ITC 3/53
Information Technology and Control
Vol. 53 / No. 3/ 2024
pp. 675-694
DOI 10.5755/j01.itc.53.3.35588

AMF-SparseInst: Attention-guided Multi-Scale Feature Fusion Network Based on SparseInst

Received 2023/11/11 | Accepted after revision 2024/06/25

HOW TO CITE: Chen, Y., Wan, L., Li, S., Liao, L. (2024) AMF-SparseInst: Attention-guided Multi-Scale Feature Fusion Network Based on SparseInst. *Information Technology and Control*, 53(3), 675-694. https://doi.org/10.5755/j01.itc.53.3.35588

AMF-SparseInst: Attention-guided Multi-Scale Feature Fusion Network Based on SparseInst

Yiyi Chen, Liang Wan

State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, 550025, Guiyang, Guizhou, China; e-mails: 2686997250@qq.com, lwan@gzu.edu.cn

Shusheng Li

College of Computer Science and Engineering, Northeastern University, Shenyang 110189, China; e-mail: 2310731@stu.neu.edu.cn

Liang Liao

State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, 550025, Guiyang, Guizhou, China; e-mail: gs.lliao21@gzu.edu.cn

Corresponding author: lwan@gzu.edu.cn

SparseInst is an efficient real-time instance segmentation model, but redundant background features will be generated in multi-scale feature fusion, which will cause feature loss for small objects with low resolution and similar pixels to the background. To address the issue, we propose a real-time instance segmentation model named AMF-SparseInst (Attention-guided Multi-Scale Feature SparseInst), which can effectively highlight the most critical features of small objects from cluttered backgrounds. Firstly, we design a pyramid pooling module (called SimAM-ASPP), which consists of some depthwise separable convolutions with three different expansion rates and a 3D attention mechanism (called SimAM). It can capture contextual information from different receptive fields and focus on small object features. Secondly, we designed the Lite -BiFPN module to associate and integrate different levels of semantic information from top to bottom and from bottom to bottom. Finally, we propose a feature enhancement module FEM, which uses N3 and N5 respectively to reweight fusion features in spatial and channel dimensions to enhance the effective information of multi-scale fusion features. Experimental results demonstrate the superiority of AMF-SparseInst over the benchmark on COCO 2017 test-dev. Specifically, the AMF-SparseInst makes a 3.6% improvement in overall segmentation accuracy, while increasing speed by 2.5 FPS. Moreover, it achieves a favorable balance between accuracy and speed on the Cityscapes validation set.

KEYWORDS: AMF-SparseInst, multi-scale feature, real-time instance segmentation, SimAM-ASPP.

1. Introduction

Instance segmentation aims at classifying, detecting and segmenting objects of interest in an image, and thus is widely used in various domains. Recent works [1-2, 27, 10, 30-31] have produced impressive results on large-scale benchmarking, such as COCO [21]. However, real-time and efficient instance segmentation models still pose challenges and urgency, especially in the field of autonomous driving and robotics.

As the most representative two-stage method in recent years, Mask R-CNN [10] is the first to use a detector to recognize the object category and its region in the image, then performs pixel-level semantic segmentation within specific region. While those methods [10, 17, 26] yield high quality masks, it faces challenges in complex scenes with multi-scale objects. Because the network needs to generate a large number of region proposals in the early stage, and the Non-Maximum Suppression (NMS) in the later stage becomes time-consuming. Additionally, the algorithms are not easily deployable to edge/embedded devices due to the use of ROI-Align on region features. Therefore, researchers have explored one-stage approaches, like YOLACT [2], which parallelizes the detection and segmentation tasks and eliminate region proposals and post-processing techniques. These approaches [2, 27, 6] enable faster inferring and offers flexibility in segmentation mask shape and size. However, while YOLACT pays attention to the speed of reasoning, the segmentation accuracy is not outstanding. Since the mask is obtained by clipping after synthesis, it lacks the ability to suppress noise outside the proposals. Consequently, false detection and missing detection of the mask may occur, especially if the anchor boxes are localized or when similar instances with large scale differences are far apart.

