mathematics

MDPI

Article

MLA-Net: Feature Pyramid Network with Multi-Level Local Attention for Object Detection

Xiaobao Yang [1,2,*], Wentao Wang [3], Junsheng Wu [4], Chen Ding [3], Sugang Ma [3] and Zhiqiang Hou [3]

[1] Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an University of Posts and Telecommunications, Xi'an 710061, China
[2] School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[3] School of Computer Science, Xi'an University of Posts and Telecommunications, Xi'an 710061, China
[4] School of Software, Northwestern Polytechnical University, Xi'an 710072, China
* Correspondence: y78h11b09@xupt.edu.cn

Abstract: Feature pyramid networks and attention mechanisms are the mainstream methods to improve the detection performance of many current models. However, when they are learned jointly, there is a lack of information association between multi-level features. Therefore, this paper proposes a feature pyramid of the multi-level local attention method, dubbed as MLA-Net (Feature Pyramid Network with Multi-Level Local Attention for Object Detection), which aims to establish a correlation mechanism for multi-level local information. First, the original multi-level features are deformed and rectified using the local pixel-rectification module, and global semantic enhancement is achieved through the multi-level spatial-attention module. After that, the original features are further fused through the residual connection to achieve the fusion of contextual features to enhance the feature representation. Extensive ablation experiments were conducted on the MS COCO (Microsoft Common Objects in Context) dataset, and the results demonstrate the effectiveness of the proposed method with a 0.5% enhancement. An improvement of 1.2% was obtained on the PASCAL VOC (Pattern Analysis Statistical Modelling and Computational Learning, Visual Object Classes) dataset, reaching 81.8%, thereby, indicating that the proposed method is robust and can compete with other advanced detection models.

Keywords: object detection; convolutional neural network; self-attention; feature pyramid network

MSC: 68Q04

Citation: Yang, X.; Wang, W.; Wu, J.; Ding, C.; Ma, S.; Hou, Z. MLA-Net: Feature Pyramid Network with Multi-Level Local Attention for Object Detection. *Mathematics* **2022**, *10*, 4789. https://doi.org/10.3390/math10244789

Academic Editor: Ioannis G. Tsoulos

Received: 24 November 2022
Accepted: 13 December 2022
Published: 16 December 2022

1. Introduction

With the development of neural network, many detectors based on CNN and Transformer-based architectures have been proposed in recent years [1,2]. Among them, the feature pyramidal network (FPN) [3] has become an almost necessary and effective component in current object detectors, which significantly improves the performance of detectors by learning multi-scale features for objects of different scales.

In object-detection algorithms, pyramid feature-fusion networks enhance the expressiveness of features mainly on the backbone output, and FPN combines top-down branch and lateral linking to fuse the semantic information of deep features and the location information of shallow features, thereby, opening up the research on object detection through multi-level features. Subsequently, PANet [4] investigated an additional bottom-up information pathway based on FPN to further add deep location and semantic information. Further, in 2020, EfficientDet [5] then proposed a weighted bi-directional FPN, which achieves feature fusion by repeatedly stacking the same bi-directional BiFPN blocks multiple times. Clearly, the FPN-based approach greatly improves the performance of object detection by increasing the information interaction between multi-scale features, which is the key to further enhancing the model performance.

However, these current multi-level information interactions based on pyramidal features lack a focus on potentially salient objects when fusing them. Recently, the superior performance of the self-attention algorithm [6] in the field of natural-language processing has led to their widespread use and rapid development in the field of computer vision. In particular, Non-Local [7] network with self-attention focuses on the connections within a sequence of same-scale pixels. AC-FPN [8] introduced self-attention in the FPN part to design CEM and AM to resolve the conflict between feature map resolution and perceptual fields and to augment the discriminative power of feature-representation operations. However, this scales up the distance to the global level in the long-range correlation process and ignores the ultra-long-range uncorrelated nature of the image features.

From the above, attention is more oriented towards feature interaction between multi-level feature maps than in FPN, where attention and particularly self-attention is more about finding the salience of pixels as weights and filtering the original features with a mask composed of all pixel-corresponding weights. In addition, both channel attention and spatial attention in the attention mechanism facilitate inter-feature information interaction between pixels at the same scale.

Taken together, the current approaches based on FPN and the attention mechanism have certain limitations: (1) a lack of effective communication between multi-level features, (2) although self-attention is effective in improving FPN performance, the processed global features undoubtedly contain more redundant features, and (3) the sequences processed by self-attention contain only single-level features rather than multi-level features. Therefore, how to joint learn between multi-level information interaction, multi-level feature sequences, and local attention is necessary to improve the performance of detectors through better feature representation.

