Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 2920062, 15 pages
https://doi.org/10.1155/2021/2920062

Research Article

Self-Recurrent Learning and Gap Sample Feature Synthesis-Based Object Detection Method

Lvjiyuan Jiang,[1,2] **Haifeng Wang**,[1,2] **Kai Yan**,[1,2] **Chengjiang Zhou**,[1,2] **Songlin Li**,[3] **Junpeng Dang**,[3] **Rong Chang**,[3] **Jie Peng**,[3] **Yanbin Fang**,[3] **Chenkai Dai**,[3] **and Yang Yang**[1,2]

[1]*School of Information Science and Technology, Yunnan Normal University, Kunming 650500, China*
[2]*Laboratory of Pattern Recognition and Artificial Intelligence, Yunnan Normal University, Kunming 650500, China*
[3]*Yuxi Power Supply Bureau, Yunnan Power Grid Corporation, Yuxi 653100, China*

Correspondence should be addressed to Songlin Li; lisonglin48@gmail.com and Yang Yang; yyang_ynu@163.com

Received 21 August 2021; Revised 7 September 2021; Accepted 14 September 2021; Published 29 September 2021

Academic Editor: Xin Tian

Object detection-based deep learning by using the looking and thinking twice mechanism plays an important role in electrical construction work. Nevertheless, the use of this mechanism in object detection produces some problems, such as calculation pressure caused by multilayer convolution and redundant features that confuse the network. In this paper, we propose a self-recurrent learning and gap sample feature fusion-based object detection method to solve the aforementioned problems. The network consists of three modules: self-recurrent learning-based feature fusion (SLFF), residual enhancement architecture-based multichannel (REAML), and gap sample-based features fusion (GSFF). SLFF detects objects in the background through an iterative convolutional network. REAML, which serves as an information filtering module, is used to reduce the interference of redundant features in the background. GSFF adds feature augmentation to the network. Simultaneously, our model can effectively improve the operation and production efficiency of electric power companies' personnel and guarantee the safety of lives and properties.

1. Introduction

Deep learning-based object detection [1] has made remarkable achievements and has become a common method for object detection. Among these methods, the basic component of multiscale recognition systems adopts the feature pyramid model. However, object detection poses many challenging problems, such as small objects missing features and background noises in complex scenes. In the field of electric power inspection and maintenance, neural network [2] still experiences problems when detecting specific staff, safety helmet, safety belt, the use of equipment, and other objects. Feature fusion has achieved remarkable breakthroughs in these aspects. It is an important method for improving segmentation performance in which the fusion of multiscale features is extracted from the backbone network. The obtained information contains different scales and spatial locations. Feature fusion improves the detection performance by combining the detection results from different layers. Information propagation in neural network is extremely important, and feature fusion can realize information transmission among multilevel features extracted from backbone networks [3].

Feature fusion solves the problem of low-level and high-level features information asymmetry. Low-level features have higher resolution and contain more locations and detailed information, but they have less semantic information. High-level features have stronger semantic information, but they have low resolution and poor perception of details. Many recent studies have been conducted to explore this problem. Feature pyramid network (FPN) and its variants have made important contributions to feature fusion [4–6]. Feature pyramid or multilevel feature fusion has become an important method for integrating multiscale features in object detectors in recent years. Previously, many object detectors use multiscale features extracted directly

from the backbone network. FPN has made remarkable contributions to the field of feature fusion. It utilizes the inherent multiscale and pyramidal hierarchy of deep convolutional neural networks. Therefore, multiscale high-level semantic feature maps are constructed via a top-down approach. The horizontal linkage and features of different scales are successively combined. The advantage of the pyramidal structure is that each layer of features produces semantic information of varying intensity, including features in low layers with high resolution.

We briefly review relevant studies on object detection feature fusion as follows:

(a) Two feature fusion methods of object detection.

The large number of studies in object detection improves detection and segmentation performance by integrating multiple layers. Early fusion and late fusion [7] are classified in accordance with the sequence of fusion and prediction. The feature of early fusion is to fuse multiple features first and then train the predictor on the fused features. Early fusion is detected uniformly only after complete fusion. This type of method is also known as skip connection, that is, operations that take concat and add. The representative of this approach is Inside Outside Net [8] and HyperNet [9]. Concat uses serial feature fusion to connect two features directly. Add uses a parallel strategy to combine two eigenvectors into a complex vector late fusion which improves performance by combining the detection results of multiple layers. In contrast with early fusion, late fusion is detected at partial fuse before the final fusion is completed. The network will have multiple layers of detection, and finally, the multiple detection results will be fused. Late fusion also contains two representative research ideas. The first method involves performing prediction with multiscale features and then integrates the prediction results, such as Single Shot MultiBox Detector [10] and MS-CNN [11]. The second method inputs the feature pyramid for fusion and then makes a prediction after fusion, such as FPN.

(b) FPN method.

The combination of features that reflect the information of multiple dimensions is generated using images of multiple dimensions as input. This method has higher requirements for hardware computing power and memory size. FPN constructs feature pyramids that can reduce the consumption of computation and video memory. Three basic processes are involved: (1) generation of features of multiple dimensions from the bottom-up pathway; (2) feature enhancement from the top-down pathway; (3) the correlation expression between features of a CNN [12] network layer and the final output of each dimension. Ordinary CNN features are concentrated to express features layer by layer from the bottom-up pathway. The lower layer reflects the shallow level of image information, whereas the higher layer reflects the deeper level of image features. The feature map of the last level of each level is selected, and the information of the previous level is used as the reference for its input when processing the information of each level. Through upsampling, a high-level feature map is enlarged to the same size as the feature map of the previous stage. The upsampling method is the nearest neighbour interpolation. This method can preserve the semantic information [13] of the feature map to the greatest extent during the sampling process, integrating the feature map with the corresponding feature map with rich spatial information.

(c) Related variants of FPN.

Since it was proposed, FPN has been only used in the top-down pathway integration. To promote better transmission of information, a new feature fusion structure based on FPN was proposed. PANet is the first model to propose bottom-up secondary fusion. Based on FPN [14], PANet [15] adds a bottom-up path to increase the low-level information of deep features and shorten the information path. Libra R-CNN [16] integrates all levels of features to generate more balanced semantic features that enhance the original features. The proposed PANet proves the effectiveness of two-way fusion [17], and the two-way fusion of PANet is relatively simple. Therefore, the number of studies has further adopted the direction of FPN and attempted to develop more complex two-way fusion, such as AsFF [18], NAS-FPN [19], and BiFPN [20]. ASFF is the model of adaptive fusion by learning adaptive weights when integrating multiple-scale features. This model focuses on the best approach for integrating information from multiple-scale features. NAS-FPN and BiFPN search for a valid block in FPN and then repeat the stack. A new structure is added to the output based on FPN or the extracted multilevel feature map is reprocessed; thus, information transmission focuses on the resolution of adjacent and other features to provide the fused features with balanced information from each feature resolution.

