Multi-Scale Residual Aggregation Feature Pyramid Network for Object Detection

Hongyang Wang 1 and Tiejun Wang 2,*

1 Key Laboratory of China’s Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou 730000, China
2 School of Mathematics and Computer Science, Northwest Minzu University, Lanzhou 730000, China
* Correspondence: wtj@mail.lzjtu.cn

Abstract: The effective use of multi-scale features remains an open problem for object detection tasks. Recently, proposed object detectors have usually used Feature Pyramid Networks (FPN) to fuse multi-scale features. Since Feature Pyramid Networks use a relatively simple feature map fusion approach, it can lead to the loss or misalignment of semantic information in the fusion process. Several works have demonstrated that using a bottom-up structure in a Feature Pyramid Network can shorten the information path between lower layers and the topmost feature, allowing an adequate exchange of semantic information from different layers. We further enhance the bottom-up path by proposing a multi-scale residual aggregation Feature Pyramid Network (MSRA-FPN), which uses a unidirectional cross-layer residual module to aggregate features from multiple layers bottom-up in a triangular structure to the topmost layer. In addition, we introduce a Residual Squeeze and Excitation Module to mitigate the aliasing effects that occur when features from different layers are aggregated. MSRA-FPN enhances the semantic information of the high-level feature maps, mitigates the information decay during feature fusion, and enhances the detection capability of the model for large objects. It is experimentally demonstrated that our proposed MSRA-FPN improves the performance of the three baseline models by 0.5–1.9% on the PASCAL VOC dataset and is also quite competitive with other state-of-the-art FPN methods. On the MS COCO dataset, our proposed method can also improve the performance of the baseline model by 0.8% and the baseline model’s performance for large object detection by 1.8%. To further validate the effectiveness of MSRA-FPN for large object detection, we constructed the Thangka Figure Dataset and conducted comparative experiments. It is experimentally demonstrated that our proposed method improves the performance of the baseline model by 2.9–4.7% on this dataset and can reach up to 71.2%.

Keywords: object detection; feature pyramid network; Thangka image

1. Introduction

Object detection is one of the most fundamental and challenging problems in computer vision, aiming to predict a bounding box with category labels for objects of interest in images. It has a wide range of applications in many fields, such as autonomous driving [1], face detection [2], remote sensing [3], and medical care [4]. In recent years, significant progress has been made in applying neural networks to image processing, and some researchers have also tried to introduce convolutional neural networks into object detection tasks. The input image of the object detector has to go through several convolutional layers of the backbone to extract features first, in which the scale of the input image is shrinking, the number of channels is increasing, and the semantic information represented becomes more and more abstract. In other words, the feature maps obtained from the shallow layer of the backbone contain more spatial information, which is beneficial for detecting small objects and predicting object locations. In contrast, the deep layer of the backbone contains more abstract semantic information, which is beneficial for detecting...
large objects and predicting classification, thus causing the problem of semantic imbalance between different feature maps. Some early object detectors use feature maps of different scales obtained from different stages of the backbone for direct prediction to enhance the model’s detection capability. However, this approach does not solve the problem of semantic imbalance between feature maps well. Therefore, the authors of [5] proposed a Feature Pyramid Network (FPN), as shown in Figure 1. As can be seen from the figure, the FPN passes the high-level features of the backbone network to the low-level feature maps through a top-down structure, which complements the low-level semantics, so that a high-resolution, strongly semantic bottom-level feature map can be obtained, which is beneficial for the detection of small objects. The structure achieves the exchange of semantic information between feature maps at different levels, alleviating the imbalance problem between feature maps.

![Figure 1. Feature Pyramid Network.](image)

However, the FPN still has some problems. (1) Information decay. When FPN aggregates information from feature maps at different scales, multiple upsampling leads to information decay or misalignment, thus making the semantic information exchange between different feature maps inadequate.

Some methods counteract information decay by enhancing features or introducing new upsampling modules. AugFPN [6] reduces the information loss in the top feature by residual feature enhancement. At the same time, FaFPN [7] enhances upsampling by introducing deformable convolution, thereby reducing the misalignment generated by feature map fusion. CARAFE [8] proposed a general and lightweight feature upsampling method to mitigate information loss during upsampling. There are also methods to mitigate information decay by changing the structure of the FPN. In contrast to the single top-down fusion structure of FPN in Figure 2a, the PANet [9] shown in Figure 2b adds a bottom-up structure to FPN. This structure enhances the entire feature hierarchy using accurate low-level spatial information, thus shortening the information path between the low-level and high-level features. The BiFPN [10] in Figure 2c adds residual connections to the PANet, removes the nodes of single input edges, and finally superimposes multiple BiFPN Layers to enhance the perception of multi-scale features of the detector. FPANet [11] augmented the BiFPN with the proposed SeBiFPN. Second, the lateral connection of the FPN directly compresses the feature map channels from the backbone, which also causes a large amount of information fading. CE-FPN [12] mitigates the information decay of lateral connections by introducing sub-pixel convolution, and reference [13] applies deformable convolution to lateral connections with FPN to enhance the semantics. (2) Aliasing effects during fusion. Feature maps obtained at different stages of the backbone have completely different semantic information. A simple additive fusion of these feature maps will result in aliasing effects. Methods such as CE-FPN and FaFPN redirect the fused feature maps for channel guidance to mitigate aliasing effects.
To address these issues, we have made three improvements to the FPN. (1) Unidirectional Cross-Level Residual Aggregation Module (UCLRAM). Models such as PANet and BiFPN have achieved good results in combating information fading using the underlying feature maps. However, we believe it is still possible to improve the scale awareness of FPN by enhancing this bottom-up structure. To this end, we improved the tree-structured aggregation network of DLA [14] (as in Figure 3), which aggregates the features extracted from two adjacent feature extraction blocks in the backbone of a tree structure that can better fuse semantic and spatial information. We only keep the lateral and bottom-up connections of the tree-structured aggregation network, add the bottom-up aggregation connections of the same stage, and finally adjust the overall structure to be better for feature fusion. We name the adjusted aggregation structure UCLRAM. UCLRAM aggregates the semantic information contained in the feature maps from different scales in a triangular structure from the bottom up to the top layer, countering the information decay during upsampling (top-down) by enhancing the semantic information at the topmost layer. (2) Introduction of residual structure. We introduce the information in the high-level of FPN to the bottom layer through the residual structure in upsampling, which also mitigates the information decay to a different extent. (3) Residual Squeeze and Excitation Module (RSEM). In order to mitigate aliasing effects, we designed a Residual Squeeze and Excitation Module and placed it at the three aggregation nodes with the highest degree of fusion to achieve the guidance of global semantic information by weighting the channel information of the feature map.