In practical scenarios, the dependency between multi-level local features is more extensive than that between same-scale features, and the semantics of surrounding multi-level features need to be referred when deciding the importance of this feature [9], and the aggregation of multi-level local features as attention units of action is more powerful for the network to learn the salience of features. Inspired by Deformable DETR [10], we proposed a feature pyramid networks with a multi-level local attention method that feeds the multi-level feature maps from the residual backbone network into two parallel branches—a top-down branch and an attention branch—the former being used to complement the semantics lacking in the shallow information and the latter to build up the semantics for multi-level local attention, dubbed as MLA-Net.

We propose a correlation mechanism for multi-level local information, and finally the corresponding layer outputs of the two are fused to generate enhanced features as the detection head input. The proposed approach in this paper can be easily plugged into existing FPN-based models [9,11,12] and trained end-to-end without additional supervision.

2. Related Work

2.1. Advanced Detectors

With the development of deep-learning techniques, detection models are mainly divided into two-stage [13–16], one-stage [2,17,18], and transformer-architecture-based object-detection categories. The representative work of the two-stage detection method, R-CNN [19], first, used selective search to generate region suggestions and then refined the suggesting regions by extracting region features through convolutional networks, which was the first implementation of a deep-learning-based object-detection method.

Later, in order to improve the speed of training and inference, Fast R-CNN [20] extracted the feature map of the whole image using a convolutional network; then used the spatial pyramid pool and the region-of-interest (ROI) pool to generate the region features, respectively; and finally used the region features to refine the suggested regions to improve the accuracy. Faster R- CNN [9] was proposed as a region-suggestion network and developed an end-to-end trainable detector that significantly improved performance and sped up inference, a milestone performance in the development of object-detection

methods. More recently, Cascade R-CNN [17] introduced multi-stage refinement to the faster R-CNN, further enabling more accurate object-location prediction.

One-stage detectors have higher inference speed compared to two-stage detectors. SSD [2] and RetinaNet [12] places anchor frames densely on multi-level features and makes predictions based on these anchors. FCOS [11] was proposed with the concept of being anchor-free, which eliminated the design of anchor frames and resulted in certain problems, such as the Intersection over Union (IoU) computation yet improved the performance.

Both detection models of two-stage and one-stage select the positive samples by the assignment of anchor box, while DETR [18] based on the transformer architecture dispensed with anchor box assignment, IoU computation, non-maximal suppression, and other operations, and pioneered a new architecture for detection using learnable anchor boxes and bipartite graph matching. Furthermore, SMCA-DETR [16] designed a plug-and-play (Spatially Modulated Co-Attention) module to replace the existing co-attention mechanism in DETR and achieved faster convergence and higher performance with simple modifications. Deformable DETR [10] introduced the deformable attention module that accepts multi-level feature inputs and makes several improvements to DETR to achieve higher performance.

In summary, these works have made significant progress in different ways. This paper continues to investigate how to better exploit multi-level features to solve the feature representation problem in detection using the pipeline flow of the one-stage object detector with the anchor box assignment approach.

2.2. Feature Fusion Networks

FPN [3] constructed an effective framework for solving the scale variation problem by fusing features via a top-down path. The problem has since been widely applied and further investigated. PANet [4] was investigated with an additional bottom-up path to further exploit low-level information. NAS-FPN [13] uses a neural architecture search to better learn all cross-scale connections. EfficientDet [5] was proposed with a weighted BiFPN for simple and fast feature fusion. PSPNet [14] uses a pyramidal pooling approach to extract hierarchical global contexts. The literature [15] proposed a contextual optimization algorithm to optimize the proposals for each region. In this paper, we add a local pixel-rectification module and multi-level spatial-attention module to enhance the feature representation in the feature-fusion network part.

2.3. Attention

SE [21] compressed each 2D feature map by simply compressing it and, thus, efficiently constructing interdependencies between channels. CBAM [22] took this idea further by introducing spatial-information encoding through large scale kernel convolution. Later studies, such as GENet [23] and GALA [24], extended this idea by employing different spatial attention mechanisms and designing advanced attention blocks. Non-local [25] or self-attention focuses on constructing feature correlation matrices to generate linear transformation weights between two features. Typical examples include GCNet [7] and CCNet [26], both of which used the self-attention mechanism to capture different types of spatial information. The local pixel-rectification module and the multi-level spatial-attention module proposed in this paper were inspired by self-attention and use multi-level local feature sequences for correlation operations to, thus, enhance saliency features.

3. Our Method

3.1. General Framework

The general pipeline process for object detection is to extract features from an image using a classification network as the backbone, to use a feature pyramid network to feature-enhance the extracted features, and then the output multi-level feature map is fed to the detection head to make predictions for each scale object. In the paper, we use RetinaNet, a representative work of one-stage object detection, as a benchmark, and improve the feature

pyramid part along with higher quality features to enable the downstream detection head to better perform the classification and regression tasks.

