applied sciences

MDPI

Article

An Efficient Module for Instance Segmentation Based on Multi-Level Features and Attention Mechanisms

Yingchun Sun, Wang Gao *, Shuguo Pan *, Tao Zhao and Yahui Peng

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China; 220193288@seu.edu.cn (Y.S.); zhaotao@seu.edu.cn (T.Z.); 220203443@seu.edu.cn (Y.P.)
* Correspondence: gaowang1990@seu.edu.cn (W.G.); psg@seu.edu.cn (S.P.)

Featured Application: The proposed method has a potential application value in assisting the normal driving of unmanned delivery vehicles and unmanned cleaning vehicles in urban street scenes. It can aid unmanned vehicles to detect and segment surrounding objects and plan safe driving routes to avoid obstacles according to the results of instance segmentation.

Abstract: Recently, multi-level feature networks have been extensively used in instance segmentation. However, because not all features are beneficial to instance segmentation tasks, the performance of networks cannot be adequately improved by synthesizing multi-level convolutional features indiscriminately. In order to solve the problem, an attention-based feature pyramid module (AFPM) is proposed, which integrates the attention mechanism on the basis of a multi-level feature pyramid network to efficiently and pertinently extract the high-level semantic features and low-level spatial structure features; for instance, segmentation. Firstly, we adopt a convolutional block attention module (CBAM) into feature extraction, and sequentially generate attention maps which focus on instance-related features along the channel and spatial dimensions. Secondly, we build inter-dimensional dependencies through a convolutional triplet attention module (CTAM) in lateral attention connections, which is used to propagate a helpful semantic feature map and filter redundant informative features irrelevant to instance objects. Finally, we construct branches for feature enhancement to strengthen detailed information to boost the entire feature hierarchy of the network. The experimental results on the Cityscapes dataset manifest that the proposed module outperforms other excellent methods under different evaluation metrics and effectively upgrades the performance of the instance segmentation method.

Keywords: AFPM; multi-level features; inter-dimensional interaction; attention mechanism; instance segmentation

check for updates

Citation: Sun, Y.; Gao, W.; Pan, S.; Zhao, T.; Peng, Y. An Efficient Module for Instance Segmentation Based on Multi-Level Features and Attention Mechanisms. *Appl. Sci.* **2021**, *11*, 968. https://doi.org/10.3390/app11030968

Received: 22 December 2020
Accepted: 18 January 2021
Published: 21 January 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Object instance segmentation [1] is one of the most challenging tasks in computer vision, which needs to locate the object position in the image, classify it, and segment the pixels accurately [2]. The instance segmentation technology can be applied in many fields. For example, in industrial robotics, instance segmentation algorithms can detect and segment parts in different backgrounds, improve the efficiency of automatic assembly and reduce labor cost. Moreover, it can be used for tumor image segmentation, to carry diagnosis and for other aspects to assist the treatment of diseases in terms of intelligent medicine. Especially in autonomous driving, instance segmentation technology can be applied to the perception system of automatic driving vehicles to detect and segment pedestrians, cars and objects in the driving environment, and it can provide data support for the decision-making of automatic driving vehicles. Furthermore, the instance segmentation methods can accomplish the segmentation of obstacles in the image captured by the vehicle camera so as to facilitate the subsequent estimation of its trajectory and ensure safe driving [3].

At present, the instance segmentation methods based on a convolutional neural network (CNN) are mainly divided into two categories, namely, one-stage instance segmentation methods and two-stage instance segmentation methods [4]. The two-stage instance segmentation methods contain two ideas: detection followed by segmentation [5–8] and embedding cluster [9–11]. Two-stage instance segmentation methods which are based on the principle of detecting then segmentation first exploit object detection algorithms to find the bounding box of the instance, and then perform semantic segmentation algorithms in the detection box and output the segmentation results as different instances. Two-stage instance segmentation methods with the foundation principle of embedding cluster perform semantic segmentation at a pixel level in images, and then different instances are distinguished by clustering and metric learning. The average precision of the instance segmentation methods on account of two steps is not satisfactory when segmenting crowded, occluded as well as irregular objects, and the speed of generating low resolution mask is not ideal.

There are also two categories of one-stage instance segmentation methods according to different solutions. One is inspired by the one-stage anchor-based object detection algorithms, forgoing the sequential execution steps of two-stage instance segmentation methods and making the network learn to locate the instance mask through a related parallel design [12]. The other is aroused by one-stage anchor-free object detection methods which get rid of the limitation of anchor location and scale in structure, and rely on a dense prediction network to achieve precision object detection and segmentation [13]. One-stage instance segmentation methods have more advantages in inference time and need to be further improved in accuracy. Generally, two-stage instance segmentation methods can achieve slightly higher accuracy compared with the one-stage instance segmentation methods, but the inference speed of mask generation is slower.

It is difficult for single-layer feature maps, whether high-level or low-level, to cope well with the scale change of instance objects and the imbalance of category data. Therefore, multi-level feature networks are more and more widely used in instance segmentation algorithms to meet the challenge [14]. By fusing the detailed location information of low-level features and rich semantic information of high-level features [15], multi-level feature networks can enhance the representation capacity of features and provide more abundant and beneficial information for detection and segmentation. However, due to the different contributions of different feature maps or even different regions in the same feature map to the object, the features obtained by the multi-level feature networks are sweeping and multifarious, which cannot meet the requirements of the task accurately. Consequently, it is necessary to screen the information extracted from the multi-level network, and improve the performance of the instance segmentation method by biasing the allocation of usable computational resources to the most informative components [16].

An attention mechanism has been successfully applied to many computer vision tasks, such as object recognition and pose estimation, because it can assist the network to choose efficient features pertinently and enhance the learning ability of the network [17]. Furthermore, the rapid development of the attention mechanism also shows that the attention module makes the model pay more attention to the region of the image related to the object, filters out the feature map that interferes with the task, and helps the subsequent neural network precisely select effective features through learning [18,19]. Consequently, the combination of the attention mechanism and the multi-level network can be conducive to the instance segmentation method to extract efficacious features related to the object.

In this paper, the distinctive structure of an attention-based feature pyramid module (AFPM) is proposed for instance segmentation. The AFPM combining the attention mechanism and branches used to enhance location information based on feature pyramid networks (FPN) [15] is composed of feature extraction, lateral attention connections and feature enhancement. Specifically, we apply a convolutional block attention module (CBAM) [20] in bottom-up feature extraction architecture to increase the attention of a multi-level network to instance-related features. Then, a convolutional triple attention

module (CTAM) [21] is included in lateral connections to filter the redundant information in network by capturing the interaction of cross-dimension between the spatial and channel dimension. Finally, we exploit the branches of strengthening spatial information to improve entire feature hierarchy; for instance, segmentation without additional parameters. The experimental results show that the proposed module can significantly boost the performance of instance segmentation on the Cityscapes dataset.

