLOPs	2389B	479B	1259B	1019B	1428B	**228B**

which has a large number of small objects, and this indicates that MPFP-Net is able to successfully detect small objects. Fig. 7 shows the PRCs of the three classes of LEVIR and mean AP by HSF-Net, Sig-NMS, LV-Net, MS-OPN, HSP, and MPFP-Net, respectively. The blue curves that represent the performance of our proposed model are higher or similar to those of different classes.

3) Performance Evaluation on DOTA: In Table V, we evaluate the HBB detection performance of MPFP-Net on 15 classes of the DOTA dataset in comparison with the other approaches and Fig. 6 presents some detected objects by our proposed model on this dataset. The results demonstrate that the

TABLE V

PERFORMANCE COMPARISON FOR THE HBB TASK ON THE DOTA DATASET

	HSF-Net	Sig-NMS	LV-Net	MS-OPN	HSP	MPFP-Net
Plane	82.34	83.67	85.38	88.61	90.36	**90.49**
BD	79.66	80.29	84.63	86.11	**86.85**	86.79
Bridge	48.74	50.12	52.81	54.76	62.51	**62.68**
GTF	76.38	77.51	79.49	**79.96**	79.83	79.71
SV	65.42	68.37	71.64	75.29	78.07	**78.22**
LV	64.92	69.74	70.83	78.34	81.79	**81.97**
Ship	75.37	77.46	80.72	83.17	**85.27**	85.24
TC	84.68	86.69	88.48	**90.91**	90.82	90.88
BC	79.42	81.45	84.39	86.12	**87.24**	87.21
ST	74.19	79.76	81.73	84.96	85.90	**85.98**
SBF	60.33	61.94	65.75	68.58	69.93	**70.18**
RA	65.53	68.44	70.51	71.27	**72.08**	72.02
Harbor	69.58	71.45	73.40	76.12	84.11	**84.24**
SP	68.92	68.64	70.85	71.59	80.92	**81.13**
HC	64.60	66.39	66.42	67.77	**69.81**	69.78
mAP	70.67	72.65	75.23	77.38	80.36	**80.43**
Parameters	155M	72.8M	116M	105M	128M	**54.8M**
FLOPs	2394B	481B	1286B	1028B	1529B	**242B**

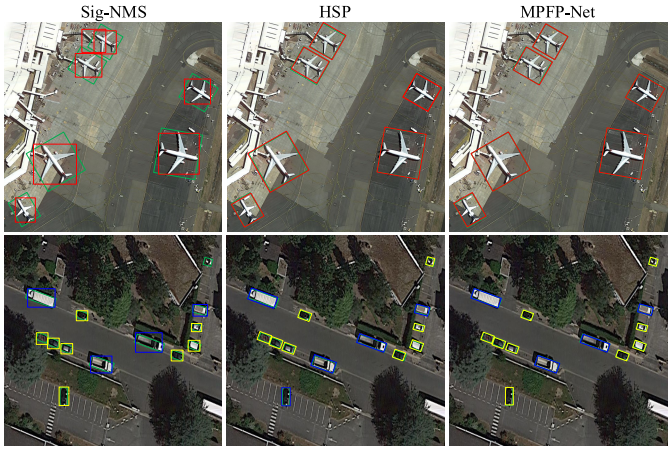

Fig. 8. Qualitative detection comparison by different models on the DOTA dataset. The green boxes show ground truth. The red, blue, and yellow boxes indicate the detected planes, LVs, and SVs, respectively.

proposed method achieves state-of-the-art results and comparable performance with the other object detection approaches.

To show the advantages of MPFP-Net over the other methods, we perform a qualitative comparison of different methods on the DOTA dataset. The results are shown in Fig. 8. The results show the other approaches cannot robustly detect the objects in images, and the background is misdetected as the foreground. Moreover, in the other methods, the BBs are not well fit the detected objects, whereas Sig-NMS [32] only generates the horizontal box. However, our method can stably produce precise results.