To alleviate the information loss of salient features, based on the baseline model, we propose two components in the attention branch, which are the local pixel-rectification module and the multi-level spatial-attention module. The overall structure of the network is shown in Figure 1. The feature-fusion network proposed in this paper mainly contains two parallel branches—the top-down branch and the attention branch, where the top-down branch inherits the traditional FPN, and the attention branch consists of the local pixel-rectification module and the multi-scale spatial-attention module.

Given that the output layer of the backbone network is {C3, C4, C5} and the step size is {8, 16, 32} compared to the input image, the method in this paper uses channel reduction to form {F3, F4, F5}, after which the top-down branch simply fuses the deep information into the shallow layer, and the attention branch uses attention to provide additional attention to the salient features of F3, F4, and F5. Finally, the features from the two branches are fused to generate the five-layer detection features {P3, P4, P5, P6, P7} in the Retina network, where P6 is obtained by 3×3 convolution of P5 and P7 is obtained by 3×3 convolution of P6.

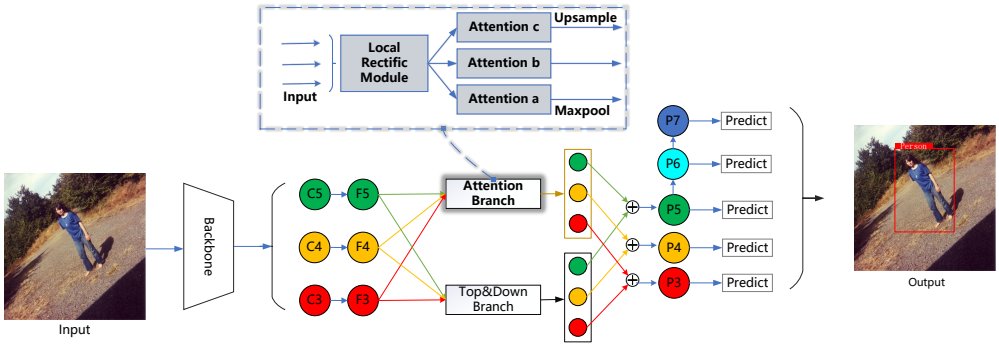

Figure 1. Model framework of our MLA-Net. The outputs C3, C4, and C5 of the backbone network are changed to F3, F4, and F5 through channel reduction. F3, F4, and F5 are fused through two parallel branches to generate a feature map that predicts all scale objects.

After that, we continue to use the detector head and loss function of the benchmark [12]. In short, the five-layer feature map deals with the prediction of objects of different scales, which are then transmitted to the detection head. The model generates the prediction of the category and the regression vector of the bounding box at the detection head. The final loss calculation includes classification loss and regression loss. We use the CE function to calculate the classification loss and the smooth L1 function to calculate the regression loss, which are formulated as the following:

$$LOSS = ClsLoss + RegLoss \tag{1}$$

$$ClsLoss = \frac{1}{N}\Sigma_i^N L_i \tag{2}$$

$$L_i = \Sigma_c^M L_{ic} \tag{3}$$

$$L_{ic} = \begin{cases} -(1-p)^r \times \log(p), & if\ ct = 1 \\ -p^r \times \log(1-p), & if\ ct = 0 \end{cases} \tag{4}$$

$$RegLoss = \frac{1}{N'}\Sigma_i^{N'} |r_i - rt_i| \tag{5}$$

The loss calculation of each input picture is shown in Formula (1). This is composed of the classification loss (ClsLoss) and regression loss (RegLoss). The classification loss is the average of the classification loss of N selected anchor boxes as shown in Formula (2).

The classification loss of each sample is the sum of the binary losses of M categories of this sample as shown in Formula (3). Furthermore, Formula (4) is the expression of the binary loss function, where ct is the supervision signal, p is the prediction signal, and r is the manually set super parameter. Regression loss is a simple smooth L1 loss function, such as Formula (5). This is the average loss of N' samples, where N' is the number of positive samples for training, r_i is the prediction signal of anchor regression, and rt_i is the supervision signal of anchor regression.

3.2. Local Pixel-Rectification Module

In the pipeline flow of Figure 1, the three-level feature map (C3, C4, and C5) output by the backbone network has a pyramidal structure. In this paper, we redesign a multi-level sliding window on this multi-level feature map inspired by ACMix [27], and input the sequence of pixels in the window at each slide into the attention module to obtain the attention weight of each pixel in the window. The window is scanned at each slide not at a single scale of the pyramid but at three scales. The local pixel-rectification module is responsible for adapting the reduced tertiary features {F3, F4, F5} of the channel to the sequence-based data format required by the multi-level spatial-attention module.