Real-time instance segmentation is critical for many application scenarios, such as autonomous driving, robot vision, and monitoring systems, which need to quickly and accurately identify and segment different objects in a scene. To overcome the limitations of slow inference and low accuracy in multi-scale mask segmentation, this paper presents a novel one-stage and anchor-free approach for instance segmentation. SparseInst [4] is one of the latest methods of real-time instance segmentation, it ranks first in the CVPR list in 2022. SparseInst provides a decentralized set of features, such as activating new object representation maps that effectively highlight the information area of each rendered object. By employing the bipartite-matching algorithm [45], SparseInst achieves one-to-one prediction of the object mask, eliminating the need for post-processing NMS and striking a balance between accuracy and real-time performance. Most importantly, SparseInst serves as a baseline architecture that gives us a powerful and efficient starting point to further investigate and optimize the real-time instance segmentation task. Therefore, for this contribution, we adopted SparseInst as the baseline architecture, but SparseInst still has two problems with instance segmentation. 1) SparseInst has poor segmentation quality and low efficiency for images containing multi-scale objects. 2) SparseInst is usually disturbed by complex background, which generates a large amount of redundant background information and overwhelms small object information, resulting in a large number of missed detections. To solve the above problems, we propose a model named Attention-guided Multi-Scale Feature SparseInst, which combines with the module AMF. The focus of our work is to enhance the model's perception of effective features in multi-scale fusion features, so that it can improve the robustness of instance segmentation for multi-scale objects and small objects. To do this, we use attention modules on the network backbone and the output of the feature pyramid network at multiple scales. In addition, we use technology optimization capabilities to explore the advantages of the connectivity of the feature pyramid network. The main contributions of our work can be summarized as follows:

1 We design the SimAM-ASPP (Atrous Spatial Pyramid Pooling module with SimAM attention), and introduce the atrous depthwise separable convolution and SimAM attention mechanism to obtain the correlation features of different receptive fields, while reducing the number of model parameters and enhancing the effectiveness of small object features.

2 We propose the FEM (Feature Enhancement Module) to improve channel attention and purify the fused multi-scale features from the spatial and channel dimensions to enhance the extraction of effective information.

3 Building upon SimAM-ASPP and FEM, we enhance the FPN connection and design a network named AMF-SparseInst. Thorough experiments on COCO 2017 and Cityscapes dataset demonstrate that AMF-SparseInst exhibits improved segmentation performance for small objects.

2. Related Works

2.1. Overview of Deep Learning

Deep learning is the core of machine learning and artificial intelligence, which processes data using deep neural networks by simulating the human brain. Without manual feature engineering, deep learning can automatically extract features from large data sets for end-to-end processing. In the field of computer vision, deep learning is widely used, including data dimensionality reduction, pattern recognition, etc., and has achieved remarkable results. Zheng's many research achievements [39-43] in the field of deep learning have provided useful enlightenment and reference for computer vision tasks such as instance segmentation. The training process involves data preprocessing, model construction, forward propagation prediction, backpropagation calculation gradient, parameter updating, iterative optimization of the model to improve the prediction accuracy.

2.2. Real-Time Instance Segmentation

Although extensive research has been conducted on real-time object detection and semantic segmentation, there remains a scarcity of studies focusing on real-time instance segmentation. The challenge lies in the complexity of the instance segmentation task, which involves localizing the position of an object in an image, segmenting all salient objects at the pixel level, and distinguishing different instances of the same class. Currently, real-time methods typically employ one-stage models that detect and segment branches in parallel, performing localization, classification, and segmentation simultaneously in a single phase. In general, we call current state-of-the-art models [8] when they can be reasonably pioneering in the field of instance segmentation, obtain the highest AP values on several public datasets, such as COCO, and are not only highly accurate but also sufficiently efficient in several segmentation tasks. In recent years, with the development of deep learning research, the state-of-the-art models (e.g., YOLACT [2], CondInst [27], QueryInst [6]) that perform on the COCO dataset tend to be faster, highly applicable and easy to deploy on edge/embedded devices than the two-stage models (e.g., Mask R-CNN [10], MS R-CNN [17]).