Regarding the overall fusion structure, there is some similarity between MSRA-FPN and the previously proposed PANet, as well as BiFPN, which both use a bottom-up structure to enhance the FPN. However, the differences in the structure of MSRA-FPN bring unique
advantages. First, we front-loaded the bottom-up aggregation structure UCLRAM so that it can directly access the raw features from the backbone network, which can further enhance UCLRAM. Second, the triangular structure of UCLRAM has a more robust aggregation capability than the single-layer structure of PANet and BiFPN, enabling the high-level feature to have richer semantic information to enhance the detector’s ability to detect large objects.

Experimentally, it is demonstrated that on the PASCAL VOC [15] dataset, our proposed method improves 0.5–1.9% compared to the baseline model and has an advantage over other methods. Furthermore, on the MS COCO [16] dataset, our proposed method can also improve the AP of the baseline model by 0.8% and the $AP_l$ by 1.8%. On the Thangka Figure Dataset (TKFD), our proposed model improves the performance of the three baseline models by 2.9–4.7%, validating the effectiveness of the MSRA-FPN for Thangka figure detection as well as large object detection.

2. Related Work

2.1. CNN-Based Object Detection

As one of the most fundamental and challenging tasks in computer vision, object detection tasks have received a lot of attention from researchers. Especially after the rise of convolutional neural networks, a large number of excellent object detectors have emerged. Object detection consists of two tasks—classification and regression—where the classification task predicts the class of an object, while the regression task is responsible for predicting the bounding box of that object. Depending on the architecture of the network, the existing detector can be classified into anchor-based and anchor-free detectors.

Anchor-based detectors can be further divided into two-stage detectors and single-stage detectors. Among the representative models of two-stage detectors are Faster R-CNN [17], RFCN [18], and Sparse R-CNN [19], etc. and the representative models of single-stage detectors are RetinaNet [20], SSD [21], and YOLO [22]. The two-stage detectors must first use a Region Proposal Network (RPN) to generate proposals, distinguish between foreground and background, and complete the final detection on top of the proposals. The single-stage detectors locate and recognize objects directly on the feature map, which is more practical because of its uniform model structure and efficient inference speed. The two-stage design ensures the model’s accuracy, but its inference speed is much less than that of the single-stage detectors. All these anchor-base detectors first generate anchors of different sizes on each pixel point of the feature map and then predict the classes corresponding to each anchor and the relative coordinates based on the anchor.

As the anchor itself is a priori data, it has to be redesigned manually for different datasets. In addition, the anchor-based detectors generate many anchors each time, which increases the computational cost of the model and causes a severe imbalance between positive and negative samples, so some researchers have tried to discard anchors and designed anchor-free detectors. Anchor-free detectors can be broadly classified into two types. One approach is based on keypoints, such as Reppoints [23] and CornerNet [24], and CenterNet [25]. Another type of method is dense prediction, such as YOLO, DenseBox [26], FCOS [27], and FoveaBox [28].

In addition, some researchers have introduced Transformers [29] into the object detection task with good results, such as DETR [30], ViT [31], and Swin Transformer [32].

2.2. Multi-Scale Feature Fusion in Object Detection

Early object detectors based on the neural network did not perform multi-scale feature fusion. The Feature Pyramid Network [5] (FPN) constructs a top-down path that allows the high-level feature to be fused downward, enabling the exchange of information between feature maps of adjacent levels to mitigate semantic imbalance between different layers. The deeper features of the backbone contain richer semantic information, while the shallow features contain more spatial information. FPN ignores the importance of the low-level features, while PANet [9] designs an additional bottom-up structure on top of FPN to
further exploit the low-level spatial information. BiFPN [10] uses a weighted bi-directional fusion structure to fuse low-level features and high-level abstract semantic features, which effectively improves detection performance.