Specifically, as shown in Figure 2, given a batch size of C at training, the multi-level $(H_i \times W_i)$ feature map obtained after channel reduction is F^i, $i \in [3, 4, 5]$, where a 2D sliding window with step S_i, kernel size PS_i, hole rate DR_i, and filled pixels P_i slides over it synchronously, with each synchronous slide scanning $PS_i \times PS_i$ pixels and where the sampling range varies according to the void rate. If the actual sampling range of the window is denoted by WS_i , then:

$$WS_i = PS_i + (PS_i - 1) \times (DR_i - 1) \tag{6}$$

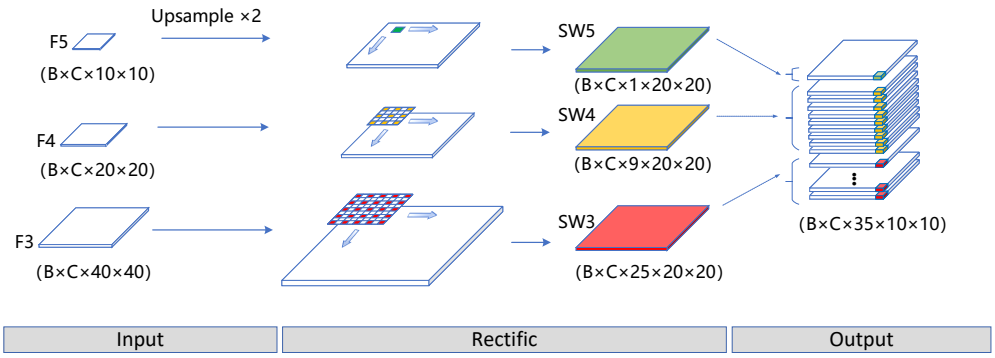

Figure 2. Diagram of the local pixel-rectification operation. Three 2D windows slide synchronously in different steps to simulate the sliding of 3D windows. During this period, the pixels from each sliding scan are spliced and stored in additional dimensions.

After multiple sliding rectification in the horizontal and vertical directions, the output feature map SW^i has a data shape of $C \times (PS_i \times PS_i) \times H \times W$, where H is equal to the maximum number of vertical slides and W is equal to the maximum number of horizontal slides according to the following formula.

$$H = \frac{W_i + 2 \times P_i - WS_i}{S_i} + 1 \tag{7}$$

$$W = \frac{H_i + 2 \times P_i - WS_i}{S_i} + 1 \tag{8}$$

The sequence of $PS_i \times PS_i$ pixels at the (m, n) position of SW^i is denoted as $SW^i_{m,n}$ and $F^i_{r,c}$ is the pixel point in the rth row and cth column on the feature map F^i, and thus their correspondence can be expressed as:

$$
\begin{aligned}
&SW^{i}_{m,n} = \{F^{i}_{r,c}\} \\
&r = i \times S_i + \frac{k}{S_i \times DR_i \times WS_i + K\%S_i \times DR_i} \\
&c = j \times S_i + \frac{k^{'}}{S_i \times DR_i \times WS_i + K'\%S_i \times DR_i} \\
&k', k \in [0, ps_i]
\end{aligned} \tag{9}
$$

This module uses the synchronous sliding of three 2D windows, which equate to a 3D multi-level sliding window, and the sequence of features extracted from the multi-level sliding window is fed into the sequence-based self-attention algorithm model at each synchronous sliding.

3.3. Multi-Level Spatial Attention

Although the local pixel-rectification module in Section 3.1 extracts a large range of features under different perceptual fields, not all features are beneficial to the model's classification and regression of the object. Feature information is prone to information decay as the network depth increases, and too many redundant features will actually degrade the detection performance. Therefore, in order to eliminate redundant information and emphasize effective information, a multi-level spatial-attention module is inserted after the local feature pixel rectification to further enhance the expressiveness of the feature mapping, to suppress redundant information among many features, and to better exploit the semantic correlation between two features to better establish the mapping.

The output of the local pixel-rectification module is processed by the self-attention module and mapped into enhanced features at all levels in two ways—Type A mapping and Type B mapping as shown in Figure 3. The multi-level spatial attention first adds position encoding to the feature sequence, and then adds corresponding hierarchical encoding for each feature scale, and after the self-attention processing, finally builds the FC (fully connected mapping) on the one hand and also builds the mapping using the insertion of Tokens (tokens) in ViT [28]. The mapped features are then summed, fused, and output. Type B attention only builds mappings in the form of categorical tokens.