However, the anchor-free detector [6] that use the center pixel to represent the object and its bounding box can be limited. Additionally, exploring various methods for object contour representation (e.g., Polarmask [33], ESE-Seg [34]) has revealed deficiencies in segmenting objects with hollow regions or multiple parts, making it challenging to distinguish between different object instances. Table 1 lists the advantages and disadvantages of different instance segmentation algorithms.

Several real-time instance segmentation algorithms, including YOLACT [2], YOLACT++ [1], CondInst [27], SOLO [30], and SOLOv2 [31], have shown progress in generating instance masks using different techniques. YOLACT [2] and YOLACT++ [1] assemble mask coefficients and prototype masks to generate instance masks. CondInst [27] employs dynamic convolution and absolute positional coordinates to generate instance-aware convolution kernels, while SOLO [30] and SOLOv2 [31] use absolute positional coordinates to define instance categories.

Despite the advancements, there still exists a considerable segmentation accuracy gap when compared to Mask R-CNN [10]. To address this, SparseInst [4] introduces a novel approach using a sparse set of instance activation mappings, representing objects based on salient regions and features. SparseInst predicts objects on a one-to-one basis using a binary matching algorithm [45], which eliminates the need for NMS in post-processing. As a result, this method significantly improves segmentation accuracy and accelerates inference speed.

Table 1
Comparisons the advantages and limitations of different instance segmentation algorithms

method	year	technology	advantages	limitations
Mask R-CNN [10]	2017	Combination of Faster RCNN, ResNet-FPN, ROI Align, the mask branch of FCN	Completes two tasks of target detection and segmentation in parallel	Dependent on target detection results
MS R-CNN [17]	2019	Modify the evaluation criteria of mask	Evaluate the mask result reasonably	The network is huge and time-consuming to train
PANet [29]	2019	Bottom-up feature path, adaptive fusion ROI pooling	Enhance information fusion and feature utilization between different scales	The correlation between the proposal and the different levels of features is not strong
YOLACT [2]/ YOLACT++ [1]	2019	Fusion prototype drawing and inspection box	Real-time instance segmentation	The precision is lower than that of the two-stage methods
Fcos [26]	2019	Anchor-free detector that use the center pixel to represent the object	Simple structure, easy to adapt to other tasks	The speed is slower than the one-stage detection algorithm
SOLO [30] / SOLO V2 [31]	2019	Extract the point features of the target, divide the grid, Matrix NMS	Fast speed, high accuracy	Long training time
Polarmask [33]	2020	Polar coordinate modeling mask	Novel method	Edge information ambiguity
CondInst [27]	2020	The dynamic network output mask directly	High speed and high accuracy	Large objects lack segmentation details
CenterMask [19]	2020	Decompose into local and global masks	Balancing speed and accuracy	There's room for improvement
QueryInst [6]	2021	$Dynconv_t^{mask}$, integrated object queries and Hungarian matching, transformer Multi-head attention module	High accuracy	Low speed on COCO and Cityscapes
CenterPoly [23]	2021	Polygon regression head and the vertex selection strategy	Fast and lightweight, can run at real-time speed	Lack of obtained mask details, polygon vertex output is not flexible
SparseInst [4]	2022	Novel object representation by instance activation maps, bipartite matching	Better balance accuracy and speed	Poor performance on small objects

2.3. Spatial Pyramid Pooling

Spatial Pyramid Pooling (SPP) [12] converts input of different sizes into output of fixed sizes to avoid information loss and distortion of position information. DeepLab series [3] use multiple parallel atrous convolutions with different rates to capture rich contextual information. DSNet [7] proposes a introduces a Context-Guided Dynamic Sampling (CGDS) module that adaptively captures useful segmentation information in the sampling space by obtaining effective representations of rich shape and scale information. ICNet [37] adopts a multi-resolution approach, dividing the image into high, medium, and low-resolution layers. It generates coarse segmentation results from the low-resolution images through a semantic segmentation network. Then, it combines cascaded label guidance and strategies to incorporate high-resolution features and iteratively refine the coarse segmentation results generated before. PSPNet [38] utilizes a Pyramid Pooling Module (PPM) to aggregate contextual information from different regions, enhancing its ability to capture global information. APCNet [9] designs an adaptive context module that utilizes GLA to compute context vectors at each local location in order to aggregate contextual information.