2. Related Work

There are a number of approaches; for instance, segmentation. Mask R-CNN (Region-CNN) [6] increased a branch on the basis of Faster R-CNN [22], which can detect and segment the instance objects efficiently. Succeeding Mask R-CNN, Li et al. [7] proposed an end-to-end instance segmentation method based on the fully convolutional network by introducing position-sensitive internal and external score maps. To solve the problem of instance segmentation, sequential grouping networks (SGN) [23] gradually constructed object instance mask through a series of sub-grouping networks, and can tackled the problem of object occlusion faced by instance segmentation. To enhance information flow in proposal-based framework, Liu et al. extended Mask R-CNN by adding a bottom-up path augmentation and presents adaptive feature pooling to avoid arbitrary allocation of proposal [14]. In order to deal with the instance-aware features and semantic segmentation labels simultaneously, single-shot instance segmentation with affinity pyramid networks (SSAP) [24] was proposed, which was a proposal-free instance segmentation method and calculated the probability that two pixels pertained to the same object in a hierarchical way according to a pixel-pair affinity pyramid. However, the real-time problem of instance segmentation was not completely solved. Bolya et al. [12] decomposed the instance segmentation problem into two parallel subtasks to improve real-time performance, and combined the prototype masks with the mask coefficients produced by the two subtasks linearly to generate the final result. After that, Wang et al. allocated categories to per-pixel within an instance based on the location and size of the instance, and transforms instance segmentation into a solvable single classification problem [25]. Besides, Xie et al. [26] presented two valid methods to cope with high-quality center samples and optimize the dense distance regression, respectively, which can obviously enhance the performance of instance segmentation and simplify the inference process. In addition, SOLOv2 [27] also followed the idea of segmenting objects by location (SOLO) [25] to learn the mask head of the instance segmenter dynamically to develop masks with higher accuracy.

Multi-level feature networks are widely used in instance segmentation tasks to improve the performance of algorithms [28]. The low-level features in instance segmentation networks obtain high resolution and abundant detail information but lack semantic information. Moreover, the high-level features contain abundant semantic information, but the resolution is low and the perception of details is weak. Therefore, the appropriate fusion of low-level features and high-level features can improve the network performance. Fully convolutional networks (FCN) [29] merged semantic features from deep and coarse layer with appearance features from shallow and fine layer through skip-connections to segment accurately and in detail. Correspondingly, Inside-outside net (ION) [30] adopted skip pooling to connect the feature maps of different convolutional layers to realize multi-level feature fusion. Subsequently, Ronneberger et al. [31] combined high-level features with low-level features by a contraction path for capturing context and a symmetric extension path for precise positioning. Inspired by the human visual pathway, Top-down modulation (TDM) networks [32] utilized a top-down modulation network to supplement the standard bottom-up feedforward network, which is accomplished by lateral connections. Similarly, FPN [15] exploited the inherent multi-scale and pyramid hierarchy of convolutional networks to build feature pyramids, and proposes a top-down structure with lateral connections to construct high-level semantic feature maps. Besides, single shot multibox detector (SSD) [33] integrated the predictions of multiple feature maps from different resolutions to deal with objects of various sizes naturally.

Many researchers have successfully integrated an attention mechanism into a convolutional neural network. Wang et al. [17] proposed a residual attention network embedded with bottom-up and top-down feedforward structures, and the deep residual attention network can be well trained by their proposed attention residual learning method. However, the residual attention network was quite computationally complex in comparison to other recent attention methods. Squeeze-and-Excitation Networks (SENet) [16] were devised by Hu et al., which can effectively increase the depth of the network and solve the over-fitting problem after increasing the number of layers in the deep network. It explicitly modelled the interdependence between channels, and designed Squeeze-and-Extraction module to improve the quality of neural network representation. Compared with SENet, Convolution Block Attention Module (CBAM) [20] proposed by Woo et al. generated the attention maps of input feature map from channel-wise and spatial-wise, which made the network focus more on the region of interest to boost the performance of the network. After definitely analyzing the advantages and disadvantages of SENet, Cao et al. proposed Global-Context Networks (GC-Net) [34], which can effectively model the global context and keep the network lightweight. More recently, Misra et al. [21] introduced a convolutional triplet attention module (CTAM) which aimed to catch cross-dimension interaction. It established inter-dimensional correlation through rotation operation and residual transformations, which can improve the representation of network while maintaining low computational cost.

3. Materials and Methods

In this section, we present the architecture of the proposed module and successively introduce feature extraction, lateral attention connections and feature enhancement in detail. Moreover, we also show the implementation details in the experiment.

3.1. The Framework

As shown in Figure 1, the workflow of the attention-based feature pyramid module (AFPM) consists of three parts: feature extraction, lateral attention connections and feature enhancement. In the bottom-up feature extraction structure, ResNet-50 [35] network is utilized to forward propagation, and the output of the last residual block in each stage is denoted as $\{C_1, C_2, C_3, C_4, C_5\}$, and $\{C_2, C_3, C_4, C_5\}$ is considered to participate in the subsequent calculation. In order to improve the expression of the region of interest, convolutional block attention module (CBAM) is added to the end of the first stage and the last stage. The output of the last residual block in each stage is denoted as $\{C_1', C_2, C_3, C_4, C_5'\}$ after adding convolutional block attention module. In the lateral attention connections, convolutional triplet attention module is included to focus on the cross-dimension dependencies and capture the abundant discriminating feature representation by catching the interaction between spatial dimension and channel dimension. The feature map after 3×3 convolution operation is defined as $\{P_2, P_3, P_4, P_5\}$. The top-down feature enhancement structure of feature pyramid network only enhances the strong semantic information of the high-level layer but lacks the reinforcement of the low-level features [14]. Based on this, we add branches to strengthen the location information in the network to achieve the purpose of increasing the whole feature hierarchy by propagating low-level features for instance segmentation.