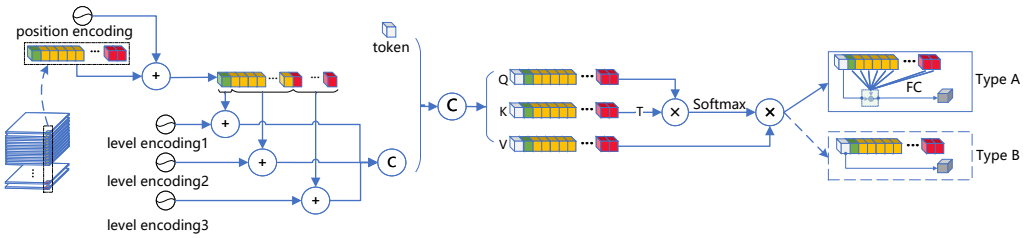

Figure 3. Multi-level spatial attention Type A/B. The rectified features can be processed by an attention algorithm based on the sequence. The processing flow as shown in the figure is the detailed flow of attention. There are two processing methods at the end of processing.

As shown in Figure 1, attention components a, b, and c are responsible for the enhancement of Level 3, 4, and 5 features, respectively, with attention component a using type-A mapping and attention component b and c using type-B mapping. Overall, this module establishes semantic associations and spatial associations between each region-of-interest feature and each level feature and features at the same level through the above two approaches, thereby, allowing the network to actively aggregate contextual information of sub-regions at each level for each region-of-interest feature, i.e., to establish semantic dependencies for the features at each level generated by the rectifier module, thus, providing the output features with clearer semantics.

4. Experiments

4.1. Datasets

The PASCAL VOC dataset can be divided into four major categories and twenty sub-categories. A total of 9963 images were used in PASCAL VOC2007, of which 5011 images were in the training set and 4952 images were in the test set; 122,216 images were used in MS COCO, of which 117,266 images were in the training set with 4952 images in the test set. In this paper, the ablation experiments were conducted on the MS COCO dataset and PASCAL VOC. Furthermore, we compared the results with various algorithms on the PASCAL VOC dataset in our experiments. We use python 3.8 and pytorch 1.7.1 to run all our experiments. Two NVIDIA GeForce GTX 1080Ti GPUs were used for training, and only one was used for evaluation.

4.2. Experimental Setup

The ResNet model, pre-trained on the ImageNet [28] dataset, was used as the backbone feature extraction network, and the baseline RetinaNet and the proposed method were trained by first pre-processing the input images with data enhancement operations, such as image flipping, aspect warping, and color scrambling, using a SGD optimizer, and the initial learning rate was set to 0.0025 during training. The total number of training epoches for the experiments on MS COCO was 15 epoches, and the learning rate decreased to one-tenth of the original rate in epoch 8 and 12. For Pascal VOC, the training epoches were set to 16, and the learning rate decreased to one-tenth of the original rate in epoch 12.

4.3. Ablation Studies

In order to verify the effectiveness of the proposed method, extensive ablation experiments were conducted on the MS COCO dataset and PASCAL VOC for the proposed algorithm. Relevant experiments were conducted on the top-down branching effectiveness, the method of fusion of the two major branches, the way of mapping the features after attention, the importance of shared and independent linear mappings at each level of attention, the number of attention heads, and the number of channels, and the experimental results are shown in the following tables. In the tables, we use checkmarks to indicate that the corresponding model of the current row is configured with the corresponding network structure or algorithm of the current column. The horizontal bar indicates that the corresponding network structure or algorithm of the current column cannot be installed in the corresponding model of the current row. The best result in this experiment is shown in bold.

The model described in Section 3.1 of this paper has two major parallel branches compared to the original benchmark top-down fusion network—namely, the top-down branch and the attention branch. The results in Table 1 show that the model lacking the top-down branch performs slightly lower than the benchmark, indicating that the top-down branch is a crucial structure.

Table 1. Comparison of top-down and attention branches.

Models	Dataset	Top-Down Branch	Attention Branch	mAP
RetinaNet-base	MS COCO	√		31.7
RetinaNet-ours	MS COCO		√	31.5
RetinaNet-ours	MS COCO	√	√	**32.4**

As show in Table 2, this paper explores the best way to fuse the two main branches, where splicing fusion is the operation of stitching two features together and then reducing them using a convolutional channel, which results in more computation and parameters, but is less effective compared to summation fusion, which is not only simple but also very effective.

Table 2. Comparison of the branch-fusion approaches.

Models	Dataset	Cat	Add	mAP
RetinaNet-base	MS COCO	-	-	31.7
RetinaNet-ours	MS COCO		√	**33.1**
RetinaNet-ours	MS COCO	√		32.4

After determining the basic branching and fusion methods, a series of experiments were performed in this paper for the internal structure of the multi-level spatial-attention module. As shown in Figure 1, attention components a, b, and c in the attention branch are responsible for the enhancement of small, medium, and large scale features, respectively, as can be seen from the results in Table 3: shallow features are responsible for small-object detection, and small objects should focus more on the salience of the features associated with them due to their lesser association with the environment. Finally, we conducted experiments on the performance enhancement of the number of self-attention heads and the number of channels per head. The experimental results in Table 4 demonstrate that increasing the number of heads was more effective in improving the performance when compared with increasing the number of channels per head.