E. Experiment on Large-Scale Images

In Fig. 9, we test MPFP-Net performance on a large-scale RSI. It is observed that the pretrained MPFP-Net has an adequate flexibility on different image sources and conditions, which shows the effect of multiple patches and FP learning.

V. ABLATION STUDY

This section ablates different components of MPFP-Net on the LEVIR test set. We evaluate the case of different parameter

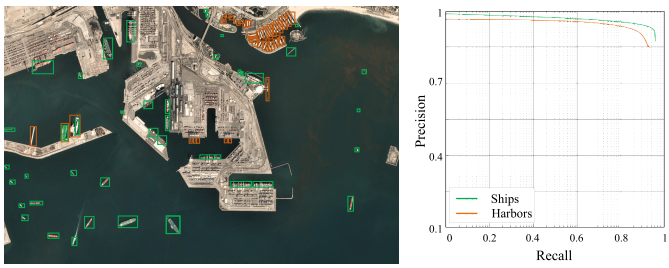

Fig. 9. Detection results on a large-scale RSI from Geomatica-CGI System. Green and orange, respectively, show ships and harbors.

TABLE VI

PROPOSED MODEL EVALUATION WITH DIFFERENT BACKBONES

Configuration	mAP	Parameters	Flops
MPFP-Net-S_6+ FPN	78.41	122M	1253B
MPFP-Net-S_6+ SPN	83.69	84.9M	387B
MPFP-Net-S_6+ SPN+ ESS	86.73	51.2M	228B

sizes, FLOPs, and latency on the P40 GPU. Each model is run five times with the batch size of 1, and the mean and standard deviation are reported in Table II. Fig. 10 shows the model's size, FLOPs, and GPU latency. In comparison with the other models, MPFP-Net is up to 3.7× faster. We conduct a set of evaluations to measure the contribution of the backbone network and ESS-FPN to the detection accuracy and efficiency improvements of the proposed MPFP-Net. Table VI reports the impact of each module on the overall performance of our proposed model. Starting from MPFP-Net-S_6 with FPN [14] as the backbone, first, instead of FPN, we adopt the SPN [22], which improves the detection accuracy by more than 5 mAP with a smaller number of parameters and FLOPs. Later, we add the proposed ESS-FPN, where the detection performance is improved by 3 mAP with fewer parameters and FLOPs.

A. Instant Fusion Versus Softmax and Dynamic Fusion

As earlier discussed, we propose an instant feature fusion approach that preserves the benefits of the normalized weights while reducing the softmax computation cost. In Fig. 11(a), we examine the performance of MPFP-Net while adopting dynamic fusion, softmax fusion, and the proposed instant fusion. The proposed instant fusion achieves accuracy and learning behavior similar to the softmax fusion and, however, runs 1.24× faster. During the training, the normalized weights quickly change, suggesting that various features unequally contribute to the feature fusion.

B. Multipatch and Combined Multiscaling

As discussed in Section III-B, we propose multiple-patch learning to deal with the lack of full object instance labeling and propose a combined multiple scaling method to increase all the dimensions of ESS-FPN for better learning and consequently increasing the detection performance of our proposed model. In Fig. 11(b), we compare our approach with the other models that only use patch learning or scale up a single dimension (resolution/depth/width). As the results illustrate,