(d) Feature fusion of the iterative recurrent method.

Contemporary object detection methods follow two paradigms: two-stage and one-stage. Compared with the first stage, the second stage can locate information more accurately. R-CNN uses a certain scale feature map to generate multiscale anchors and detect multiscale objects. Faster R-CNN [21] builds on the previous work and uses a deep convolutional network to improve the efficiency of object classification. FPN is also a classic two-stage algorithm. Many modern object detectors achieve excellent performance by adopting the mechanism of looking and thinking twice [22]. Furthermore, a new feature fusion structure-based FPN is proposed [23–25]. This structure, called a recursive feature pyramid (RFP) [26], is designed to integrate feedback links

into the backbone network and enhance the entire network with precise positioning signals. RFP explores this mechanism and designs it into an object detection model. On top of FPN, the output object feedback from FPN is connected to the top-down backbone network by incorporating a recursive method to enrich FPN and generate increasingly powerful representations. By expanding the recursive structure to a sequential implementation, the object detection model looks at the images two times or more, during which the backbone network and FPN run several times to help output features related to the previous features.

(e) Data augmentation improving the performance of object detection.

Image augmentation [27] technology makes a series of random changes to the training image to generate similar but different training samples, enlarging the size of the training dataset. Large datasets are the prerequisite for the successful application of deep neural networks. Another interpretation of image augmentation is that randomly changing the training sample can reduce the dependence of the model on certain attributes, thus improving the generalization capability of the model. To solve the problem of data scarcity [28], an increasing number of researchers are using data augmentation [29] technology, that is, the technique of artificially expanding the training dataset with annotation. At present, nearly all most advanced image classification models use data augmentation, and this technique is being increasingly used in other research areas, such as natural language processing [30]. Supervised data augmentation [31] can be divided into single-sample and multiple data augmentation. In single-sample data augmentation [32], all operations are performed around a sample itself when this sample is enhanced. A geometric transformation class performs geometric transformation on images, including flipping, rotating, cutting, deformation, scaling, and other operations. Flipping and rotating do not change the size of an image, whereas cropping does. A random clipping method is generally used during training.

From the preceding research, the following problems still exist:

(1) Problems with the integration of the looking and thinking twice mechanism and object detection.

At present, RFP combines looking and thinking twice mechanism with the best effect among object detection models. RFP adds feedback and a recursive repeating stack based on FPN and the backbone network, and it enhances the entire functional hierarchy through feedback connection. However, compared with FPN, the backbone network involves multilayer convolution operation, and the iterative recurrent backbone network can inevitably lead to a huge amount of computation. Under the premise of time consumption, certain equipment support is also required. In addition, a multilayer convolution operation can cause the loss of the original image information while changing the size of the feature image. The output of the feature pyramid is the feature map of the captured object and it feeds back to the backbone network. However, the unprocessed feature background may confuse the network and make distinguishing the important information in the foreground difficult.

(2) The problem of features redundancy.

The effect of pyramid structure is reflected in the sharing of information at high and low levels, and the semantic gaps between the multilevel features complement one another during information transmission. Therefore, the efficiency of information transmission exerts a certain influence on the result of feature fusion. FPN uses convolution compression feature maps. ASFF increases the adaptive weight but only simply fuses the multistage feature image with the assigned weight. PANet adds a bottom-up path after the feature pyramid, but it directly selects the low-level features of the backbone network and integrates it with the high-level features. Although these methods can effectively transmit information, the efficiency of information transmission is not high. RFP second capture features can effectively expand the scope of information transmission. However, the redundant features [33] can increase exponentially while improving object detection performance. The background information of a feature map contains the large number of redundant features that can considerably reduce the feasibility of object detection and extraction. Background with much distracting information can confuse the network, making it unable to determine the object. The information required for object detection is favorable information. The purpose of filling the information gap is to reduce the difficulty in accurately locating information in a high-level feature map and to increase the semantic information of a low-level feature map with higher resolution. In the process of information transmission, if redundant information spreads along the path of information transmission, then it will not only produce a large number of invalid calculations but also pollute the feature map.

(3) Simple sample features that can also cause problems.

The success of deep learning in computer vision can be partly attributed to having large amounts of tagged training data; when the complexity of data increases, the performance of the model generally

increases. Model training typically relies on a large amount of label data, but samples with simple features only contain a small number of labels, resulting in poor performance of the model. However, collecting enough high-quality data to train a model for good performance is frequently difficult. Data augmentation can significantly improve the normalization performance of deep learning, particularly in the aspects of image classification and object detection. Cubuk et al. [34] add the data enhancement model in the preprocessing part and it can effectively compensate for the defects brought by simple samples. However, a large dataset frequently causes a considerable computational loss obstacle.

To address the above issues, we design a self-recurrent learning and gap sample feature fusion-based object detection model to solve the aforementioned problems. This model includes three modules: SLFF, REAML, and GSFF.

The main contribution of this study is as follows:

(i) The SLFF is proposed to effectively combine the looking and thinking twice mechanism with the feature fusion structure to minimize the amount of computation and realize recurrent feature fusion, enabling the feature fusion network to acquire the capability of self-adaptive learning.

(ii) REAML learns multilevel feature information, bridges the information gap, and reduces the interference of redundant information through mutual learning among multichannel features.

(iii) GSFF combines single feature samples to obtain complex feature samples through feature fusion among gap samples, enabling the model to increase its recognition and generalization capabilities in the process of training the network.

(iv) In real scenarios, the combination of SLFF, REAML, and GSFF realizes the organic combination of artificial intelligence technology and power safety technology.

2. Materials and Methods

Our method is a feedback connection model of a feature pyramid based on the looking and thinking twice mechanism proposed by a modern object detector to address the following problems: feature filtering and sample feature augmentation within a network. The overall framework self-recurrent learning and gap sample feature fusion-based object detection are shown (see Figure 1).

We modify and improve RFP to shorten the information path while reducing redundancy features. The details are as follows: the SLFF module relearns the features of the multiscale feature diagram of the FPN output and then reduces redundant feature interference through REAML. Simultaneously, we add a sample feature enhancement mechanism to the network to improve model efficiency through object sharing among different images. In the following sections, we describe how the feature fusion model is implemented with an iterative mechanism.