Table 3. Comparison of the performance of attention components a, b, and c with type A/B.

Models	Dataset	Type A	Type B	mAP
RetinaNet-base	MS COCO	-	-	31.7
RetinaNet-ours	MS COCO	-	a, b, c	32.9
RetinaNet-ours	MS COCO	a, b, c	-	32.9
RetinaNet-ours	MS COCO	b, c	a	**33.1**

Table 4. Comparison of different numbers of attention heads and numbers of channels.

Models	Dataset	Number of Heads	Number of Channels	mAP
RetinaNet-base	MS COCO	-	-	31.7
RetinaNet-ours	MS COCO	3	64	32.3
RetinaNet-ours	MS COCO	6	32	**33.1**

In this paper, we next take the above configuration and apply it to the PASCAL VOC dataset to investigate the improvement of detection performance with multi-level features and attention mechanisms with local windows, respectively. For the effect of local windows on attention, this paper set different window sizes in the local pixel-rectification module. The corresponding window sizes for each level are shown in Table 5, and the corresponding performance is as follows.

Local features with appropriate range sizes are clearly better, and it can be seen that global attention is not the best choice. As for the effect of multi-level features on attention, this paper controlled the scale of the features from the input of the local pixel-rectification module underhand as shown in Table 6. In the attention with multi-level features, each scale did not contribute equally to the attention, and the shallow features did not contribute. No attention mechanism functioned well if the semantic information of the feature was insufficient.

Table 5. Comparison of different window sizes under Levels 3, 4, and 5 for the local pixel-rectification module.

Models	Dataset	Window Size	mAP
RetinaNet-base	PASCAL VOC	-	80.6
RetinaNet-ours	PASCAL VOC	3, 3, 1	80.9
RetinaNet-ours	PASCAL VOC	5, 5, 3	**81.6**
RetinaNet-ours	PASCAL VOC	9, 9, 5	80.9

Combining the experiments of two datasets, the performance of our proposed method for each backbone network and dataset is shown in Table 7 below. In general, the method in this paper increases significantly the performance compared to the baseline on ResNet-18 but less on ResNet-50 and ResNet-101, and increases significantly the gain of mAP on PASCAL VOC but less on MS COCO, which we will study in depth for improvements in the future.

Table 6. Comparison of the performance based on each input of multi-level spatial attention.

Models	Dataset	Multi-Level Input of Attention	mAP
RetinaNet-base	PASCAL VOC	-	80.6
RetinaNet-ours	PASCAL VOC	F3	81.7
RetinaNet-ours	PASCAL VOC	F4	81.6
RetinaNet-ours	PASCAL VOC	F5	81.4
RetinaNet-ours	PASCAL VOC	F3, F4	81.2
RetinaNet-ours	PASCAL VOC	F3, F5	81.2
RetinaNet-ours	PASCAL VOC	F4, F5	**81.8**
RetinaNet-ours	PASCAL VOC	F3, F4, F5	81.6

Table 7. Comparison of the performance based on different backbone networks.

Models	Dataset	Training Strategies	Backbone Network	mAP
RetinaNet-ours	MS COCO	1×	ResNet-18	33.1 (+1.4)
RetinaNet-ours	MS COCO	1×	ResNet-50	36.8 (+0.5)
RetinaNet-ours	MS COCO	1×	ResNet-101	39.0 (+0.5)
RetinaNet-ours	PASCAL VOC	1×	ResNet-50	80.3 (+2.0)
RetinaNet-ours	PASCAL VOC	1×	ResNet-101	81.8 (+1.2)

4.4. Qualitative Analysis

As shown in Figure 4, to illustrate the detection performance of this paper's algorithm, the detection results of the original RetinaNet detection algorithm are compared with this paper's algorithm, and some of the clearly representative images were selected for illustration. In the first row of the figure, the algorithm in this paper has no redundant prediction results and detects the object with a higher confidence level; however, the object box position of the algorithm in this paper is more accurate and has a higher confidence score.

In the second row of the figure, for people, the RetinaNet algorithm has no false detection of people, and the algorithm in this paper is able to give better detection results. In the third row of plots, the RetinaNet algorithm misdetects the bird as a sheep, while the algorithm in this paper gives a high confidence level of correct judgment. In the fourth row, the algorithm does not have low-quality redundant detection compared to RetinaNet.

The experiments show that the algorithm in this paper was generally able to detect the object class and give a certain confidence score, and that the overall confidence score and the accuracy of the object frame were higher than for the original algorithm.

(**a**) (**b**) (**c**)

Figure 4. Experimental results on PASCAL VOC. Each row is a few representative pictures selected by us. Columns (**a**–**c**) represent the ground truth, the prediction of the benchmark algorithm, and the prediction of our algorithm. As shown in the figure, our algorithm had better performance. (**a**) denotes the annotations of GT. (**b**) represents the detection result of RetinaNet. (**c**) illustrates the result of ours.