Meanwhile, since the upsampling process of FPN uses relatively primitive interpolation methods, which causes misalignment and loss of semantic information to a certain extent, some researchers have tried to improve the upsampling method. For example, FaFPN [7] introduced deformable convolution [33] during feature fusion to mitigate the offset generated during upsampling. CE-FPN [12] performs feature enhancement through the channel guidance module and introduces sub-pixel convolution to mitigate the information decay during channel compression.

In addition, some studies argue that there are specific semantic differences between feature maps at different scales, and directly fusing them will reduce the representational power of the features. AugFPN [6] proposed a series of methods to reduce this semantic discrepancy. PSPNet [34] proposes pyramid pooling to extract global contextual information for feature enhancement. Other researchers have also focused on the semantic balance of the model, such as Libra R-CNN [35].

Integrating the research on CNN object detection and multi-scale feature fusion with the practical problem of object detection in Thangka images, we design and propose a method MSRA-FPN using a bottom-up structure for FPN enhancement.

3. Method

3.1. OverAll

Adding a Feature Pyramid Network [5] to the object detector is intended to transfer information between multi-level feature maps better. FPN conveys more abstract semantic information from the top to the bottom layer, alleviating the semantic imbalance between different feature maps.

This type of information transfer relies heavily on a top-down fusion structure. This structure requires multiple levels of input \( \mathbf{P}^{\text{in}} = (P_1^{\text{in}}, \ldots, P_4^{\text{in}}) \), where \( P_i^{\text{in}} \) denotes features with a resolution of the \( 1/2^i \) level of the input image, and the scale of the feature map increases to twice the original size with each upsampling, as shown in Equation (1).

\[
\begin{align*}
P_4^{\text{out}} &= \text{Conv}(P_4^{\text{in}}) \\
P_3^{\text{out}} &= \text{Conv}(P_3^{\text{in}} + \text{Resize}(P_4^{\text{out}})) \\
P_2^{\text{out}} &= \text{Conv}(P_2^{\text{in}} + \text{Resize}(P_3^{\text{out}})) \\
P_1^{\text{out}} &= \text{Conv}(P_1^{\text{in}} + \text{Resize}(P_2^{\text{out}}))
\end{align*}
\]

(1)

To counteract the semantic loss in upsampling and reduce the confounding phenomenon during feature map fusion, we designed MSRA-FPN based on the structure of FPN. MSRA-FPN consists of a bottom-up unidirectional cross-layer residual aggregation module (a), a top-down fusion module with residual connection (b), and a bottom-up fusion module (c), as shown in Figure 4. The process of feature fusion is as follows.

1. The MSRA-FPN inputs a four-layer feature map from the backbone, labeling it as \( \{C_1, C_2, C_3, C_4\} \). Its channel number is \( \{256, 512, 1024, 2048\} \). To allow feature maps of the same scale to be summed, it is necessary to ensure that their channel numbers are of the same size, so it is necessary to input \( \{C_1, C_2, C_3, C_4\} \) to a set of \( 1 \times 1 \) convolutions first so that the channel numbers are normalized to 256. The normalized \( \{C_1, C_2, C_3, C_4\} \) is input to the UCLRAM for feature fusion enhancement and output as three new feature maps \( \{A_2, A_3, A_4\} \), whose number of channels remains 256. We also designed the Residual Squeeze and Excitation Module to mitigate the aliasing effects caused by the fusion of multiple feature maps at different scales.
Figure 4. Overall structure of MSRA-FPN, where (a) is a three-input node as an example of computation, (b) is a two-input node as an example of computation, and (c) is a down-sampling fusion as an example of computation.

(2) Input \{A_2, A_3, A_4\} into the structure of (b) for top-down feature fusion to obtain the feature map \{F_2, F_3, F_4\}. The up-sampling in the fusion process uses the nearest neighbor interpolation method with a sampling rate of two. After the summation of the upsampled feature map and the feature map obtained from the lateral connection, one branch continues to aggregate downward, and the other branch needs to be summed with the feature map of the first stage of UCLRAM.

(3) Three feature maps \{F_2, F_3, F_4\} for prediction are obtained after inputting \{P_2, P_3, P_4\} into the structure of (c) with 256 channels. For better prediction at multiple scales, \{P_5, P_6\} are obtained after downsampling using Maxpooling in turn, starting from \(P_4\), and finally \{P_2, P_3, P_4, P_5, P_6\} is input into the detection head module for category and bounding box prediction.

3.2. Unidirectional Cross-Level Residual Aggregation Module

To fully fuse the feature information from multiple layers into the top-level feature map, we designed the unidirectional cross-layer residual aggregation module UCLRAM, as shown in Figure 4a. This module consists of four stages, and each stage aggregates the feature maps of the previous stage to form a new feature map, which we call an aggregation node. The number of aggregation nodes in each stage is reduced by one from front to back, forming a unidirectional triangular aggregation structure. These aggregation nodes are divided into two-input (e.g., \(A_2^2\)) and three-input (e.g., \(A_3^3\)) aggregation nodes according to the number of input feature maps.