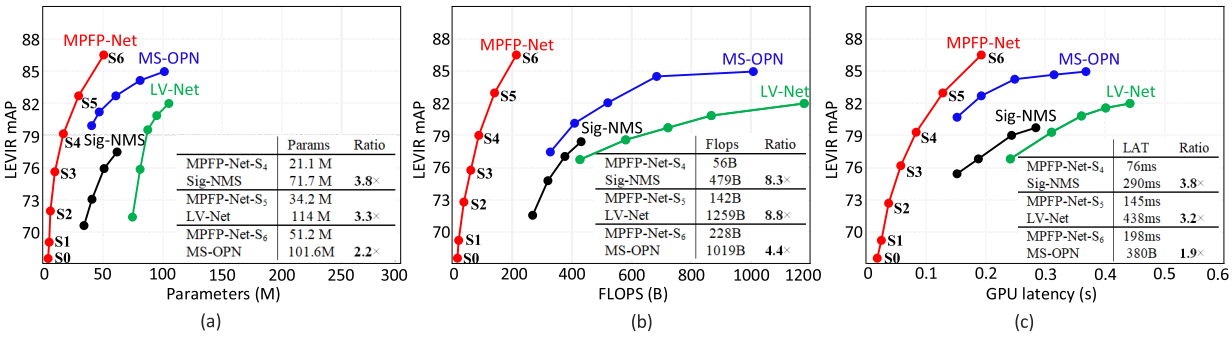

Fig. 10. Comparison with different model sizes and inference latencies. Latency is measured with batch size 1 on the machine equipped with a P40 GPU. The proposed MPFP-Net models are 2.2× ∼ 3.8× smaller and 1.9× ∼ 3.7× faster than the other detectors. (a) Model size. (b) Flops. (c) GPU latency.

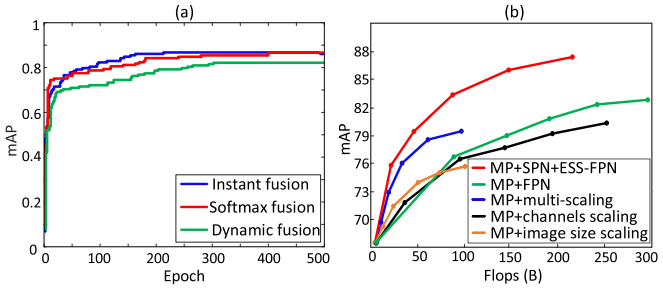

Fig. 11. (a) Instant fusion versus softmax fusion and dynamic fusion. (b) Comparison of different multipatch scaling methods.

our model results in better efficiency compared to the other baselines, which signifies the advantages of patchwise and jointly scalewise learning.

VI. Conclusion

In this article, we have proposed a novel weakly supervised model for detecting multiscale objects in RSIs using a multipatch FPN. First, we integrated automatic patch selection, feature aggregation, and semantic domain projection within a single unified framework. Second, we proposed a weighted FPN that uses multidirectional connections for fast and efficient scalewise feature fusion to further optimize object detection in RSIs. Moreover, a joint loss was used to train the whole network end-to-end. On the basis of these methodologies, we introduced a new detector, named MPFP-Net, which obtains better accuracy and efficiency than the other state-of-the-art methods on RSIs. To evaluate the performance of MPFP-Net for multiscale object detection, three publicly available datasets were used. On these datasets, we evaluated the performance of our proposed method compared with several CNN-based object detection models. Experimental results demonstrate that MPFP-Net can effectively and efficiently detect multiscale objects.

References

[1] B. Kalantar, S. B. Mansor, A. A. Halin, H. Z. M. Shafri, and M. Zand, "Multiple moving object detection from UAV videos using trajectories of matched regional adjacency graphs," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5198–5213, Sep. 2017.

[2] C. Marin, F. Bovolo, and L. Bruzzone, "Building change detection in multitemporal very high resolution SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2664–2682, May 2015.

[3] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.

[4] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[5] D. Hong, N. Yokoya, J. Chanussot, and X. X. Zhu, "An augmented linear mixing model to address spectral variability for hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1923–1938, Apr. 2019.

[6] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 245–284, 2015.

[7] W. Zhao, X. Hou, X. Yu, Y. He, and H. Lu, "Towards weakly-supervised focus region detection via recurrent constraint network," *IEEE Trans. Image Process.*, vol. 29, pp. 1356–1367, Sep. 2019.

[8] P. Shamsolmoali, M. Zareapoor, H. Zhou, and J. Yang, "AMIL: Adversarial multi-instance learning for human pose estimation," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 1s, pp. 1–23, Apr. 2020.