4.5. Comparison with State-of-the-Art Models

4.5.1. Comparison of Performance on PASCAL VOC2007

The results of the comparison between the algorithm in this paper and the mainstream object-detection algorithm on the PASCAL VOC dataset are shown in Table 8. The algorithm in this paper achieves a detection accuracy of 81.8%, which is a 1.2% improvement compared to the benchmark algorithm. Compared with the two-stage classical algorithms Faster-RCNN (ResNet-101) and R-FCN, the detection accuracy of this algorithm was improved by 5.4% and 2.3%, respectively. Compared with the one-stage detection algorithms SSD, RetinaNet, YOLOv3, YOLOX-s, and DSSD, the detection accuracy was improved by 5.0%, 2.3%, 2.5%, 0.8%, and 0.3%. Compared with the one-stage detection algorithm CenterNet, the detection accuracy was improved by 3.1%.

Table 8. Comparison of the mAP performance based on the PASCAL VOC2007 dataset with different advanced models.

Models	Dataset	Network	Resolution	mAP (%)
Faster-RCNN	VOC07+12	VGG-16	1000 × 600	73.2
Faster-RCNN	VOC07+12	ResNet-101	1000 × 600	76.4
SSD	VOC07+12	VGG-16	512 × 512	76.8
YOLOv3	VOC07+12	DarkNet-53	544 × 544	79.3
CenterNet	VOC07+12	ResNet-101	512 × 512	78.7
DSSD	VOC07+12	ResNet-101	513 × 513	81.5
R-FCN	VOC07+12	ResNet-101	1000 × 600	79.5
RetinaNet	VOC07+12	ResNet-50	(800,1333)	78.5
RetinaNet	VOC07+12	ResNet-50	(600,1000)	77.3
ExtremeNet	VOC07+12	Hourglass-104	512 × 512	79.5
YOLOX-S	VOC07+12	DarkNet-53	640 × 640	81.0
RetinaNet+Ours	VOC07+12	ResNet-50	(600,1000)	80.3
RetinaNet+Ours	VOC07+12	ResNet-101	(600,1000)	**81.8**

4.5.2. Comparison of Single-Category Performance

The detection accuracy of each category of this paper's algorithm applied to RetinaNet was compared with other algorithms on the PASCAL VOC dataset, and the results are shown in Table 9. The algorithm in this paper was able to reach the optimal level in all 10 categories, where seven categories increased by more than 1%.

Table 9. Comparison of the mAP performance based on the PASCAL VOC2007 dataset with different advanced models.

Class	Ours	YOLOX-S	Faster R-CNN	R-FCN	SSD512	RetinaNet	CenterNet-DLA
aero	85.8	86.5	76.5	82.5	82.4	**89.4**	85.0
bike	87.9	**89.5**	79.0	83.7	84.7	86.6	86.0
bird	**84.4**	77.3	70.9	80.3	78.4	79.8	81.4
boat	**74.7**	73.9	66.5	69.0	73.8	67.8	72.8
bottle	**72.2**	71.6	52.1	69.2	53.2	70.8	68.4
bus	86.6	**88.2**	83.1	87.5	86.2	85.4	86.0
car	88.9	**91.9**	84.7	88.4	87.5	90.5	88.4
cat	**89.4**	87.4	86.4	88.4	86.0	88.8	86.5
chair	**68.7**	66.7	52.0	65.4	57.8	61.0	65.0
cow	86.2	82.0	81.9	**87.3**	83.1	75.6	86.3
table	72.0	**79.6**	65.7	72.1	70.2	65.8	77.6
dog	**88.9**	82.9	84.8	87.9	84.9	84.1	85.2
horse	87.2	**89.1**	84.6	88.3	85.2	84.4	87.0
mbike	84.9	**86.7**	77.5	81.3	83.9	84.9	86.1
person	85.5	**88.7**	76.7	79.8	79.7	85.7	85.0
plant	**59.1**	53.9	38.8	54.1	50.3	52.1	58.1
sheep	**84.0**	78.3	73.6	79.6	77.9	77.7	83.4
sofa	**80.8**	79.8	73.9	78.8	73.9	74.2	79.6
train	86.0	86.3	83.0	**87.1**	82.5	85.8	85.0
tv	**83.9**	79.0	72.6	79.5	75.3	79.6	80.3

4.5.3. Discussion

The method in this paper improved the performance substantially with the introduction of a small number of additional parameters. Furthermore, the method in this paper was applied to RetinaNet (ResNet-50) with fewer parameters than RetinaNet (ResNet-101) to obtain better performance. The results show that the improvement brought by the method comes mainly from the fine-grained design rather than additional parameters.