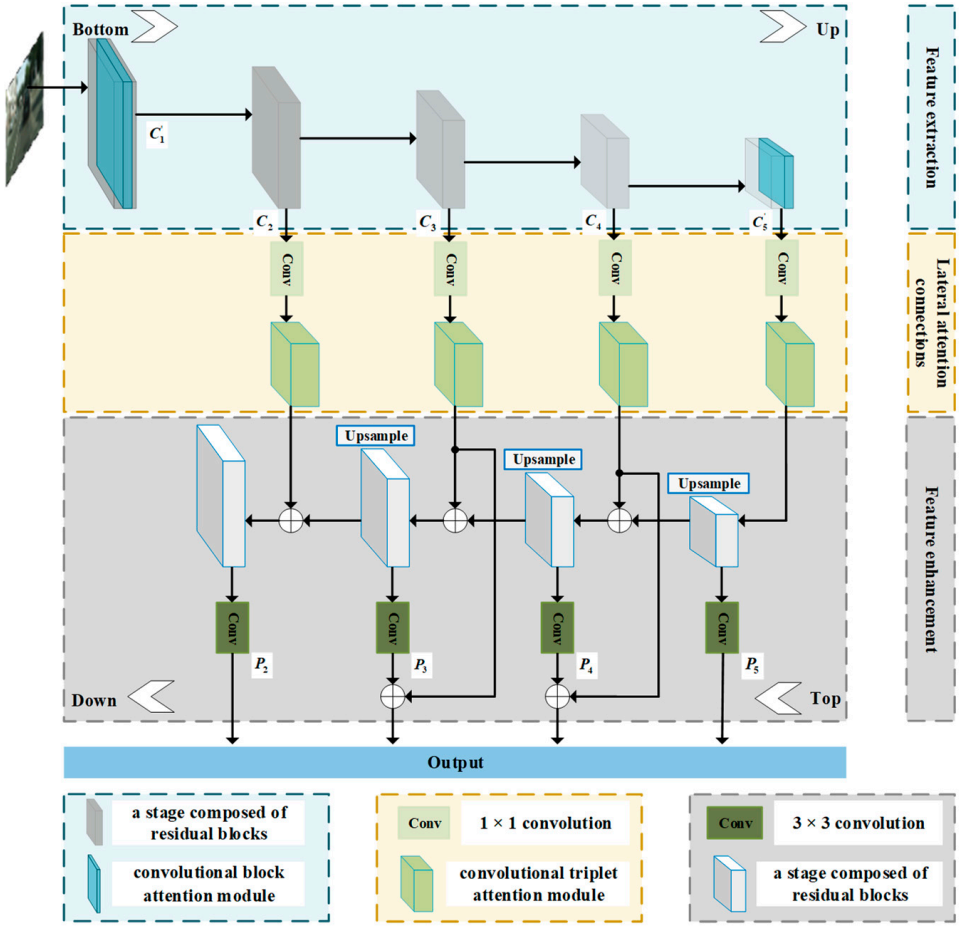

Figure 1. The architecture of the attention-based feature pyramid module (AFPM).

3.2. Feature Extraction

The features extracted by multi-level feature pyramid network are not all beneficial to the instance segmentation tasks, because some of the extracted features contain redundant and disturbing information. Therefore, in order to avoid the interference of redundant features and improve the effectiveness of instance segmentation network parameters, a convolutional block attention module (CBAM) is introduced into the feature extraction network. The CBAM as an important attention mechanism can focus on effective features and suppress unnecessary features by paying attention to what is meaningful and where is useful information in the input image. The size of the feature map generated by the first stage of ResNet-50 network before max pooling is reduced by 2 times compared with the input image. And $\{C_2, C_3, C_4, C_5\}$ have strides of $\{4, 8, 16, 32\}$ pixels relative to the input image. In ResNet-50 network, the feature map generated by the first stage before max pooling has the smallest size reduction with respect to the input image and contains the most abundant detail information such as contours, edges and textures in the five stages. Besides, the feature map generated by the last stage possesses the largest size reduction in relation to the input image and includes the richest abstract semantic information after many convolution and pooling operations in all stages. Therefore, the CBAM is included in front of the max pooling of the first stage and behind the last residual block of the last stage in ResNet-50 network to generate attention maps which focuses on instance related features in the low-level detail information and high-level semantic information. The detailed architecture is shown in Figure 2.

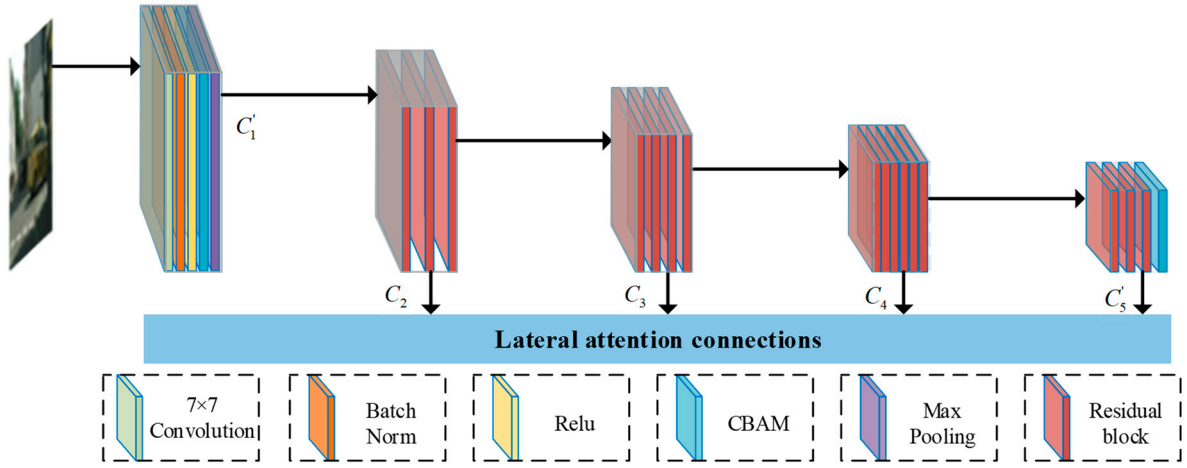

Figure 2. The architecture of feature extraction. We exploit ResNet-50 with convolutional block attention module (CBAM) to extract instance related features.

The convolutional block attention module (CBAM) is composed of a channel attention module and spatial attention module, which can capture the essential features along the channel dimension and spatial dimension. The architecture of CBAM is shown in Figure 3. In the channel attention module, spatial information of input features is obtained through the global average pooling (GAP) and global max pooling (GMP) operations, and two spatial context descriptors F^c_{avg} and F^c_{max} are generated. Then, F^c_{avg} and F^c_{max} generate channel attention map M_c through a shared network of multi-layer perceptron (MLP) with a hidden layer.

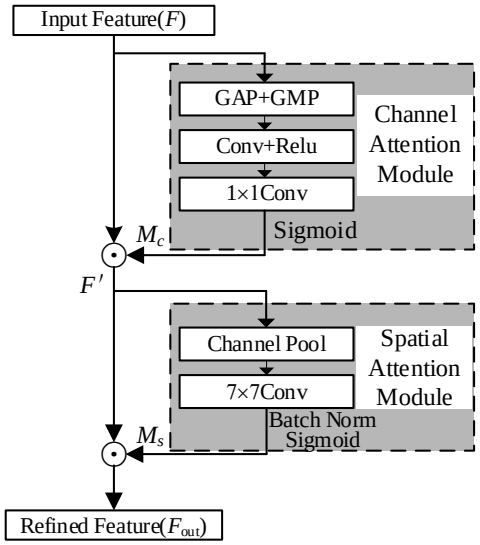

Figure 3. The architecture of CBAM [21].

When input tensor is $F \in \mathbb{R}^{C\times H\times W}$ ($\mathbb{R}^{Channel\times Height\times Width}$), the calculation formula of channel attention in CBAM is as follows:

$$M_c(F) = \sigma(W_1R(W_0g(F)) + W_1R(W_0\delta(F))) = \sigma(W_1R(W_0F^c_{avg}) + W_1R(W_0F^c_{max})), \quad (1)$$

where $M_c(F) \in \mathbb{R}^{C\times 1\times 1}$; $W_1 \in \mathbb{R}^{C\times C/r}$; $W_0 \in \mathbb{R}^{C/r\times C}$; r represents the reduction ratio in the bottleneck of the MLP and the value of r is set to 16; $\sigma(\cdot)$ represents the sigmoid activation function; $R(\cdot)$ represents the rectified linear unit; $g(\cdot)$ is the global average pooling (GAP) function; $\delta(\cdot)$ is the global max pooling (GMP) function.