SpineNet-Seg [24] is the first module to capture contextual information via the NAS (Neural Architecture Search), scale-Perarm web semantic segmentation discovery. CFPNet [22] uses Channel-wise Feature Pyramid (CFP) to jointly extract feature mappings of mutiple scales and reduces the number of parameters. Furthermore, MobileNets [14] first proposed separable convolution with fewer parameters and fewer computations. ShuffleNet [36] adopts depthwise separable convolution to achieve lightweight feature extraction. MobileNetV3 [15] makes full use of depthwise separable convolution, possessing a highly optimized structure and efficient feature extraction capability, making it an ideal choice for real-time image recognition and segmentation on computationally constrained devices. The depthwise separable convolution decomposes the ordinary convolution computation into depthwise and pointwise convolution processes, is employed in our approach to reduce the number of parameters in the pyramid pooling module.

2.4. Multi-Scale Feature Fusion

In instance segmentation tasks, detecting objects with varying scales requires robust performance with the use of multiscale features [28]. FPN (Feature Pyramid Network) [20] is a classical approach for multiscale feature fusion, combining shallow detailed features with deep semantic features to improve object detection across different scales. However, the long path between deep and shallow features makes accessing accurate localization information challenging. To address this, PANet [29] introduces a bottom-up path aggregation network on top of FPN, shortening the information access paths and leveraging shallow accurate localization information to enhance the feature pyramid. BiFPN [25] uses a composite scaling method to repeat the PAFPN layers, eliminating transversal connections that contribute less to the fused features, and adding a residual edge in the transversal connection from P4 to P6 to get the more advanced fusion features. Therefore, in this paper, a lightweight version of BiFPN is utilized to enhance the robustness of multi-scale fusion features.