2.1. Self-Recurrent Learning-Based Feature Fusion. RFP enables tasks to process sequential information, such that the input before and after the correlation forms a recurrent structure, highlighted. The model effect can be improved by clearer and richer features obtained from the multiple iterations of image convolution. However, one of the paradoxes of a feature pyramid that includes feedback connections is that it extends the scope of the looking and thinking twice mechanism. Repeat iterations of the backbone network cannot only produce a huge amount of computation but also weaken the advantages of the looking and thinking twice mechanism. Such an approach not only lengthens the information path but also multiplies redundant features.

In the self-recurrent learning-based feature fusion model, we avoid the backbone network participating in the iterative operation and only retain the function of the feature pyramid part. Whether the feature pyramid structure can extract features more effectively depends not only on the structure design but also on the input of the model. High-quality feature maps are frequently easier to enhance and extracted through repeated operations of a pyramid structure. A valuable goal is an explicit goal in the image. We aim to make feature fusion tend to incorporate dominant features. To improve the effect of feature fusion, we put three layers of feature maps at different scales output from the feature pyramid into the pyramid structure again (see Figure 2).

RFP is self-recurrent learning based on the backbone. The output of the first feature pyramid is used as the input to carry out multiple convolution calculations. Finally, the feature maps of different scales are detected, respectively. SLFF is further improved on the self-recurrent structure. In accordance with the original structure, self-recurrent learning uses the output of the backbone network as the input of FPN to obtain the multiscale feature map.

Through REAML, the blank information is used to obtain the feature f_i^t, and then feedback is connected to the top-down pyramid structure. After secondary propagation, the output feature is defined as follows:

$$\mathrm{Sp} = W^n F^n(X) + \sum_{i=1}^{n} w^i B. \tag{1}$$

This formula is a recursive operation. $F^n(X)$ represents the calculation of feature fusion. $n \in [1, \ldots, N]$, and it is the number of times of recurrent expansion of the feature pyramid. W is composed of the weighted w^i of multilevel feature number i. B represents a learnable constant. After the calculation of the preceding formula, the output result Sp is obtained.

Compared with the RFP structure, features are extracted two times but not through the backbone network. We carry out self-recurrent based on FPN. On the one hand, the multiple feature capture aims to find out the missing features in the background image; on the other hand, it aims to enhance the identification of dominant features. The output of FPN has

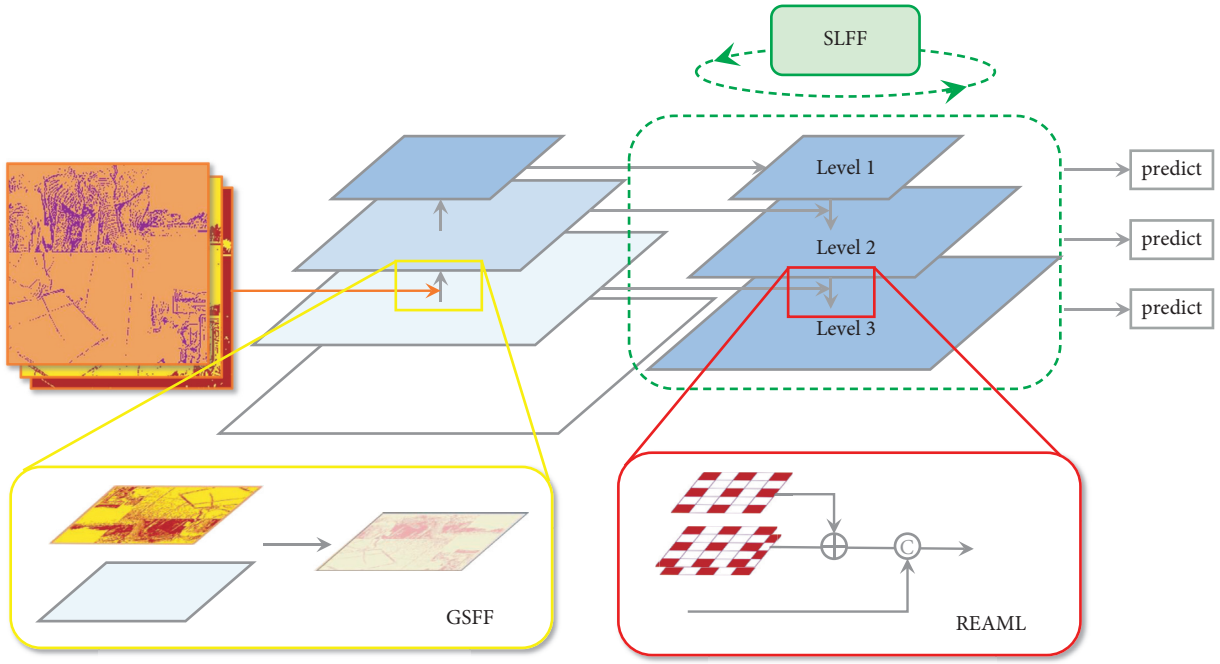

Figure 1: The proposed framework includes three modules: self-recurrent learning-based feature fusion (SLFF), residual enhancement architecture-based multichannel (REAML), and gap sample-based feature fusion (GSFF). *Note.* ©, channel splicing operation; ⊕, feature fusion for automatic weight allocation.

more salient features, and these salient features are obtained by calculating and processing FPN. These processes are more conducive to extract. Through a top-down path and horizontal connection process of the feature pyramid, images realize the function of cross-spreading information among multiscale feature maps. It makes the model more inclined to mine the missing features in the background to enhance the dominant features. The nature of SLFF is generally to construct a recursive operation that does not involve a backbone network. Thus, our model is simpler but produces better results.

2.2. Residual Enhancement Architecture-Based Multichannel Learning. The second capture of the feature map not only improves the efficiency of a feature but also leads to the multiplication of redundant features. A redundant feature is different from a recessive feature. Recessive feature points are the effective information that is difficult to be captured by a neural network, and they contain exploitable feature points that can be mined. By contrast, redundant features exist in the background of the feature image, reducing the ratio between dominant and recessive features. The background contains a variety of features with different information contents, some of which may interfere with the calculation. Background information with low sensitivity also contains considerable distracting information, which confuses the network and makes it unable to judge objects.