In the spatial attention module, the pooling operations including average pooling and max pooling are performed along the channel axis, and two two-dimensional feature maps $F^{'s}_{avg}$ and $F^{'s}_{max}$ are generated. Then, they are concatenated and convoluted through the standard convolution layer to generate the spatial attention map M_s. The calculation formula of spatial attention in CBAM is as follows:

$$M_s(F') = \sigma(f_{7\times7}([g(F'); \sigma(F')])) = \sigma(f_{7\times7}(F^{'s}_{avg}; F^{'s}_{max})), \quad (2)$$

where $f_{7\times7}$ denotes a convolution operation in which the convolution kernel is 7×7; $M_s \in \mathbb{R}^{1\times H\times W}$; $F^{'s}_{avg} \in \mathbb{R}^{1\times H\times W}$; $F^{'s}_{max} \in \mathbb{R}^{1\times H\times W}$.

The overall process of calculating attention map can be summarized as:

$$\begin{cases} F' = M_c(F) \odot F \\ F_{out} = M_s(F') \odot F' \end{cases}, \quad (3)$$

where $\odot$ represents element-wise multiplication.

3.3. Lateral Attention Connections

Promotion in instance segmentation performance requires constructing high-level semantic feature maps at all levels to improve bottom-up and top-down pathways. The lateral connections act on the bottom-up feature extraction structure and provide feature maps with abundant information for the top-down feature enhancement structure, which play a significant connecting role in building high-level semantic mapping. To propagate effective and instance-related feature maps between feature extraction structure and feature enhancement structure, convolutional triplet attention module (CTAM) is added to the lateral connections to obtain useful semantic feature information by capture cross-dimension interaction. The architecture of CTAM is shown in Figure 4. CTAM consists of three branches, two of which focus on the dependencies between the spatial axis and channel axis, and the other branch is used to establish spatial attention. More specifically, the first branch captures the dependencies between the channel dimension and the height dimension of input tensor, the second branch captures the dependencies between the channel dimension and width dimension of input tensor, and the final branch captures the dependencies between the height dimension and width dimension of input tensor. The attention map with the same shape as the input tensor is obtained by averaging the outputs of the three branches of CTAM.

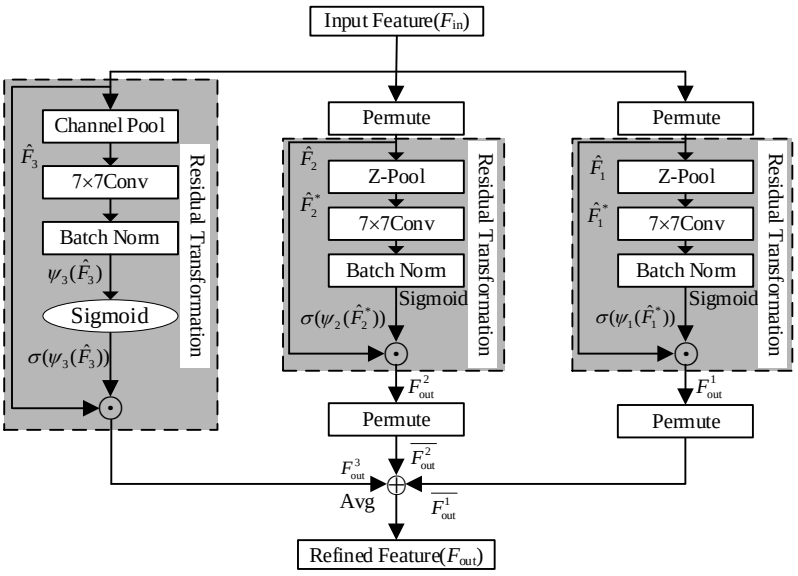

Figure 4. The architecture of the convolutional triplet attention module (CTAM) [21].

When input tensor is $F_{\text{in}} \in \mathbb{R}^{C\times H\times W}$, the calculation formula of CTAM is as follows:

$$F_{\text{out}} = \frac{1}{3}(\overline{\hat{F}_1\sigma(\psi_1(\hat{F}_1^*))} + \overline{\hat{F}_2\sigma(\psi_2(\hat{F}_2^*))} + F_{\text{in}}\sigma(\psi_3(\hat{F}_3))) = \frac{1}{3}(\overline{F_{\text{out}}^1} + \overline{F_{\text{out}}^2} + F_{\text{out}}^3), \quad (4)$$

where $F_{\text{out}} \in \mathbb{R}^{C\times H\times W}$; $\overline{F_{\text{out}}^1} \in \mathbb{R}^{C\times H\times W}$; $\overline{F_{\text{out}}^2} \in \mathbb{R}^{C\times H\times W}$; $F_{\text{out}}^3 \in \mathbb{R}^{C\times H\times W}$; $\psi(\cdot)$ is the standard two-dimensional convolutional operation; $\overline{F_{\text{out}}^1}$ represents a clockwise rotation of 90 degrees by F_{out}^1 along the height axis; $\overline{F_{\text{out}}^2}$ represents a clockwise rotation of 90 degrees by F_{out}^2 along the width axis; $\hat{F}_1$ is obtained by rotating input tensor 90 degrees anti-clockwise along the height axis; $\hat{F}_2$ is obtained by rotating input tensor 90 degrees anti-clockwise along the width axis; $\hat{F}_1^*$ represents the result of Z-pool operation of $\hat{F}_1$; $\hat{F}_2^*$ represents the result of Z-pool operation of $\hat{F}_2$; $\hat{F}_3$ represents the result of Z-pool operation of F_{in}.

The Z-pool operation is applied to decreasing the zeroth dimension to two through concatenating the features generated by average pooling and max pooling across that dimension. For example, the Z-Pool operation of an input tensor of shape (C × H × W) leads to a tensor of shape (2 × H × W). It can be formulated by the following equation:

$$Z_{-pool}(x) = [\sigma_{0d}(x), g_{0d}(x)], \quad (5)$$

where $0d$ is the 0th-dimension across which the max pooling and average pooling operations take place.

3.4. Feature Enhancement

The bottom-up feature mapping contains plentiful detail information, but has a low-level semantic. Zeiler et al. [36] indicate that high-layer neurons energetically respond to whole objects but other neurons are more possible to be stimulated by regional texture and patterns. The insightful point demonstrates the indispensability of augmenting a top-down pathway to propagate strongly semantical features in FPN [15]. Therefore, in order to boost the feature hierarchy of the whole pyramid, the top-down feature enhancement pathway is extended. Specifically, the top-down pathway produces higher resolution features by up-sampling the spatially coarser but semantically stronger feature map from a higher pyramid level. The features generated by each lateral connection are integrated into the top-down architecture for up-sampling and fusion at the next level. At last, feature maps are generated, which combines the low-level detail information and the high-level semantic information.