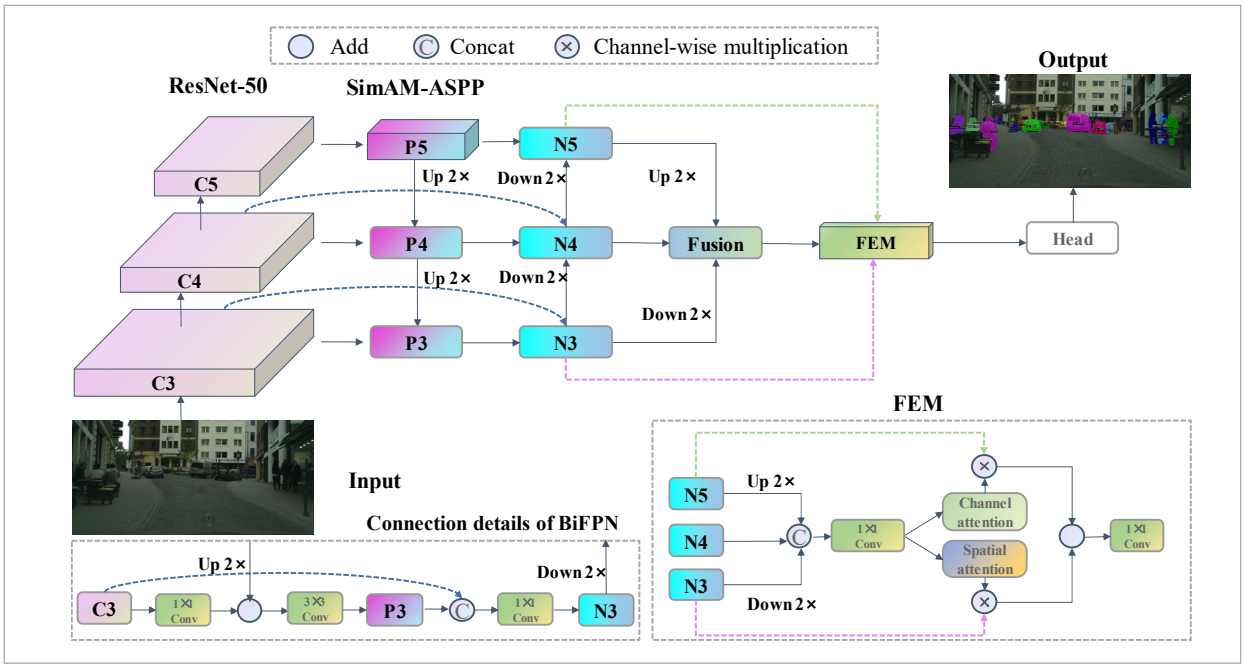
2.5. Attention Mechanisms

In computer vision tasks, the attention mechanism enables the network to focus on the most relevant features. Over time, researchers have continuously designed various attention modules in convolutional neural networks. SE [16] was one of the pioneering methods that recalibrated original features in the channel dimension by capturing contextual cues from a global field of view. However, SE's two fully connected layers may cause channel information loss due to channel reduction. CenterMask [19] proposed eSE (effective Squeeze-Excitation), using only one fully-connected layer to effectively avoid channel information loss. CBAM [32] achieves a reweighting of channel and spatial dimensions to combine one-dimensional attention with two-dimensional attention. However, it still fails to emulate the attention mechanism of the human brain. To overcome this limitation, SimAM [35] proposes a uniformly weighted attention module, assigning unique weights to each information-rich neuron based on neuroscience theory. It achieved 3D attention without parameters. Inspired by the above, this paper refines features using an improved CBAM module and introduces SimAM into pyramid pooling networks to extract salient features of small objects.

3. The Proposed Method

In this section, we introduce AMF-SparseInst in three subsections. Firstly, the SimAM-ASPP core component ASPP is used to enhance the capture of important features of multi-scale objects and small objects. Secondly, FEM refines the multi-scale fusion features of the lightweight version of BiFPN output. The overall structure of AMF-SparseInst is shown in Figure 1. Specifically, we first input the image into the ResNet-5-d-DCN backbone, which is both efficient and powerful, enhancing the recognition of targets with complex shapes and poses. Secondly, the features of three scales from the backbone are fed into the SimAM-ASPP module. The SimAM-ASPP module is responsible for capturing contextual and global information of multi-scale targets, with a particular focus on the features of small targets. To strengthen the connections between features of different scales, we borrow the idea from BiFPN and fuse features through residual connections without increasing costs, such as connecting C3 to N3 and C4 to N4. The specific details are shown in the dashed box on the lower left of the figure. To ensure feature align-

Figure 1
The overall architecture of AMF-SparseInst



ment, we perform 2x upsampling on C5 and N5, and 2x downsampling on C3 and N3, aligning them with C4 and N4. Subsequently, addressing the multi-scale and semantic differences in the fused features, we designed the Feature Enhancement Module (FEM), which reassesses the importance of features, enhances key features, and filters out interference information, thus improving the quality of the fused features. Finally, the enhanced features are fed into the SparseInst head structure, and the final segmentation result is obtained after 4x upsampling.