When feature maps have enhanced features but more redundant features are reentered into the network, the feature pyramid will have lower sensitivity, and the network structure will be confused about which should be enhanced to extract which. Therefore, we design REAML to realize information transmission while reducing redundant feature interference. The model formula is as follows:

$$F^{t+1} = \sum_{i=1}^{n} R\left(D\text{conv}\left(F_i^t\right), F_i^t\right). \tag{2}$$

REAML retains the output F_i^t of the first FPN, n represents the feature map of the i th layer that we extracted, and d is set as the number of all layers outputted from the pyramid of FPN. F_i^t contains information that can be complementary to F^{t+1}. Dconv is a 3×3 dilated convolution [35] with two dilatations. R represents the calculation of REAML. After size adjustment of dilated convolution, F_i^t achieves a larger receptive field than ordinary convolution. F_i^t is based on F^{t+1} learning weights. We operate filters to filter the information of each layer's features and integrates the filtered results in accordance with the degree of contribution of the features. To realize REAML, the implementation of the residual function is shown in the following. $\Delta\rho$ is used as the residual structure in the model. $n \in [1, \ldots, n]$, and $p^{(i)}$ represents the i-th channel of the feature map. $w^{(i)}$ and $b^{(i)}$ represent the weight and self-learning constant of the channel, respectively.

$$\begin{aligned} x &= \Delta\rho + x, \\ \Delta\rho &= \sum_{i=1}^{n} w^{(i)} D\text{conv}\left(p^{(i)}\right) + b^{(i)}, \\ x &= \sum_{i=1}^{n} w^{(i)} D\text{conv}\left(p^{(i)}\right) + b^{(i)} + x. \end{aligned} \tag{3}$$

In the process of information transmission, the network structure without REAML processing also leads to the

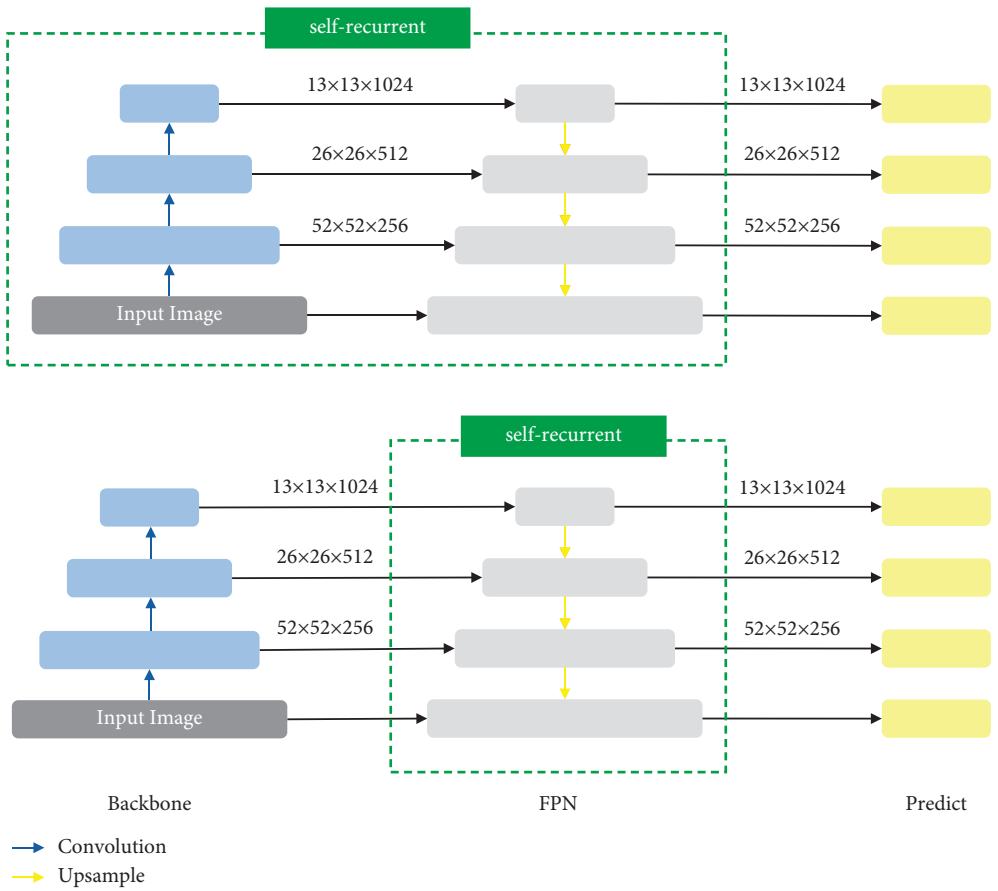

Figure 2: Above is RFP. Below is SLFF. SLFF uses the output of the backbone network as the input of FPN to obtain the multiscale feature map. The self-recurrent module incorporates feedback connections into FPN. RFP connects them back to the bottom-up backbone. SLFF lets them connecting back to the FPN. Predict module uses the prediction part of YOLOv3.

transmission of redundant information when the information is shared between multiscale feature maps. Nonsense transmission is caused by redundant information, which will not only pollute the feature map but also add a lot of extra computation. Therefore, redundant features will not only affect the representation of multiscale features but also reduce the ability of multiscale feature images to extract semantic information by direct fusion of these features, thus reducing the ability of iterative feature enhancement and affecting the efficiency of information transmission.

Our design REAML considers the feature map after iteration to minimize the interference of redundant features. Simultaneously, the structure of a residual network is used as a reference to ensure the effective object and improve the robustness of the model (see Figure 3).

In summary, the features are directly fused after iteration and the result is a feature image with superimposed redundant features, which affect the accuracy of object detection. Our model maximizes the value of feature fusion and reduces the interference of redundant information in a feature map.

We fuse the model in the process of filling the information gaps. The information gap is filled to reduce the difficulty in locating information accurately in a high-level feature map and to increase the semantic information of low-level features with higher resolution. The information required for object detection is favorable information, which is also the information to be perfected via feature fusion. REAML uses the adjusted feature map to fuse with the original feature map matching its size, learns the beneficial information in the feature map from the original feature map, and obtains a feature map that contains more explicit information. The structure of REAML is shown in the dotted box (see Figure 4).

REAML is in the middle of SLFF and is responsible for information dissemination and filtering during the transition phase of feature map inclusion. We use dilated convolution to adjust the size of a feature map and design fuse nodes. To integrate the output features at the same scales, they can be spliced with the original layer feature, to integrate feature information more effectively. We first use interpolation and dilated convolution to resize the multilevel features to an intermediate size. The difference between dilated and traditional convolution is that the convolution kernel of the dilated convolution contains voids, which can increase the reception field while the scale of the feature image is changing, thus reducing information loss. The size of a feature map of different information in dilated

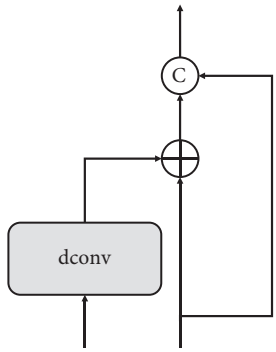

Figure 3: Residual structure in the residual enhancement architecture.

convolution is transformed. After the size changes, the feature map is fused with the original pyramid feature map with the same 2D scale, and the weight is learned for each feature point to form a weight space of a certain size. The weight space can filter the feature image to retain feature points that are beneficial for filling the blank information of the scale, maximizing the value of feature fusion, and reducing the interference of redundant information in the feature image. In our network, we also design a model like the ResNet network to retain part of the functions of the powerful ResNet network, enhance the multiscale context information, reduce the information loss of feature map at the highest level of the pyramid, and ensure the basic effect of the network.