Figure 5a shows an example of the detailed structure of the two-input aggregation node \(A_2^2\). We abstract the computation of the two-input aggregation node, as shown in Equation (2).

\[
A_i^j = \text{Conv}_{1 \times 1}(A_{i-1}^j) + \text{Conv}_{3 \times 3}(A_{i-1}^{j-1})
\]  

where \(i\) denotes the \(i\)-th fusion stage, \(j\) denotes the \(j\)-th aggregation node of the current stage. The feature map \(A_{i-1}^j\) from the same layer of the previous stage is through a \(1 \times 1\) convolution.
convolutional layer to obtain a new feature map $\hat{A}_{i-1}^j$ with size $[256, W, H]$ ($W$ denotes the width of the feature map, and $H$ denotes the height of the feature map). The feature map $A_{i-1}^j$ from the lower layer of the previous stage is of size $[256, 2W, 2H]$. To make them the same size, the fused feature map is then obtained by using cross-layer residual connection ($3 \times 3$ convolutional layer) to downsample $A_{i-1}^j$ into a feature map $\hat{A}_{i-1}^j$ of size $W \times H$. Afterward, $\hat{A}_{i-1}^j$ and $A_{i-1}^j$ are added together to obtain the fused feature map.

Figure 5. Detailed structure of the aggregation node. (a) Two-input aggregation node $A_2^j$; (b) Three-input aggregation node $A_3^j$.

Figure 5b shows an example of the detailed structure of the computational three-input node $A_3^j$. We define the three-input aggregation node as Equation (3).

$$
N = \text{Conv}_{1 \times 1}(A_{i-1}^j) + \text{Conv}_{3 \times 3}(A_{i-1}^j)
$$

$$
A_i^j = F(N + \text{Conv}_{3 \times 3}(A_{i-1}^j) + A_{i-1}^j)
$$

$$
A_{i+1}^j = \text{Conv}_{3 \times 3}(N)
$$

When computing the feature map $A_i^j$ of the three-input convergence node, it is necessary to first pass the $j$-th feature map $B$ from stage $A_{i-1}^j$ through a $1 \times 1$ convolution layer to obtain a new feature map $\hat{A}_{i-1}^j$ (whose size is $[256, W, H]$).

The feature map $A_{i-1}^j$, one layer lower than the current aggregation node, is of size $[256, 2W, 2H]$. In order to fuse $\hat{A}_{i-1}^j$ and $A_{i-1}^j$, we need to use cross-layer residual connectivity ($3 \times 3$ convolutional layer) to normalize the feature map $\hat{A}_{i-1}^j$ to the size of $[256, W, H]$ and then add $\hat{A}_{i-1}^j$ and $A_{i-1}^j$ to obtain the feature map $N$. Afterward, $A_{i-1}^j$ is downsampled to obtain the feature map $A_{i-1}^j$, and $N$ and $A_{i-1}^j$ are summed and then guided by the RSEM module to obtain the aggregation node $A_i^j$.

It should be noted that in the calculation of $A_3^j$, the feature map $N$ will enter a double branch to continue the aggregation, one of which is consistent with the above and obtains $A_3^j$ after aggregation, while the other will continue to downsample and complete the aggregation after fusion with the topmost feature map, as shown in the dashed part in Figure 5b, which we define as Equation (4).

3.3. Residual Squeeze and Excitation Module

Different layers in the Feature Pyramid Network have different scales and channels, representing different semantic information. The feature at the deeper layer of the backbone is more abstract and richer in semantic information due to multiple convolution operations. In contrast, the feature map at the bottom layer is less abstract and contains more spatial information. The low-level feature has more spatial information, which helps object localization tasks and small object detection, as well as provides better access to local features of
the image. The high-level feature maps are relatively small in scale. After multiple convolutional kernel downsampling of the backbone, most of its spatial information is transformed into abstract semantic information, which is more conducive to object classification and large object detection. At the same time, after the feature map reduces, the receptive field increases due to the rise in the relative area covered by one sliding of the convolutional kernel to focus more on the global information of the image.

Based on the above conclusions, due to the aggregation of a large number of feature maps at different levels in MSRA-FPN, aliasing effects are caused to some extent. To alleviate this problem, we introduced SENet [36] and used a skip connection to add the input feature maps to a weighted feature map. The aliasing effects are mitigated by a guide of global-information-like channels, the structure of which is shown in Figure 6. To balance the performance and parameters of the model, we did not use the module on all aggregation nodes but used RSEM on the three input aggregation nodes of UCLRAM.

![Residual squeeze and excitation module](image)

**Figure 6.** Residual squeeze and excitation module.

In RSEM, the feature map obtained in the previous stage is first changed into a $1 \times 1 \times C$ ($C$ is the number of channels) feature map $z$ by an Avgpooling, and this step extracts the global information of the input feature map. Then the weighted vector $u$ of $z$ is obtained using the weighting function $f()$. Then $u$ is multiplied by the input feature map to obtain the guided feature map $C$. Finally, add $C$ and the input feature map using the residual connection. The operation is essentially a channel attention mechanism. The process is abstracted, as shown in Equations (5) and (6).

$$
\begin{align*}
  z &= \text{Avgpooling}(m) \\
  u &= f(z) \\
  C &= u \ast m \\
  \hat{C} &= C + m \\
  f(z) &= \text{Sigmoid}(\text{Conv}_{1 \times 1}(\text{Rule}(\text{Conv}_{1 \times 1}(z))))
\end{align*}
$$

4. Experiments

We validated the MSRA-FPN on the publicly available dataset PASCAL VOC [15]. We performed comparison experiments with other state-of-the-art FPN [5] methods and ablation experiments on this dataset for different components of our proposed method. We also conducted comparative experiments on MS COCO. In addition, comparative experiments were carried out on the Thangka figure dataset to verify the effectiveness of MSRA-FPN for detecting large objects in images.