In addition, we further improve the localization capability of the overall feature hierarchy by adding two branches to propagate powerful responses of low-level patterns. Compared with the input image, the resolution of the feature map generated by the C_2 level is decreased by a quadruple amount, which includes rich detailed feature information but also contains certain interfering noise information. In order to reduce the introduction of disturbing information when enhancing the network localization capability, we chose the feature maps generated by C_3 and C_4 which performed a more effective feature extraction to add to the subsequent network. More explicitly, the feature maps generated by C_3 level with 512 channels and C_4 level with 1024 channels are enhanced in the lateral attention connections. A 3 × 3 convolution is acted on each merged map from top-down structure and generate the feature map of P_3 as well as P_4 with 256 channels. The feature maps generated by C_3 and C_4 levels after being acted by CTAM are integrated with P_3 and P_4 respectively to boost the overall feature hierarchy of the pyramid network.

3.5. Implementation Details

We took SOLOv2 [27] and FPN [15] as a base-network and applied the proposed module to it. The detailed software and hardware environment are shown in Table 1. Besides, at the beginning of training, the weight decay for SGD optimizer is set to 0.0001, and learning rate updated by step policy is set to 0.0025. The proposed architecture is trained with 145 epochs (203 k iterations) on 2048 × 1024 original training images, and

reduces the learning rate to 0.00025 at 6 epochs. The pre-trained models used in the experiments are publicly available, and the corresponding pre-trained models originate from ImageNet. We applied two images in one image batch for training and used one NVidia Titan V GPU.

Table 1. Hardware and software environment of experiments.

Items	Contents
Processor	Intel i9-10940x
Graphics card	NVIDIA Titan V
CPU memory	128 GB
Graphics memory	24 GB
Deep learning framework	Pytorch [37]
Deep learning toolbox	MMDetection [38]

4. Results and Discussion

In this section, we perform a set of experiments based on the Cityscapes dataset [39] in the same environment to explore the validity of the proposed module.

4.1. Dataset and Evaluation Metrics

The Cityscapes dataset records the urban street scenes data of 50 European cities in several seasons such as spring, summer and autumn. It is composed of 5000 finely annotated images with both semantic and instance information and 20,000 coarsely annotated images with only semantic information that are all at a resolution of 2048 × 1024 pixels. There are 19 object classes in the Cityscapes dataset, including 11 "stuff" classes and 8 instance-specific "thing" classes. Moreover, 5000 finely annotated images are divided into 2975 for the training set, 500 for the validation set and 1500 for the test set. The task of instance segmentation is to complete the detection and segmentation of 8 instance-specific classes. We have counted the respective object number of 8 instance classes in the Cityscapes dataset, as shown in Table 2.

Table 2. The number of each instance class in the Cityscapes dataset.

	Car	Rider	Bicycle	Person	Bus	Train	Motorcycle	Truck
Size (k)	26.9	1.8	3.7	19.9	0.4	0.2	0.7	0.5

In addition, as shown in Table 3, we make statistics on the scale of 8 instance classes according to the size of each class. In Table 3, the scale of each instance object is obtained by multiplying the width and height of the object and then taking the square. The mean, range and standard deviation are found by statistical analysis of scale values. The drastic change of object size and scale as well as the dynamic scene of object occlusion and aggregation increase its complexity, which makes it a challenging dataset for instance segmentation methods.

Table 3. The scale statistics results of each instance class in the Cityscapes dataset.

	Mean (pixel)	Range (pixel)	Standard Deviation (pixel)
car	101.4	1.7–850.7	104.5
rider	76.6	6.5–474.6	64.2
bicycle	73.8	3.9–538.1	61.5
person	63.1	2.0–669.8	61.5
bus	160.2	6.9–816.6	146.6
train	255.0	14.5–1045.8	230.3
motorcycle	93.0	4.5–462.0	76.6
truck	143.9	9.8–908.3	145.5

Besides, we employ average precision (AP) as the evaluation metric to assess the performance of the instance segmentation method. Explicitly, we follow the Cityscapes dataset instance segmentation evaluation criteria and adopt 10 overlaps in the range of 0.5 to 0.95 in steps of 0.05 to avoid bias for some specific values. As minor metric, we add AP_{50} for an overlap value of 50% as well as frames per second (FPS) to assess the performance of the network.

4.2. Instance Segmentation Results

In this subsection, we show the result of instance segmentation on the Cityscapes validation set. In the SOLOv2 network, the total loss function consists of category loss and mask loss. As shown in Figure 5, with the increase of network iterations, the loss function of each method gradually tended to be dynamically stable, and finally the instance segmentation models were obtained.