[9] W. Zhang, X. Sun, K. Fu, C. Wang, and H. Wang, "Object detection in high-resolution remote sensing images using rotation invariant parts based model," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 74–78, Jan. 2014.

[10] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 806–813.

[11] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[12] D. Hong, X. Wu, P. Ghamisi, J. Chanussot, N. Yokoya, and X. X. Zhu, "Invariant attribute profiles: A spatial-frequency joint feature extractor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 3791–3808, Jun. 2020.

[13] H. Guo, X. Yang, N. Wang, B. Song, and X. Gao, "A rotational libra R-CNN method for ship detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5772–5781, Aug. 2020.

[14] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[16] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7036–7045.

[17] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, "HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7147–7161, Dec. 2018.

[18] C. Li, R. Cong, J. Hou, S. Zhang, Y. Qian, and S. Kwong, "Nested network with two-stream pyramid for salient object detection in optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9156–9166, Nov. 2019.

[19] J. Yang, J. Guo, H. Yue, Z. Liu, H. Hu, and K. Li, "CDnet: CNN-based cloud detection for remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 6195–6211, Aug. 2019.

[20] D. Hong *et al.*, "More diverse means better: Multimodal deep learning meets remote-sensing imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4340–4354, May 2021.
[21] N. Yokoya, C. Grohnfeldt, and J. Chanussot, "Hyperspectral and multispectral data fusion: A comparative review of the recent literature," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 2, pp. 29–56, Jun. 2017.
[22] P. Shamsolmoali, M. Zareapoor, H. Zhou, R. Wang, and J. Yang, "Road segmentation for remote sensing images using adversarial spatial pyramid networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4673–4688, Jun. 2021.
[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.
[25] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[27] Y. Ding *et al.*, "Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification," 2020, *arXiv:2002.03353*. [Online]. Available: http://arxiv.org/abs/2002.03353
[28] S. Qiu, G. Wen, Z. Deng, Y. Fan, and B. Hui, "Automatic and fast PCM generation for occluded object detection in high-resolution remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1730–1734, Oct. 2017.
[29] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020.
[30] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019.
[31] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
[32] R. Dong, D. Xu, J. Zhao, L. Jiao, and J. An, "Sig-NMS-based faster R-CNN combining transfer learning for small target detection in VHR optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8534–8545, Nov. 2019.
[33] Z. Deng, H. Sun, S. Zhou, J. Zhao, L. Lei, and H. Zou, "Multi-scale object detection in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 3–22, Nov. 2018.
[34] F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye, "Min-entropy latent model for weakly supervised object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1297–1306.
[35] H. Oh Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell, "On learning to localize objects with minimal supervision," 2014, *arXiv:1403.1024*. [Online]. Available: http://arxiv.org/abs/1403.1024
[36] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2846–2854.
[37] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2843–2851.
[38] F. Wan, C. Liu, W. Ke, X. Ji, J. Jiao, and Q. Ye, "C-MIL: Continuation multiple instance learning for weakly supervised object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2199–2208.
[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[40] Y. Hu, Y. Chen, X. Li, and J. Feng, "Dynamic feature fusion for semantic edge detection," 2019, *arXiv:1902.09104*. [Online]. Available: http://arxiv.org/abs/1902.09104
[41] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10781–10790.
[42] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4780–4789.
[43] Z. Zou and Z. Shi, "Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1100–1111, Mar. 2018.
[44] G.-S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.
[45] W. Diao, X. Sun, X. Zheng, F. Dou, H. Wang, and K. Fu, "Efficient saliency-based object detection in remote sensing images using deep belief networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 2, pp. 137–141, Feb. 2016.
[46] Y. Hu, X. Li, N. Zhou, L. Yang, L. Peng, and S. Xiao, "A sample update-based convolutional neural network framework for object detection in large-area remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 6, pp. 947–951, Jun. 2019.
[47] C. Xu, C. Li, Z. Cui, T. Zhang, and J. Yang, "Hierarchical semantic propagation for object detection in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4353–4364, Jun. 2020.
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980