2.3. Gap Sample-Based Features Fusion. Existing object detection models include the concept of data augmentation in feature fusion, such as RFP. Data augmentation is a means to improve the performance of deep neural networks in detecting objects. The feature map of the backbone network input will be fused with the original image for the second time, to realize the object sharing of the same image in different states, effectively disseminating information and enhancing the explicit expression of features. However, from the perspective of redundant feature increment, the superimposed feature image will generate more computation amount. Therefore, to simplify the computation cost in the process, the author adds dilated convolution to the network to expand the receptive field and enhance the robustness of the network. In common image recognition tasks, the process of data augmentation is generally regarded as one of the tasks in the preprocessing stage. Before model training, data should be flipped, scaled, and other operations to increase datasets capacity.

Based on data augmentation and basic operation, we propose the GSFF module inside a neural network. GSFF automatically enriches dataset capacity and improves the learning and generalization capability of the model by increasing the number of objects in an image. Our overall model iterations omit the backbone network. Therefore, to avoid the generation of data augmentation as an independent preprocessing method, we proposed the GSFF module (see Figure 5).

The original image is fused with the C_1 feature map obtained by the convolution calculation and feature map of batch t. Both images are at the same moment in the feature extraction of the batch and have the same size and level of abstraction. We choose a feature map with relatively comprehensive image information retention in the network and relatively accurate location information. The fused feature map is continuously input into the network.

The purpose of GSFF is to enhance the features of a simple sample across two samples with a certain distance and thus improve the generalization capability of the network model. Considerable information differences occur between the two images, and the information of the feature points in the same position is not necessarily compatible. A simple fusion operation is likely to destroy the information required for detecting an object, producing counterproductive results. Therefore, let S denote the sample to be sampled. Then, $i \in [1, \ldots, n]$, and it is the number of feature drawings of batch t; $d \in [1, \ldots, D]$ and represents the number of channels of the sample features. The output Gp with complex sample features is defined as

$$\begin{aligned} \mathrm{Gp} &= S + F^t, \\ S &= \sum_{i=1}^{n} W_i F_i^{t-1} + B = \sum_{i=1}^{n} \left(\sum_{d=1}^{D} w_d^i \sum_{d=1}^{D} f_d^i \right) + B^i, \\ \mathrm{Gp} &= \sum_{i=1}^{n} \left(\sum_{d=1}^{D} w_d^i \sum_{d=1}^{D} f_d^i \right) + B^i + F^t, \end{aligned} \tag{4}$$

where T is the number of samples, and we use F^t to represent the feature images of each other's samples at t. B represents the feature offset difference. f_i^t represents channel d of the ith feature map, and w_d^i represents the weight of channel d of the ith sample. After calculation of the preceding formulas, the result Gp is obtained. Finally, the output of GSFF is the result of spanning two different feature maps F^t and S. This feature image exerts data augmentation effects to improve the generalization capability of the model. Traditional data augmentation uses the human eyes to process the object on the original image, but a neural network is better to understand the thinking of a neural network. In our model, the object in the feature map processed by a neural network is used to enrich the object in the feature map of the same size. The neural network automatically trains the most favorable tools for GSFF and effectively improves the sensitivity of the model to dominant features. S of the first batch of feature maps is set as 0. S is zeroed to the overlap label to determine whether any overlap occurs between the labels of F and S. The feature point value of this part of F_i^t is set to 0. We learn a weight based on two feature maps to obtain the weight space generated for each feature point. By weighing the feature points in F_i^t, the effective object is added to the $t + 1$ feature map to obtain the feature map F_i^{t+1} with more information to realize GSFF.

3. Experiments

In this section, we introduce the experimental environment and evaluation criteria, compare the contributions of different modules to the original structure and the cooperation

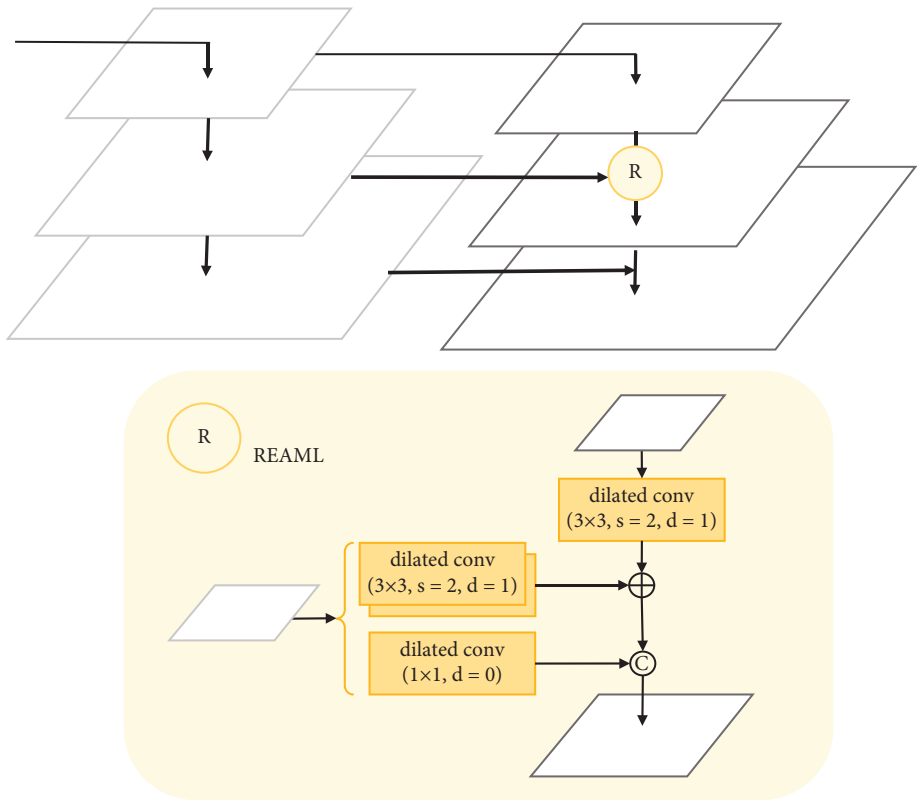

Figure 4: REAML. This figure unrolled the self-recurrent structure to a 2-step sequential network. Dilated convolution is used to transform the size of the feature image and fuse it. Note: ©, channel splicing operation; ⊕, feature fusion for automatic weight allocation.