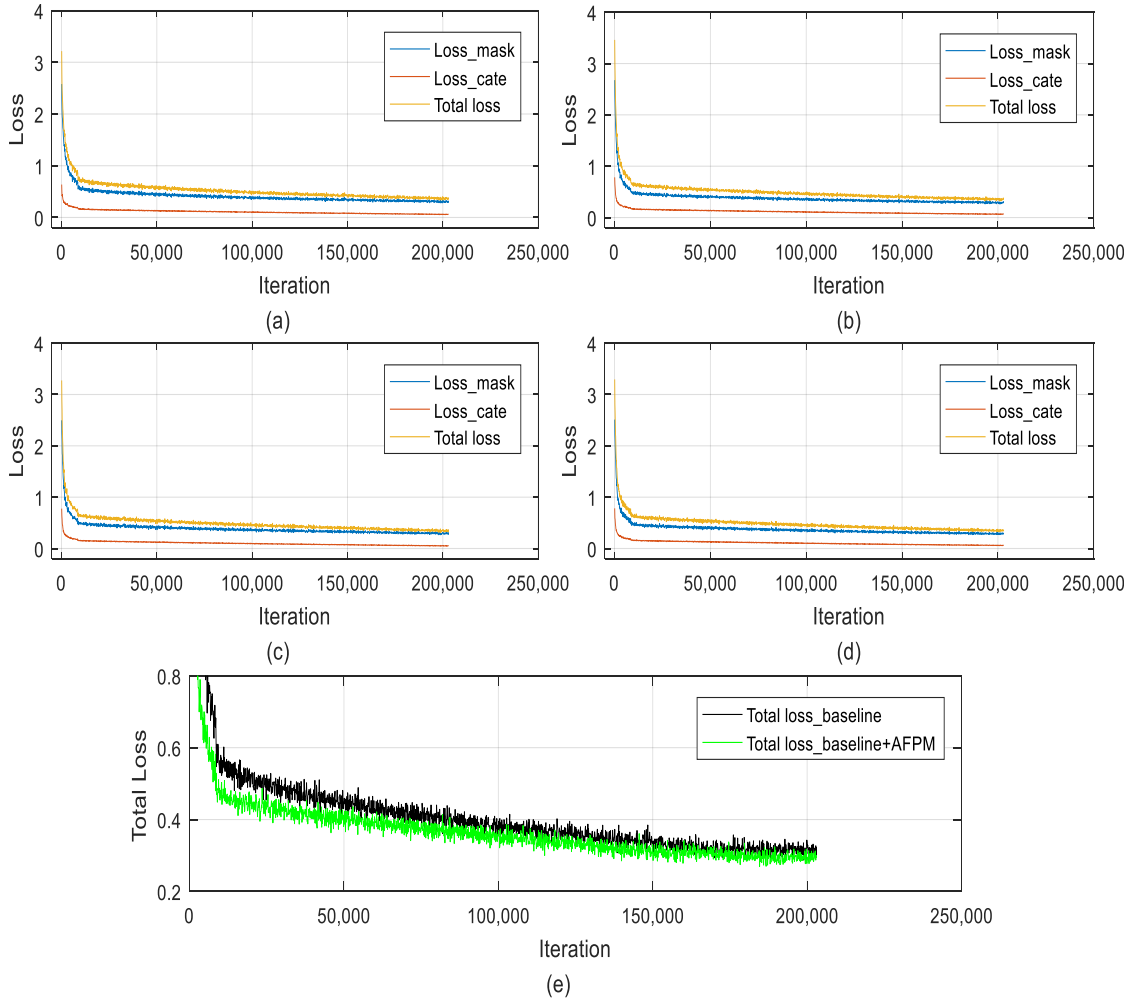

Figure 5. The loss function curve of different methods. (**a**) The loss curve of baseline; (**b**) the loss curve of baseline with CBAM; (**c**) the loss curve of baseline with CBAM and CTAM; (**d**) the loss curve of baseline with AFPM; (**e**) the total loss curve of baseline and baseline with AFPM.

As shown in Table 4, the instance segmentation method containing AFPM achieves the best performance. More specifically, AP gets 2.7% improvement and 9.5% improvement rate over baseline as well as AP_{50} gets 2.4% improvement and 4.6% improvement rate over baseline, which demonstrates that the proposed module can efficaciously assist the performance of instance segmentation. Although the proposed architecture makes the parameter increase slightly, which is 0.53 million, the average precision of the instance

segmentation algorithm improves obviously. We proved the effectiveness of each step by ablation experiments. Explicitly, we initially adopted SOLOv2 [28] with FPN [15] as backbone to conduct baseline experiments. The AP value and AP_{50} value of base-network was 28.3% and 51.7%, respectively. When we added convolutional triplet attention module (CTAM) to baseline, we got 28.7% AP value and 52.0% AP_{50} value. Then we added convolutional block attention module (CBAM) on the basic network and got 30.2% AP value as well as 52.5% AP_{50} value, which made the overall average precision was significantly promoted. Subsequently, we employed CTAM to the model which involved CBAM and the AP_{50} value increased effectively from 52.5% to 53.3% with negligible parameters. On this basis, we included branches that incorporate low-level features to construct the proposed framework and achieved 31.0% AP value as well as 54.1% AP_{50} value.

Table 4. Results of instance segmentation on the Cityscapes validation set.

Method	AP (%)	AP_{50} (%)	Parameters(M)
Baseline	28.3	51.7	32.79
Baseline + CTAM	28.7	52.0	32.79
Baseline + CBAM	30.2	52.5	33.32
Baseline + CBAM + CTAM	30.7	53.3	33.32
Baseline + AFPM	31.0	54.1	33.32

In Table 5, we report in detail the average precision of different methods on 8 instance class. By adding CBAM to the feature extraction structure, the redundant features extracted by the multi-level network are well filtered and the effectiveness of network parameters is improved, which makes the performance of instance segmentation method significantly boosted on all instance categories, especially in person class and rider class. Through combining lateral connections with CTAM, the high-level semantic features between bottom-up and top-down pathways are efficiently propagated. Among them, bus class and train class are enhanced by 0.9% and 0.7%, respectively. In the method in which AFPM is applied, the bus class with AP of 52.5% and car class with AP of 50.9% perform well. Furthermore, compared with the base-network, bicycle class and person class obtain higher improvement rate, which are 23.1% and 14.5%, respectively.

Table 5. Comparison of AP (%) for each instance class on the Cityscapes validation set.

Method	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
Baseline	25.5	21.5	48.4	26.3	48.1	24.4	16.0	16.0
Baseline + CTAM	25.7	22.0	49.0	26.7	49.1	24.8	16.4	16.0
Baseline + CBAM	28.9	24.6	49.3	26.4	51.2	25.2	16.5	19.5
Baseline + CBAM + CTAM	28.9	24.8	49.8	26.9	52.3	26.5	17.0	19.6
Baseline + AFPM	29.2	24.8	50.9	27.0	52.5	26.9	17.3	19.7

Moreover, as shown in Figure 6, we qualitatively display the segmentation results of ablation experiments. In Figure 6b, the basic network has the problems of missing and false detection in instance segmentation. For example, the network mistakenly segmented the car in the right edge of the first image when multiple objects overlap and failed to correctly identify the bicycle in the second image as well as the car in the right middle of the third image. As shown in Figure 6c, when CBAM and CTAM are integrated to the baseline, the problems mentioned above are powerfully improved and more object contours are detected and segmented. For instance, the distant bicycle in the first image and the car in the third image are accurately identified, and the outline of the truck in the second image is roughly segmented compared with the base-network. In Figure 6d, AFPM can assist the instance segmentation algorithm to detect different sizes and categories of objects more precisely, which aids the network to exactly segment two people on the left edge of the first

image, distant objects in the second image and the white truck similar to the background in the third image.

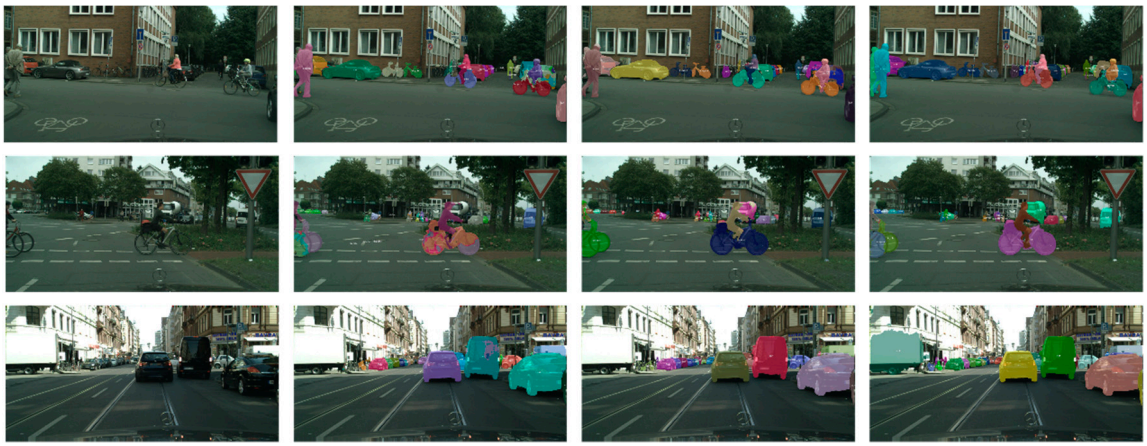

Figure 6. Comparison of instance segmentation results of different methods. (**a**) Original images; (**b**) the instance segmentation results of baseline; (**c**) the instance segmentation results of baseline with CBAM and CTAM; (**d**) the instance segmentation results of baseline with AFPM.