4.1. Implementation Details

We set the training batch size to 8, used a 3080ti for training, set the initial learning rate to 0.001, the momentum to 0.9, the weight decay to 0.0001, and used the first 500 iterations as a warm-up with a warm-up rate of 0.001. A simple random flip was used for data enhancement, with a flip rate of 0.5. It should be noted that we used the backbone parameters pre-trained on ImageNet for the initialization of the model and froze the backbone parameters.
parameters in the first phase. Secondly, PASCAL VOC2007 and TKFD used the same training strategy for a total of 100 epochs, with a learning rate decay at epochs 60 and 90, with a decay rate of 0.1. Note that we trained a total of 36 epochs on MS COCO, decayed the learning rate by a factor of 0.1 at 30 epochs, and set the initial learning rate to 0.006 and scaled the input size to 320.

We used the Pytorch framework and the open source object detection framework MMDetection2 [37] as experimental platforms and used three classical object detectors with FPN, RetinaNet [20], FoveaBox [28], and Reppoints [23], as baseline models. All hyperparameters of the models in the experiments were kept consistent except for the Feature Pyramid Network part and used the same training strategy and experimental setting.

4.2. Dataset
4.2.1. PASCAL VOC

The PASCAL VOC dataset provides a standardized dataset including 20 object classes for object detection tasks. The comparative experiments section used two datasets to train the model, one using the PASCAL VOC2007 trainval set (5011 images and 12,608 objects in total). The other set was trained using the PASCAL VOC2007 and PASCAL VOC2012 trainval set (a total of 16,551 images and 40,058 objects) together. Both sets of experiments were tested on the test set of VOC2007.

4.2.2. MS COCO

MS COCO has 80 object classes containing 115k images for training (train2017) and 5k images for validation (val2017) and 20k (test-dev) images for testing. We trained the model on train2017 and report the results on val2017 for AP, AP50, AP75 and on small, medium, and large objects (APs, APm, APl).

4.2.3. Thangka Figure Dataset

The Thangka is a unique Chinese painting art and an intangible piece of world cultural heritage. Figure 7 shows various types of Thangka images. Our research group has been researching issues related to the digital preservation of Thangka for a long time, among which the object detection of Thangka images is a crucial research problem.

Most Thangkas are depictions of figures in Buddhism, which tend to take up a considerable proportion of the entire canvas, with a high degree of similarity between some of the figures, as in Figure 7a,b, and a large number of figures in some of the Thangka images, as in Figure 7c. In addition, the Thangka images themselves are characterized by bright colors, rich content and complex backgrounds, making them difficult to detect.

Figure 7. Images of various types of Thangka. (a) Amitabha; (b) Sakyamuni; (c) Five Dhyani Buddha; (d) Green Tara.

We have studied the detection of figures in Thangkas, produced a Thangka figure dataset (TKFD) with 26 types of typical figures as detection objects, and validated our proposed method, MSRA-FPN, on this data.
TKFD uses a total of 1693 Thangka images with 4327 objects, of which there are 1340 in the training set, 156 in the validation set, and 197 in the test set, involving a total of 26 detection categories, whose classes distribution is shown in Figure 8. We classify objects in the dataset with an area less than $32 \times 32$ as small objects, those with an area greater than $32 \times 32$ and less than $96 \times 96$ as medium objects, those with an area greater than $96 \times 96$ and less than $512 \times 512$ as large objects, and those with area more than $512 \times 512$ as huge objects. As can be seen in Figure 9, more than half of the objects in the Thangka figure dataset belong to large objects.

**Figure 8.** Distribution of the number of object categories in TKFD.

**Figure 9.** Scale distribution of objects in TKFD.
4.3. Main Results

4.3.1. Qualitative Evaluation

We compared the detection results of MSRA-FPN and other state-of-the-art FPN-based methods on PASCAL VOC2007 and TKFD, respectively. ResNet50 [38] is uniformly used here as the backbone as well as the same model hyperparameters and experimental settings to ensure fairness.

**PASCAL VOC:** In Figure 10, MSRA-FPN correctly predicts ‘bicycle’ in the figure, while both FPN and PANet [9] detect it as ‘motorbike’; from the second row, it can be seen that MSRA-FPN detected the large object ‘airplane’, while FPN identified it as ‘bus’ and PANet did not identify any object; from the third row, it can be seen that MSRA-FPN detected the small object further away ‘sheep’, which was not detected by FPN or PANet. This demonstrates that our proposed MSRA-FPN has better detection results on large objects and can effectively fuse information on more scales, improving the detection capability and accuracy of the model.

**TKFD:** The first row of Figure 11 shows a Thangka with a predominantly green color. Due to the background and object’s extremely similar color, FPN and PANet did not detect the ‘green-tara’ in the upper left corner, which MSRA-FPN correctly detected. The second row shows that FPN did not detect ‘manjusri’ and ‘vajradhara’ at the bottom of the Thangka image and missed the ‘bhaishajyaguru’ at the top right of the image, and PANet missed the ‘vajradhara’ at the top left of the image. MSRA-FPN detects the objects not detected by FPN and PANet. It can be seen from the third row that FPN did not detect ‘bhaishajyaguru’ and ‘aksobhya’ at the top left of the image, while PANet missed the ‘green-tara’, and MSRA-FPN detected these three missed figures.