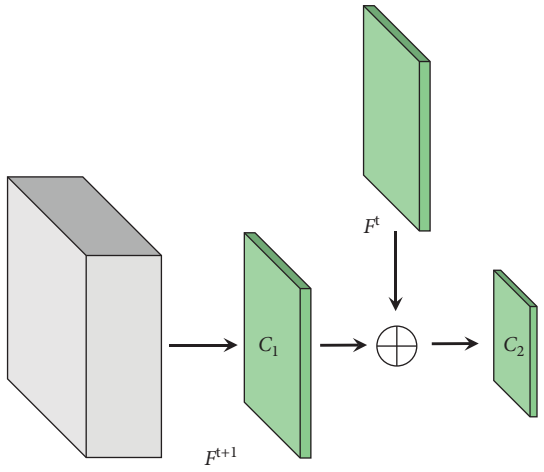

Figure 5: GSFF. The effective information in the previous feature map is extracted and fused with this feature map.

effect between modules, and present the experimental results of a series of module optimization experiments and comparison experiments.

3.1. Datasets and Experimental Environment

3.1.1. Dataset Preparation. All our experimental models are implemented based on the YOLOv3 [36] framework, using the same training platform and datasets. We strictly follow the experimental setup using COCO2017 training set training and then test the model on COCO2017 validation set and test-dev. The size of the images being input into the network model is controlled between 416 and 512 pixels, and a total of 115,000 images are included. We train each model in the COCO2017 [37] dataset with 80 classes and in the VOC2007 dataset with 20 classes of objects which are evaluated. We selected the experimental results of 150 epochs. We apply the model with good test results to practical applications, such as safety helmet detection at electric power construction site (see Figure 6).

The safety helmet dataset contains 20000 training images and 2000 test images. Our training set is collected from the Internet. 1000 images in the test set are randomly selected from the training set, and another 1000 images are provided by Yuxi Power Supply Bureau of Yunnan Power Grid Corporation. At the end of the study, the proposed pattern is compared with the actual application performance of existing modules.

3.1.2. Data Augmentation. Data augmentation is primarily used to prevent overfitting when a dataset is small. At the beginning of the training, a neural network is not sufficiently smart, and simple graphics transformation operations can fool the network. For example, an untrained neural network may think that the same object that shifts by only a few pixels is a different image. Therefore, the data augmentation method takes advantage of the features of neural networks to

Figure 6: Dataset of the safety helmet.

transform images by means of random rotation, random cropping, color dithering, and image flipping, to enrich the dataset. To train a helmet detection model with high prediction accuracy, it typically requires the support of a large number of samples. With enough parameter samples, a more efficient model can be trained even with the same structure. However, the number of training samples is limited. To overcome the problems cause by the limited number of original training samples, we adopt two data augmentation approaches to increase the quantification training samples (see Figure 7). Our first attempt is to perform traditional data augmentation operations on a dataset. Traditional data augmentation is performed by translating, rotating, and scaling the training samples. Rotation operation refers to the rotation of the image at a random angle. The dimensions of the image may not be the same after rotation. If the shape of the image is square, then rotating it at right angles will retain the image size. If the object is a rectangle, then a 180 rotation will retain the same size. In addition, we also apply translation and zooming in and out to increase the training sample. Zoom in means to zoom in on the object and place it back in the picture.

The enlarged image should be of appropriate resolution to serve the purpose of identifying the object. These approaches are designed to enhance the diversity of images, and they exert evident effects on the identification and detection of objects, such as safety helmets. To obtain more data, we must only make small changes to the existing dataset. When increasing the number of datasets, we must also focus on how to reduce massive unrelated features in the datasets.

3.1.3. Training. All experiments are conducted on a desktop with an Intel single-core i7 CPU and an Nvidia GTX-2080TI GPU (11 GB video memory). The other experimental environments are provided in Table 1. During training, we set the initial learning rate to 0.001. Then, on the 30,000th and 35,000th iterations, we divide the learning rate by 10. Batch size is set to 16. Weight loss is 0.0005, and momentum is 0.9.

3.2. Ablation Experiment. Table 2 provides the individual ablation study of self-recurrent feature fusion and GSFF based on object detection where we present their enhancement data. The ablation study in Table 2 shows how the mechanism based on looking and thinking twice achieves the best configuration within the design space we explored (see Table 2).

SLFF. We first evaluate the contribution of the SLFF module for better reference. The recurrent structure of the iterative model follows the hierarchy of the feature pyramid such that the output feature map is reentered into the network as input.

To avoid interference from other modules, we select the original YOLOv3 model for testing.

As indicated in Table 2, all the values have improved, reaching a precision of 40.62%. From the observation results, the secondary observation mechanism can more easily find the more critical occluded objects in the context information.

REAML. The REAML module is divided into REAML-1 and Model REAML-2 depending on whether it contains a residual-based design. By comparing the accuracy of the two REAML models with the original FPN model, REAML-1 reaches 54.35% accuracy and REAML-2 reaches 52.7% accuracy, an increase of 15.95% and 14.3%, respectively, over the single-scale FPN baseline. By observing the results of REAML-1 and REAML-2, the concept of ResNet can improve the stability of a network, effectively improving its accuracy.

SLFF and REAML. We also demonstrate the improved results of the fusion of the SLFF and REAML modules. We analyze whether and how to use the REAML module to achieve the best fusion effect with SLFF cooperation, as shown in the experimental results in Table 2. By adding REAML to the two moments before and after the iteration, the performance of SLFF without REAML is compared with that of SLFF + REAML. The results show that the accuracy of the SLFF module alone can reach 40.62%, while that of the model with REAML can reach 49.97%. Moreover, we conclude that model addition can have a better effect before iteration. From this comparison, the accuracy of detection can be improved by 11.57% by reducing the proportion of redundant features compared with the output after the fusion of the two feature images before the recurrent process.

GSFF. Table 2 provides the ablation results for the GSFF. Compared with the model without the GSFF module, we prove the role of GSFF in improving the generalization capability of the model, showing Box and Mask AP with YOLOv3 and FPN as the backbone. We also compare the effect of GSFF with the addition of the SLFF + REAML model. GSFF also reaches a precision of 41.96%, which is slightly higher than the effect under the original framework. The reason for this finding may be the feature screening ability of SLFF + REAML indirectly playing a role in the GSFF model. In general, GSFF can increase the number of objects in the feature map as required, increasing the sensitivity of the detector.

3.3. Fusion Module Experiment Results. Under the framework of YOLOv3, we study the performance of a fusion model. FPN, as the default model in the YOLOv3 framework, is a benchmark for comparing the effect of other fusion models. For a fair comparison with the results, we report the realized results. Our final model is based on SLFF and enhanced by REAML and GSFF. We compared our model with other fusion modules (see Table 3).