As shown in Figure 7, we also qualitatively compare the instance segmentation results of ground-truth and the proposed module. The proposed method which combined attention modules and a multi-level feature network can not only recognize the object correctly, but also detect and segment the instance object, which is contained in the input image but not labeled in the ground-truth. More intuitively, it can be seen from the first two rows of images in Figure 7 that the instance object in the input image can be accurately segmented by exploiting the proposed module. Furthermore, as shown in the last two rows of images in Figure 7, our method can also predict more correct objects that are not annotated in the ground-truth.

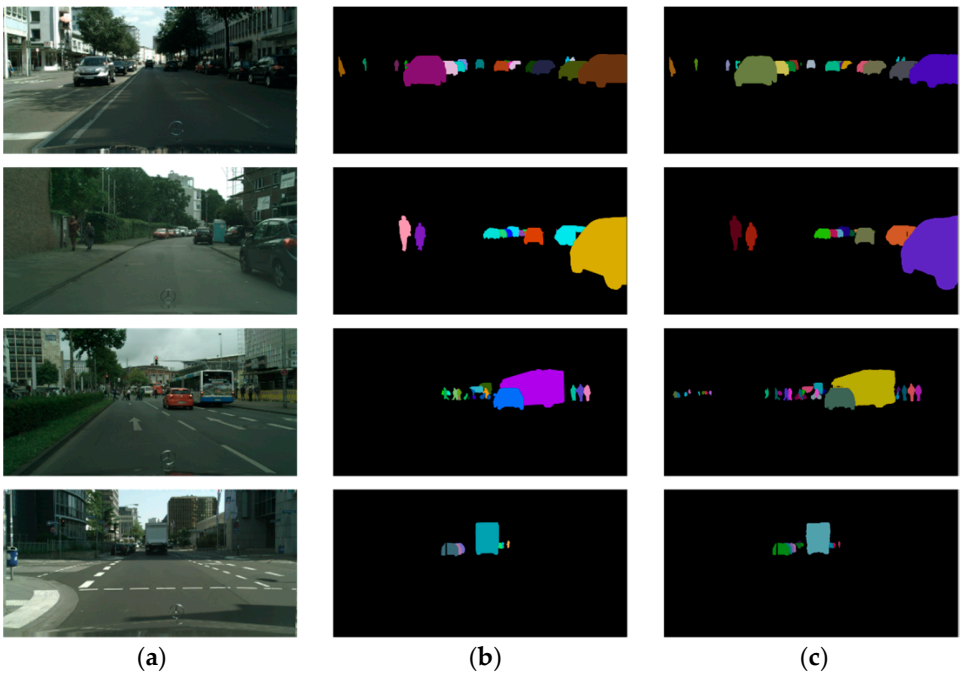

Figure 7. Comparison of instance segmentation results between ground-truth and the proposed method. (**a**) Original images; (**b**) the instance segmentation results of ground-truth; (**c**) the instance segmentation results of baseline with AFPM.

4.3. Comparison with Other Methods

Table 6 shows the evaluation results of the proposed approach and five high performing instance segmentation methods including Mask R-CNN on the Cityscapes test set in terms of AP and AP_{50} criteria. The proposed methodology outperforms other frameworks and gets the best results in terms of AP and AP_{50}, which proves the efficiency of the method in Table 6. More specifically, by applying AFPM, the AP of the instance segmentation algorithm is improved by 2.1% and AP_{50} is improved by 2.7% compared with that of the base-network. Therefore, AFPM effectively enhances the network performance. Besides, the accuracy of the proposed method not only exceeds the accuracy of the method trained on the fine set, but is also better than the accuracy of the method trained on both the fine set and coarse set.

Table 6. Comparison of instance segmentation results on the Cityscapes test set.

Method	Training Data	AP (%)	AP_{50} (%)
DIN [11]	fine + coarse	23.4	45.2
SGN [24]	fine + coarse	25.0	44.9
PolygonRNN++ [40]	fine	25.5	45.5
Mask R-CNN [6]	fine	26.2	49.9
GMIS [41]	fine + coarse	27.6	44.6
Baseline	fine	25.7	47.9
Baseline + AFPM	fine	27.8	50.6

In Table 7, we present in detail the average precision of each category of object for different algorithms on the Cityscapes test set. The AP of the instance segmentation method based on AFPM is higher than that of the basic network in all instance classes. Besides, due to the fact that the proposed module can extract the refined features of instances in different scales and sizes, the average precision of the instance segmentation method using AFPM surpass that of other methods in multiple categories. We perform well on the rider (23.8% vs. 23.7%), car (50.2% vs. 46.9%), bus (38.5% vs. 32.2%), train (24.7% vs. 18.6%) and bicycle (16.5% vs. 16.0%) than Mask R-CNN [6]. Moreover, compared with GMIS [41] which trains on both the fine and coarse sets, our method achieves well in person, car, bus and bicycle class. Especially on the bicycle class, the average precision exceeds them by 4.6%.

Table 7. Comparison of average precision (AP) (%) for each instance class on the Cityscapes test set.

Method	Person	Rider	Car	Truck	Bus	Train	Motorcycle	Bicycle
DIN [11]	20.9	18.4	31.7	22.8	31.1	31.0	19.6	11.7
SGN [24]	21.8	20.1	39.4	24.8	33.2	30.8	17.7	12.4
PolygonRNN++ [40]	29.4	21.8	48.3	21.1	32.3	23.7	13.6	13.6
Mask R-CNN [6]	30.5	23.7	46.9	22.8	32.2	18.6	19.1	16.0
GMIS [41]	29.3	24.1	42.7	25.4	37.2	32.9	17.6	11.9
Baseline	28.4	22.7	48.6	19.6	34.6	20.0	16.0	16.0
Baseline + AFPM	29.3	23.8	50.2	21.8	38.5	24.7	17.4	16.5

In addition, as shown in Table 8, we quantitatively display the results of FPS and AP of different instance segmentation methods when the input pixel is 2048×1024. Because CBAM increases the number of parameters in the instance segmentation network and CTAM has three branches, the total inference time is lightly expanded. In a word, we add attention mechanism and branches in turn to improve the average precision of the network from 25.7% to 27.8%, and at the same time slightly decrease the inference speed of the network, that is, from 7.3 to 5.5. Compared with other algorithms in Table 8, although our algorithm does not get the maximal inference speed, we achieve a relative balance between inference time and average precision.

Table 8. Comparison of frames per second (FPS) and AP on the Cityscapes test set [42].