![Detection results of FPN, PANet and MSRA-FPN on PASCAL 2007.](image)

**Figure 10.** Detection results of FPN, PANet and MSRA-FPN on PASCAL 2007. (a) Detection results for FPN. (b) Detection results for PANet. (c) Detection results for MSRA-FPN.
4.3.2. Quantitative Evaluation

**PASCAL VOC:** We compared the detection performance of the proposed MSRA-FPN with the baseline model and the existing state-of-the-art FPN-based methods. The FPN part of RetinaNet [20] was replaced with other state-of-the-art FPN-based methods, and comparison experiments were conducted with the same backbone, hyperparameters, and other experimental settings. Table 1 reports the results with the PASCAL VOC2007 trainval set alone, and Table 2 reports the results with both the trainval set of PASCAL VOC2007 and PASCAL VOC2012 together. Our proposed method improves the performance of RetinaNet.
by 1.6% when trained with PASCAL VOC2007 alone compared to the baseline model and by nearly 1% on VOC2007+VOC2012, and it is competitive with other state-of-the-art FPN-based methods.

To better validate the generality of MSRA-FPN, we also conducted comparative experiments using two anchor-free object detectors as baseline models, as shown in Tables 1 and 2 for the Reppoints and FoveaBox experimental results. As can be seen, MSRA-FPN can improve the performance of both baseline models by 0.5% to 1.9%. Thus, our proposed MSRA-FPN can effectively improve the accuracy of anchor-based and anchor-free detectors.

### Table 1. Experimental results of comparison with different FPN-based methods and baseline models on PASCAL VOC2007.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input Size</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN [17]+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>72.1</td>
</tr>
<tr>
<td>Faster R-CNN+PANet [9]</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>72.4</td>
</tr>
<tr>
<td>Libra R-CNN [35]+BFP</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>72.4</td>
</tr>
<tr>
<td>RetinaNet+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>71.8</td>
</tr>
<tr>
<td>Reppoints [23]+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>73.1</td>
</tr>
<tr>
<td>FoveaBox [28]+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>68.9</td>
</tr>
<tr>
<td>RetinaNet+CARAFe [8]</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>72.3</td>
</tr>
<tr>
<td>RetinaNet+PANet</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>72.8</td>
</tr>
<tr>
<td>RetinaNet+CE-FPN [12]</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>72.1</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>73.4</td>
</tr>
<tr>
<td>Reppoints+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>74.2</td>
</tr>
<tr>
<td>FoveaBox+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>70.8</td>
</tr>
</tbody>
</table>

### Table 2. Experimental results of comparison with different FPN-based methods and baseline models on PASCAL VOC2007+VOC2012.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input Size</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>76.5</td>
</tr>
<tr>
<td>Libra R-CNN+BFP</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>76.3</td>
</tr>
<tr>
<td>RetinaNet+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>78.1</td>
</tr>
<tr>
<td>FoveaBox+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>74.6</td>
</tr>
<tr>
<td>Reppoints+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>78.2</td>
</tr>
<tr>
<td>RetinaNet+CARAFe</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>77.3</td>
</tr>
<tr>
<td>RetinaNet+PANet</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>78.4</td>
</tr>
<tr>
<td>RetinaNet+CE-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>78.1</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>78.9</td>
</tr>
<tr>
<td>FoveaBox+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>75.8</td>
</tr>
<tr>
<td>Reppoints+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>79.0</td>
</tr>
</tbody>
</table>

A strong backbone can better extract the features of the images, and Table 3 reports the performance of MSRA-FPN with different backbones. From Table 3, it can be seen that the overall performance of the model is enhanced as the backbone parameters increase.

### Table 3. Experimental results of different backbones.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input Size</th>
<th>mAP</th>
<th>FLOPs</th>
<th>Params (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RetinaNet+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>71.8</td>
<td>234.87</td>
<td>31.85</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>EfficientNetB3</td>
<td>640 × 640</td>
<td>73.2</td>
<td>55.95</td>
<td>30.83</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>73.4</td>
<td>92.08</td>
<td>45.65</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet101</td>
<td>640 × 640</td>
<td>73.8</td>
<td>118.78</td>
<td>59.17</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNetX50</td>
<td>640 × 640</td>
<td>73.6</td>
<td>93.31</td>
<td>45.13</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNetX50-DCN</td>
<td>640 × 640</td>
<td>74.1</td>
<td>93.83</td>
<td>46.29</td>
</tr>
</tbody>
</table>
In addition, we also verified the performance of MSRA-FPN at different input scales. As seen in Table 4, the model’s performance improves as the image input size increases.

Table 4. Experimental results for different input sizes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input Size</th>
<th>mAP</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>320 × 320</td>
<td>65.4</td>
<td>23.03</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>73.4</td>
<td>92.08</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>1024 × 640</td>
<td>73.9</td>
<td>230.28</td>
</tr>
</tbody>
</table>

The above four experimental results show that our proposed MSRA-FPN can effectively fuse feature information at different scales and improve the object detection capability of the detector.