Among the existing fusion models, the original FPN model still exhibits significant advantages. The ASFF effect is the best in the single fusion model, which can achieve a precision of 48.73% and a recall rate of 81.56%. The multi-iteration fusion model BiFPN achieves the best effect, with a precision of 46.57% and a recall rate of 68.05%. Our model can achieve a precision of 61.97% and a recall rate of 77.95%. We also calculate the loss difference between the trainset and the test set to compare the fitting degree of the model.

The dataset is partitioned to prevent overfitting of the model. When the loss difference is less than zero, the model fits the original data to the maximum extent, but its performance on the test set is not ideal. When the absolute value of the loss difference is small, the moderate fitting degree of the model is indicated, to identify the model with the best effect and generalization ability. As shown in Table 3, among the existing fusion modules, the FPN model still exhibits evident advantages. Compared with state-of-the-art methods, our model still has achieved remarkable results (see Tables 1 and 4).

We further compared the rising trend of precision of the model (see Figure 8). FPN maintains a steady upward trend, and ASFF achieves the highest precision in the centralized model after 200 epochs of training. Our model achieves better results than ASFF. The overall upward trend is faster and steadier.

3.4. Actual Task Experiment Results. According to the actual task requirements, our model designs the self-recurrent learning of the fusion model. We can see the advantages of the existing fusion model by comparing it with the common dataset. On this basis, we also compare the effect of the fusion module on the safety helmet dataset. In complex actual scenarios, our model reaches 85.6% accuracy. At the same time, the accuracy of our model also obtains 80.4% and 83.9%, resulting in the scenarios of high-altitude operation and ground construction. Electrical construction scenarios involve intricate equipment supports and upper wires, but our model still achieves accuracy of 85.6% for the electric power inspection type and 83.1% for the electric power maintenance type (see Figure 9). At the same time, we experiment with the detection time of the model in different categories. The total time per image in the overall dataset is 13.2 ms. It takes 18.2 ms and 17.7 ms for high-altitude operation and ground construction, respectively. It takes 13.8 ms and 13.2 ms, respectively, for electric power inspection and electric power maintenance. It can be concluded from the above experiments that our model improves the detection accuracy of small objects and deals more flexibly with objects in a scene.

Figure 7: Results of data augmentation.

Table 1: Detection results on the COCO2017 dataset.

Model	AP	AP@0.5	AP@0.75	APsmall	APmedium	APlarge
Libra R-CNN [16]	43.0	64	47	25.3	45.6	54.6
M2Det [38]	41.0	59.7	45	22.1	46.5	53.8
Faster R-CNN [21]	40.6	58.9	44.5	22.0	42.8	52.6
Cascade R-CNN [39]	36.5	59	39.2	20.3	38.8	46.4
RefineDet512 [40]	33.0	54.5	35.5	16.3	36.3	44.3
Ours	**43.7**	**60.5**	**49.1**	**28.0**	**52.0**	**59.4**

We compared object detection results on five state-of-the-art methods to demonstrate the effectiveness of our method.

Table 2: Effect of each component.

SLFF	REAML	GSFF	Recall	Precision	map@0.5	F_1
✓			60.32	40.62	55.66	43.20
	✓		73.07	52.70	72.55	59.02
		✓	60.82	41.96	58.92	44.82
✓	✓		71.17	49.97	70.10	56.52
✓	✓	✓	77.95	61.95	79.07	65.00

Results are reported on COCO2017. SLFF: self-recurrent learning-based feature fusion, REAML: residual enhancement architecture-based multichannel, and GSFF: and gap sample-based feature fusion.

Table 3: Fusion results on the COCO2017 dataset.

Model	Schedule	Image size	Recall	Precision	map@0.5	F_1	GIoU	Obj	cls
FPN [14]	×1	416	50.82	38.40	53.40	41.58	0.045	0.793	0.532
FPN-3	×3	416	48.76	32.58	46.58	37.08	0.115	2.232	0.240
AsFF [18]	×1	416	81.56	48.73	75.76	59.00	0.041	1.349	0.398
Passp [41]	×1	416	62.31	38.13	59.13	44.73	0.138	1.933	0.305
Pacsp [41]	×1	416	56.13	40.58	56.38	44.67	0.065	1.727	0.308
NAS-FPN [19]	×1	416	34.03	16.86	28.10	19.47	0.498	4.750	0.110
AugFPN [42]	×1	416	72.37	35.58	67.60	45.57	0.025	2.14	0.300
BiFPN-1 [20]	×1	416	26.01	19.93	28.43	20.9	0.567	8.360	0.315
BiFPN-3	×3	416	68.05	46.57	70.73	53.45	0.202	1.357	0.464
Ours	**×1**	**416**	**77.95**	**61.97**	**79.07**	**65.00**	**0.053**	**1.20**	**0.331**

We compared object detection results on current popular fusion modules to demonstrate the effectiveness of our model. Ours: SLFF + REAML + GSFF.

Table 4: Mean average precision experimental results on the VOC2017 dataset.

Model	Backbone	Input size	map@0.5
Faster R-CNN [21]	VGG-16	1000 × 600	73.20
SSD513 [43]	ResNet-101	513 × 513	80.60
HyperNet [9]	VGG-16	1000 × 600	76.30
YOLO [44]	GoogleNet	448 × 448	63.40
Ours	**Darknet-53**	**416 × 416**	**88.70**

We compared object detection results on four state-of-the-art methods to demonstrate the effectiveness of our method.

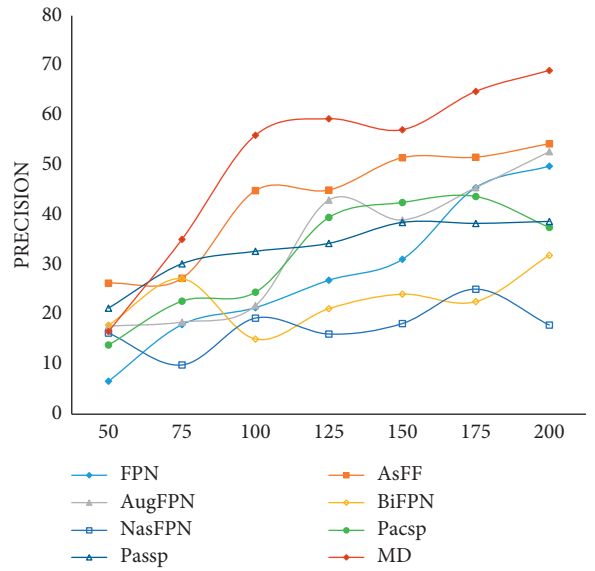

Figure 8: Increasing trend of the precision of each model.

(a)

(b)

(c)

(d)

Figure 9: Test results of monitoring power construction site. (a) High-altitude operation. (b) Ground construction. (c) Electric power inspection. (d) Electric power maintenance.

Figure 10: Examples of problems.