Method	AP (%)	AP_{50} (%)	FPS (s^{-1})
Box2Pix [42]	13.1	27.2	10.9
BAIS [43]	17.4	36.7	<1
Discriminate Loss [44]	17.5	35.9	5
DWT [9]	19.4	35.3	<3
DIN [11]	20.0	38.3	<3
SGN [24]	25.0	44.9	0.6
Mask-RCNN [6]	26.2	49.9	2.2
Baseline	25.7	47.9	7.3
Baseline + CTAM	26.0	48.3	6.5
Baseline + CBAM	27.2	49.0	6.4
Baseline + CBAM + CTAM	27.5	49.8	5.5
Baseline + AFPM	27.8	50.6	5.5

5. Conclusions

In this work, we propose a unique AFPM for the task of instance segmentation, which utilizes the attention mechanism as well as multi-level feature pyramid network and consists of feature extraction, lateral attention connections and feature enhancement. By introducing a convolutional block attention module and a convolutional triplet attention module into the instance segmentation method, the features of objects can be extracted efficiently and discriminatively. Moreover, the branches used to enhance location information are added to the feature enhancement structure to strengthen the entire feature hierarchy with negligible computational overhead. The experimental results on the Cityscapes dataset demonstrate that the average precision of the instance segmentation method using AFPM is significantly improved compared with base-network and exceeds the performance of other excellent instance segmentation approaches including Mask R-CNN. In the future, we will explore more challenging datasets covering COCO dataset and extend our model to other computer vision tasks, such as semantic segmentation and object detection.

Author Contributions: Conceptualization, Y.S. and W.G.; methodology, Y.S.; validation, Y.S. and W.G.; formal analysis, Y.S. and Y.P.; investigation, Y.S. and S.P.; data curation, W.G. and S.P.; writing—original draft preparation, Y.S.; writing—review and editing, Y.S. and W.G.; visualization, Y.S. and T.Z.; project administration, S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by the National Natural Science Foundation of China (Grant No. 41774027, 41904022) and the Fundamental Research Funds for the Central Universities (2242020R40135).

Data Availability Statement: The data presented in this study are openly available in [https://www.cityscapes-dataset.com/] at [10.1109/CVPR.2016.350], reference number [39].

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

References

1. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Martinez-Gonzalez, P.; Garcia-Rodriguez, J. A survey on deep learning techniques for image and video semantic segmentation. *Appl. Soft Comput.* **2018**, *70*, 41–65. [CrossRef]
2. Chen, L.C.; Hermans, A.; Papandreou, G.; Schroff, F.; Wang, P.; Adam, H. MaskLab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4013–4022.
3. Hafiz, A.; Bhat, G. A survey on instance segmentation: State of the art. *Int. J. Multimed. Inf. Retr.* **2020**, *9*, 171–189. [CrossRef]
4. Chen, H.; Sun, K.; Tian, Z.; Shen, C.; Huang, Y.; Yan, Y. BlendMask: Top-down meets bottom-up for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 8573–8581.
5. Zagoruyko, S.; Lerer, A.; Lin, T.Y.; Pinheiro, P.O.; Gross, S.; Chintala, S.; Dollár, P. A multipath network for object detection. *arXiv* **2016**, arXiv:1604.02135.

6. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
7. Li, Y.; Qi, H.; Dai, J.; Ji, X.; Wei, Y. Fully Convolutional Instance-Aware Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4438–4446.
8. Dai, J.; He, K.; Sun, J. Instance-Aware Semantic Segmentation via Multi-task Network Cascades. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3150–3158.
9. Bai, M.; Urtasun, R. Deep Watershed Transform for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2858–2866.
10. Neven, D.; Brabandere, B.D.; Proesmans, M.; Gool, L.V. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8837–8845.
11. Arnab, A.; Torr, P.H. Pixelwise Instance Segmentation with a Dynamically Instantiated Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 879–888.
12. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y. YOLACT: Real-time instance segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9157–9166.
13. Sofiiuk, K.; Barinova, O.; Konushin, A. Adaptis: Adaptive instance selection network. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 7355–7363.
14. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
15. Lin, T.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S.J. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
16. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
17. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3156–3164.
18. 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, 18–22 June 2018; pp. 7794–7803.
19. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. A^2-nets: Double attention networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 352–361.
20. Woo, S.; Park, J.; Lee, J.; Kweon, I. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 3–19.
21. Misra, D.; Nalamada, T.; Arasanipalai, A.; Hou, Q. Rotate to Attend: Convolutional Triplet Attention Module. *arXiv* **2020**, arXiv:2010.03045.
22. 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.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
23. Liu, S.; Jia, J.; Fidler, S.; Urtasun, R. Sequential grouping networks for instance segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3496–3504.
24. Gao, N.; Shan, Y.; Wang, Y.; Zhao, X.; Yu, Y.; Yang, M.; Huang, K. Ssap: Single-shot instance segmentation with affinity pyramid. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 642–651.
25. Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; Li, L. SOLO: Segmenting objects by locations. *arXiv* **2019**, arXiv:1604.02135.
26. Xie, E.; Sun, P.; Song, X.; Wang, W.; Liu, X.; Liang, D.; Shen, C.; Luo, P. PolarMask: Single shot instance segmentation with polar representation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–20 June 2020; pp. 12193–12202.
27. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. SOLOv2: Dynamic and Fast Instance Segmentation. *arXiv* **2020**, arXiv:2003.10152.
28. Zhang, J.; Yan, Y.; Cheng, Z.; Wang, W. Lightweight Attention Pyramid Network for Object Detection and Instance Segmentation. *Appl. Sci.* **2020**, *10*, 883. [CrossRef]
29. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
30. Bell, S.; Zitnick, C.L.; Bala, K.; Girshick, R.B. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 2874–2883.
31. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
32. Shrivastava, A.; Sukthankar, R.; Malik, J.; Gupta, A. Beyond skip connections: Top-down modulation for object detection. *arXiv* **2016**, arXiv:1612.06851.
33. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.

34. 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 International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. Zeiler, D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 818–833.
37. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 8026–8037.
38. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* **2019**, arXiv:1906.07155.
39. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
40. Acuna, D.; Ling, H.; Kar, A.; Fidler, S. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 859–868.
41. Liu, Y.; Yang, S.; Li, B.; Zhou, W.; Xu, J.; Li, H.; Lu, Y. Affinity derivation and graph merge for instance segmentation. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 686–703.
42. Uhrig, J.; Rehder, E.; Frohlich, B.; Franke, U.; Brox, T. Box2pix: Single-shot instance segmentation by assigning pixels to object boxes. In Proceedings of the IEEE Intelligent Vehicles Symposium, SuZhou, China, 26–29 June 2018; pp. 292–299.
43. Hayder, Z.; He, X.; Salzmann, M. Boundary-aware instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5696–5704.
44. De Brabandere, B.; Neven, D.; van Gool, L. Semantic instance segmentation with a discriminative loss function. *arXiv* **2017**, arXiv:1708.02551.