**MS COCO:** We also evaluated MSRA-FPN on MS COCO, as shown in Table 5. MSRA-FPN can improve the performance of the baseline model RetinaNet by 0.8% and can effectively improve the detection performance of large objects. As can be seen from the table, MSRA-FPN improves the baseline model AP1 value from 39.8% to 41.6%, an improvement of 1.8%. The same can be done to improve the performance of FCOS [27] by 0.6%.

Table 5. Comparison of the proposed method and the baseline model on the MS COCO dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input Size</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
<th>APs</th>
<th>APm</th>
<th>AP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RetinaNet+FPN</td>
<td>ResNet50</td>
<td>320 × 320</td>
<td>21.4</td>
<td>35.9</td>
<td>22.1</td>
<td>2.1</td>
<td>22.2</td>
<td>39.8</td>
</tr>
<tr>
<td>FCOS+FPN</td>
<td>ResNet50</td>
<td>320 × 320</td>
<td>18.5</td>
<td>32.1</td>
<td>18.6</td>
<td>3.1</td>
<td>18.2</td>
<td>34.2</td>
</tr>
<tr>
<td>RetinaNet+Ours</td>
<td>ResNet50</td>
<td>320 × 320</td>
<td>22.2</td>
<td>37.3</td>
<td>22.3</td>
<td>2.1</td>
<td>22.9</td>
<td>41.6</td>
</tr>
<tr>
<td>FCOS+Ours</td>
<td>ResNet50</td>
<td>320 × 320</td>
<td>19.1</td>
<td>33.2</td>
<td>19.4</td>
<td>3.6</td>
<td>19.6</td>
<td>34.8</td>
</tr>
</tbody>
</table>

**TKFD:** In order to verify the effectiveness of MSRA-FPN in detecting figures in Thangka images, we have also conducted comparison experiments with other state-of-the-art FPN-based methods on TKFD. As shown in Table 6, MSRA-FPN can significantly improve the mAP accuracy of several baseline models in TKFD. When using ResNet50 as the backbone, MSRA-FPN improved RetinaNet, Reppoints, and FoveaBox from 64.7%, 65.8%, and 65.3% to 69.4%, 68.7%, and 69.2%, respectively, on the TKFD test set by 4.7%, 2.9%, and 3.9%. After changing the backbone to ResNeXt50 [39] DCN and expanding the input size to 1024×640, the model achieved an mAP value of 71.2%. Furthermore, when compared with two-stage detectors, such as Faster R-CNN, and single-stage detectors, such as YOLOv3 [22], it is found that our proposed algorithm still has a significant advantage.

4.4. Ablation Study

To analyze the performance of each component in our proposed MSRA-FPN, we conducted ablation experiments on PASCAL VOC2007. This experiment was performed on RetinaNet with ResNet-50 as the backbone and an input image size of 640 × 640.

4.4.1. Ablation Studies for UCLRAM

Table 6 reports the number of parameters, FLOPs, and FPS for multiple models on TKFD. It can be seen that the number of RetinaNet parameters using MSRA-FPN increases from the original 36.62M to 45.77M relative to the baseline model when ResNet50 is used as the backbone and the input size is 640 × 640. In order to balance the conflict between accuracy and the number of parameters, the structure of the UCLRAM was ablated on PASCAL VOC2007 and compared with the structure shown in Figure 4, the results of which are shown in Table 7.
Table 6. Comparison of experimental results on the TKFD validation and test sets. The MSRA-FPN is compared with the baseline model and compared with several FPN-based methods. In addition to this paper, we also try to use some other object detection methods to compare with RetinaNet-MSRA-FPN and also compare their respective number of parameters, inference times, and FLOPs.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input Size</th>
<th>Validation Set mAP</th>
<th>Test Set mAP</th>
<th>FLOPs (M)</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>61.7</td>
<td>68.5</td>
<td>78.3</td>
<td>41.25</td>
</tr>
<tr>
<td>Libra R-CNN+BFP</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>60.7</td>
<td>69.2</td>
<td>78.7</td>
<td>41.51</td>
</tr>
<tr>
<td>Cascade R-CNN [40]+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>63.3</td>
<td>69.9</td>
<td>119</td>
<td>69</td>
</tr>
<tr>
<td>RetinaNet+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>54.1</td>
<td>64.7</td>
<td>86.1</td>
<td>36.62</td>
</tr>
<tr>
<td>FoveaBox+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>59.3</td>
<td>65.3</td>
<td>81.4</td>
<td>36.07</td>
</tr>
<tr>
<td>Reppoints+FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>57.3</td>
<td>65.8</td>
<td>75.9</td>
<td>36.61</td>
</tr>
<tr>
<td>RetinaNet+PANet</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>60.9</td>
<td>65.4</td>
<td>249</td>
<td>35.51</td>
</tr>
<tr>
<td>RetinaNet+CE-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>57.5</td>
<td>65.6</td>
<td>228</td>
<td>57.21</td>
</tr>
<tr>
<td>YOLOV3+YOLOv3Neck</td>
<td>DarkNet53</td>
<td>640 × 640</td>
<td>59.2</td>
<td>66.4</td>
<td>77.8</td>
<td>61.66</td>
</tr>
<tr>
<td>SSD [21]+SSDNeck</td>
<td>VGG16</td>
<td>300 × 300</td>
<td>52.4</td>
<td>58.7</td>
<td>35.6</td>
<td>27.09</td>
</tr>
<tr>
<td>FoveaBox+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>63.1</td>
<td>69.2</td>
<td>85</td>
<td>39.42</td>
</tr>
<tr>
<td>Reppoints+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>61</td>
<td>68.7</td>
<td>82.9</td>
<td>45.75</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>640 × 640</td>
<td>62.1</td>
<td>69.4</td>
<td>93.1</td>
<td>45.77</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50-DCN</td>
<td>640 × 640</td>
<td>65.1</td>
<td>69</td>
<td>87</td>
<td>49.26</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNet50</td>
<td>1024 × 640</td>
<td>63.8</td>
<td>68.9</td>
<td>149</td>
<td>45.77</td>
</tr>
<tr>
<td>RetinaNet+MSRA-FPN</td>
<td>ResNeXt50-DCN</td>
<td>1024 × 640</td>
<td>64.5</td>
<td>71.2</td>
<td>152</td>
<td>46.41</td>
</tr>
</tbody>
</table>