4. Conclusion

In this paper, we have presented a recursive FPN framework based on RFP designed with the design concept of looking and thinking twice, while adding the basic operations of data augmentation to the backbone network. We performed object detection on COCO2017's dataset and performed a separate comparative evaluation of the fusion capability of each module. Finally, we carried out the safety helmet detection, in the actual application to verify it. Our model reached 85.6% accuracy. However, in the case of faulty camera equipment or extremely dark or bright background, the detection result was not ideal (see Figure 10).

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 there are no conflicts of interest regarding the publication of this paper.

Authors' Contributions

Lvjiyuan Jiang and Haifeng Wang contributed equally to this work.

Acknowledgments

The authors acknowledge the dataset for safety belt detection provided by Yuxi Power Supply Bureau, Yunnan Power Grid Corporation, and Yunnan Province Thousand Talents Program; Postgraduate Research Innovation Fund Project of Yunnan Normal University (ysdyjs2020148); and National Students' Platform for Innovation and Entrepreneurship Training Program (202010681092).

References

[1] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[2] M. Feindt and U. Kerzel, "The NeuroBayes neural network package," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 559, no. 1, pp. 190–194, 2006.

[3] P. Yi, Z. Wang, K. Jiang, J. Jiang, T. Lu, and J. Ma, "A progressive fusion generative adversarial network for realistic and consistent video super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2021.

[4] J. Yang, J. Y. Yang, D. Zhang, and J. F. Lu, "Feature fusion: parallel strategy vs. serial strategy," *Pattern Recognition*, vol. 36, no. 6, pp. 1369–1381, 2013.

[5] Q. S. Sun, S. G. Zeng, Y. Liu, P. A. Heng, and D. S. Xia, "A new method of feature fusion and its application in image recognition," *Pattern Recognition*, vol. 38, no. 12, pp. 2437–2448, 2015.

[6] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: enhancing feature fusion for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 269–284, Munich, Germany, 2018.

[7] H. Gunes and M. Piccardi, "Affect recognition from face and body: early fusion vs. late fusion," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3437–3443, 2005.

[8] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick, "Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2874–2883, Las Vegas, NV, USA, 2016.

[9] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: towards accurate region proposal generation and joint object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 845–853, Salt Lake City, UT, USA, 2018.

[10] C. Ning, H. Zhou, Y. Song, and J. Tang, "Inception single shot multibox detector for object detection," in *Proceedings of the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 549–554, IEEE, Hong Kong, China, 2017.

[11] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proceedings of the European Conference on Computer Vision*, pp. 354–370, Springer, Amsterdam, The Netherlands, 2016.

[12] K. Ovtcharov, O. Ruwase, J. Y. Kim, J. Fowers, K. Strauss, and E. S. Chung, "Accelerating deep convolutional neural networks using specialized hardware," *Microsoft Research Whitepaper*, vol. 2, no. 11, pp. 1–4, 2018.

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

[14] N. Zheng, G. Loizou, X. Jiang, X. Lan, and X. Li, "Computer vision and pattern recognition," *International Journal of Computer Mathematics*, vol. 84, no. 9, pp. 1265-1266, 2007.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, Salt Lake City, UT, USA, 2018.

[16] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: towards balanced learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 821–830, Long Beach, CA, USA, 2019.

[17] T. Dai, L. Cao, Q. He, H. Wu, and W. Shen, "A two-way neutronics/thermal-hydraulics coupling analysis method for fusion blankets and its application to CFETR," *Energies*, vol. 13, no. 16, pp. 1–20, 2020.

[18] S. Liu, D. Huang, and Y. Wang, "Learning spatial fusion for single-shot object detection," 2019, https://arxiv.org/abs/1911.09516.

[19] G. Ghiasi, T. Y. Lin, and Q. V. Le, "Nas-FPN: learning scalable feature pyramid architecture for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7036–7045, Long Beach, CA, USA, 2019.

[20] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790, Seattle, WA, USA, 2020.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.

[22] C. Cao, X. Liu, Y. Yang et al., "Look and think twice: capturing top-down visual attention with feedback convolutional neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2956–2964, Santiago, Chile, 2015.

[23] J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1637–1645, Las Vegas, NV, USA, 2016.

[24] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3367–3375, Boston, MA, USA, 2015.

[25] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3147–3155, Honolulu, HI, USA, 2017.

[26] S. Qiao, L. C. Chen, and A. Yuille, "Detectors: detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10213–10224, Las Vegas, NV, USA, 2021.

[27] H. Salehinejad, S. Valaee, T. Dowdell, and J. Barfett, "Image augmentation using radial transform for training deep neural networks," in *Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3016–3020, IEEE, Calgary, Canada, 2018.

[28] S. Borzooei, Y. Amerlinck, S. Abolfathi et al., "Data scarcity in modelling and simulation of a large-scale WWTP: stop sign or a challenge," *Journal of Water Process Engineering*, vol. 28, pp. 10–20, 2019.

[29] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 13001–13008, 2020.

[30] G. G. Chowdhury, "Natural language processing," *Annual Review of Information Science and Technology*, vol. 37, no. 1, pp. 51–89, 2003.

[31] K. Chaitanya, N. Karani, C. F. Baumgartner, A. Becker, O. Donati, and E. Konukoglu, "Semi-supervised and task-driven data augmentation," *Lecture Notes in Computer Science*, pp. 29–41, Springer, Cham, Switzerland, 2019.

[32] O. Harel, E. M. Mitchell, N. J. Perkins et al., "Multiple imputation for incomplete data in epidemiologic studies," *American Journal of Epidemiology*, vol. 187, no. 3, pp. 576–584, 2018.

[33] J. Zhao, K. Liu, and G. Wang, "Adding redundant features for CRFs-based sentence sentiment classification," in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 117–126, Waikiki, Hawaii, 2008.

[34] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702-703, Seattle, WA, USA, 2020.

[35] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2018–2025, IEEE, Barcelona, Spain, 2011.

[36] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[37] T.-Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *Proceedings of the European Conference on Computer Vision*, pp. 740–755, Springer, Zurich, Switzerland, 2014.

[38] Q. Zhao, T. Sheng, Y. Wang et al., "M2Det: a single-shot object detector based on multi-level feature pyramid network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 9259–9266, 2019.

[39] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, 2018.

[40] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4203–4212, Salt Lake City, UT, USA, 2018.

[41] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/abs/2004.10934.

[42] C. Guo, B. Fan, Q. Zhang, S. Xiang, and C. Pan, "AUGFPN: improving multi-scale feature learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12595–12604, Long Beach, CA, USA, 2020.

[43] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, pp. 21–37, Springer, Amsterdam, The Netherlands, 2016.

[44] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788, Las Vegas, NV, USA, 2016.