To further evaluate the impact of multi-level contextual information on the attention mechanism, we used different input tiers for the local pixel-rectification module with

different kernels and expansion rates for individual inputs. Feature dependencies at large scales had small improvements in the detection performance. This situation suggests that a larger range of local features tends to introduce more redundant features, that appropriately sized local regions can both highlight salient features using inter-feature associations and avoid introducing redundant features, and that multiple feature tiers provide a significant boost to the attention mechanism. We conclude that local attention is more effective than global attention in FPN and that multi-level attention is more effective than single-scale.

5. Conclusions

In this paper, we proposed MLA-Net. The most significant processes are that the pyramid features were scanned and extracted by a local rectification module, and were subjected to multi-level spatial attention to output a feature map. While our approach was effective on the PASCAL VOC dataset, it was not as effective on MS COCO. The effect of semantic information on detection accuracy will be further investigated in subsequent studies to obtain a more representative higher-order feature representation.

Author Contributions: X.Y.'s contribution: methods, manuscript preparation, equipment resource support, verification, and data management. W.W.'s contribution: experimentation, review, editing, and supervision. S.M.'s contribution: discussion, and resource equipment support. C.D.'s contribution: discussion, and review. J.W.'s contribution: discussion, and review. Z.H.'s contribution: review. 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 (Nos. 62072370 and 61901369) and the Shaanxi Provincial Science Foundation (No. 022JQ-577).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are openly available at [y78h11b09@github.com].

Acknowledgments: In particular, the authors would like to thank Ningbo Li for the discussion and Shuai He for polishing the paper.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FPN	Feature Pyramid Networks
MS COCO	Microsoft Common Objects in Context
PASCAL VOC	Pattern Analysis Statistical Modelling and Computational Learning, Visual Object Classes
MLA-NET	Feature Pyramid Network with Multi-Level Local Attention for Object Detection
IoU	Intersection over Union
mAP	mean Average Precision
SGD	Stochastic Gradient Descent
CE	Cross Entropy
CNN	Convolutional Neural Network

References

1. Chi, C.; Wei, F.; Hu, H. RelationNet++: Bridging visual representations for object detection via transformer decoder. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 13564–13574.
2. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector-V1. *Lect. Notes Comput. Sci.* **2016**, *9905*, 21–37.
3. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.

4. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
5. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
6. Zhang, D.; Zhang, H.; Tang, J.; Wang, M.; Hua, X.; Sun, Q. Feature pyramid transformer. *Lect. Notes Comput. Sci.* **2020**, *12373*, 323–339.
7. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1971–1980.
8. Cao, J.; Chen, Q.; Guo, J.; Shi, R. Attention-guided Context Feature Pyramid Network for Object Detection. *arXiv* **2020**, arXiv:2005.11475.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
10. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. *arXiv* **2010**, arXiv:2010.04159. Available online: https://arxiv.org/abs/2010.04159 (accessed on 15 December 2022).
11. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully convolutional one-stage object detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9627–9636.
12. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; DollÃąr, P. Focal loss for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
13. Ghiasi, G.; Lin, T.Y.; Le, Q.V. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7029–7038.
14. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Venice, Italy, 22–29 October 2017; pp. 6230–6239.
15. Tripathi, S.; Lipton, Z.C.; Belongie, S.; Nguyen, T. Context matters: Refining object detection in video with recurrent neural networks. In Proceedings of the British Machine Vision Conference, York, UK, 19–22 September 2016; pp. 1–12.
16. Gao, P.; Zheng, M.; Wang, X.; Dai, J.; Li, H. Fast convergence of DETR with spatially modulated co-attention. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 3601–3610.
17. Cai, Z.; Vasconcelos, N. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1483–1498. [CrossRef] [PubMed]
18. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End object detection with transformers. *Lect. Notes Comput. Sci.* **2020**, *12346*, 213–229.
19. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
20. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 11–18 December 2015; pp. 1440–1448.
21. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
22. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional block attention module. *Lect. Notes Comput. Sci.* **2018**, *11211*, 3–19.
23. Hu, J.; Shen, L.; Albanie, S.; Sun, G.;Vedaldi, A. Gather-excite: Exploiting feature context in convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2018**, *2018*, 9401–9411.
24. Park, J.; Lee, M.; Chang, H.J.; Lee, K.; Choi, J.Y. Symmetric graph convolutional autoencoder for unsupervised graph representation learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6518–6527.
25. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
26. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. CCNet: Criss-cross attention for semantic segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 603–612.
27. Chen, Q.; Wu, Q.; Wang, J.; Hu, Q.; Hu, T.; Ding, E.; Cheng, J.; Wang, J. MixFormer: Mixing Features across Windows and Dimensions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, Louisiana, 19–24 June 2022; pp. 5239–5249.
28. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 ÃŮ 16 words: Transformers for image recognition at scale. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 4–8 May 2021; pp. 103–124.