Table 7. Ablation studies of the structure of UCLRAM on PASCAL VOC2007.

<table>
<thead>
<tr>
<th>Version</th>
<th>mAP</th>
<th>FLOPs</th>
<th>Params (M)</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCLRAM-lite1</td>
<td>72.7</td>
<td>90.66</td>
<td>38.34</td>
<td>24.6</td>
</tr>
<tr>
<td>UCLRAM-lite2</td>
<td>72.9</td>
<td>85.71</td>
<td>36.59</td>
<td>24.9</td>
</tr>
<tr>
<td>UCLRAM-RSEM1</td>
<td>73.2</td>
<td>92.08</td>
<td>51.18</td>
<td>23.6</td>
</tr>
<tr>
<td>UCLRAM</td>
<td>73.4</td>
<td>89.25</td>
<td>45.31</td>
<td>24.1</td>
</tr>
</tbody>
</table>

In this case, UCLRAM-lite1 removes the three bottom-up convolutional layers of stage2 and stage3 from the UCLRAM on the structure shown in Figure 4, turning the three-input nodes into two-input nodes. The UCLRAM-lite2 version replaces the convolution of all cross-layer residuals with MaxPooling. As seen in Table 7, the accuracy of the UCLRAM-lite2 version is 0.5% different from the original version. However, its number of parameters is essentially the same as that of the FPN. The FPS is also higher than the structure shown in Figure 4, which better balances the contradiction between accuracy and the number of parameters.

In addition, we explored RSEM’s impact on the UCLRAM structure. The UCLRAM-RSEM2 version uses RSEM as a guide behind all aggregation nodes, and as can be seen from the data in Table 7, it is less accurate than the structure shown in Figure 4, and the number of parameters has increased.

4.4.2. Ablation Studies for Each Component

We analyzed two components in MSRA-FPN and performed an ablation study on the PASCAL VOC2007 dataset. The experimental results are shown in Table 8. As can be seen from the results in Table 8, our proposed UCLRAM structure improves by 1.2% compared with FPN, and the addition of CGB structure improves MSRA-FPN by 0.4%. From this, it can be concluded that each module of our proposed method has a meaningful impact on object detection performance.
Table 8. Ablation studies of each component of MSRA-FPN on PASCAL VOC2007.

<table>
<thead>
<tr>
<th></th>
<th>RSEM</th>
<th>mAP</th>
<th>GFLOPs</th>
<th>Params (M)</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCLRAM ✓</td>
<td></td>
<td>71.8</td>
<td>85.05</td>
<td>36.5</td>
<td>26.1</td>
</tr>
<tr>
<td>✓</td>
<td></td>
<td>73.0</td>
<td>92.08</td>
<td>40.11</td>
<td>24.4</td>
</tr>
<tr>
<td>✓ ✓</td>
<td></td>
<td>73.4</td>
<td>89.25</td>
<td>45.31</td>
<td>24.1</td>
</tr>
</tbody>
</table>

5. Conclusions

We propose an enhanced Feature Pyramid Network MSRA-FPN that aggregates multi-scale information more effectively to combat attenuation during information propagation and enhances the detection and recognition of large objects. In particular, MSRA-FPN uses UCLRAM to converge feature information from different scales to the top layer through a triangular structure from the bottom to enhance the semantic information of the high-level features and then uses RSEM to mitigate the semantic overlap that occurs during the fusion of multi-scale feature maps. The experimental results show that on the PASCAL VOC dataset, our proposed method improves the performance of the baseline model by 0.5–1.9% and is highly competitive with other state-of-the-art FPN-based methods. We also validate the effectiveness of MSRA-FPN on MS COCO. In addition, experiments have been conducted on TKFD to verify the effectiveness of the proposed method for detecting figure objects in Thangkas. However, there are still some problems with the MSRA-FPN. For example, its performance on a two-stage detector is much inferior to that on a single-stage detector. In future work, we will attempt to continue to improve MSRA-FPN for a variety of different types of tasks and detectors and to validate its performance on a larger object detection dataset.

Author Contributions: Conceptualization, H.W.; data curation, H.W.; methodology, H.W.; writing—original draft, H.W.; writing—review and editing, T.W.; funding acquisition, T.W.; supervision, T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62166035); Natural Science Foundation of Gansu Province (21R7RA163); Innovation Project for Young Teachers supported by fundamental Research Funds for the Central Universities (31920210090).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are available from the email: wtj@mail.lzjtu.cn upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.