3.1. Spatial Pyramid Pooling Module

In instance segmentation, predicting objects of multiple scales can result in information loss and insufficient contextual understanding when fusing multi-scale features. Thus, we draw inspiration from ASPP and introduce SimAM-ASPP to capture contextual and global information of multi-scale objects. The architecture and components of SimAM-ASPP are illustrated in Figure 2. Moreover, Figure 3 provides a detailed comparison between ASPP and SimAM-ASPP, highlighting the specific differences between the two methods. By leveraging atrous convolution, SimAM-ASPP effectively expands the receptive field while preserving the resolution of the feature map. This empowers the model to better comprehend multi-scale instances, enhancing its performance in instance segmentation tasks. For the C5 output from the backbone network, we first input them into the ASPP consisting of ordinary convolution and atrous depthwise separable convolution with different dilation rates to aggregate multi-scale contextual information, and then use SimAM attention to emphasize the essential features related to small and multi-scale objects. Unlike ASPP in DeepLabv3+ [3], we use depthwise convolution to reduce the number of model parameters, effectively alleviating the pressure on the number of parameters due to the increase in convolution kernel size. Specifically, we use an ordinary 3×3 convolution, three atrous depthwise convolutions with different expansion rates (6, 18, and 24) and convolution kernel sizes of 3×3 to differentially expand the receptive fields of the C5 features to capture multi-scale contextual information, and a global average pooling operation to capture global information. Next, the feature maps obtained from the previous steps are stitched together in the channel dimension using the Concatenate operation, resulting in the feature map. Subsequently, these feature maps

Figure 2
SimAM and Atrous Spatial Pyramid Pooling(SimAM-ASPP)

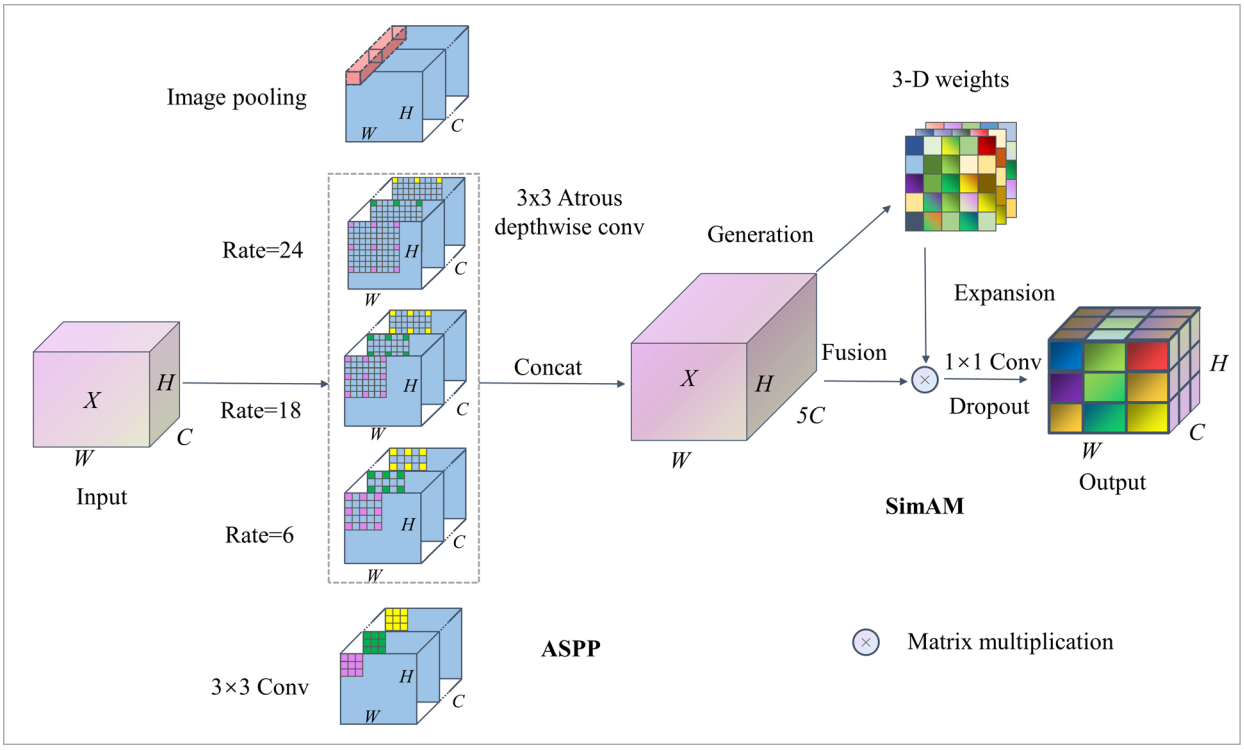
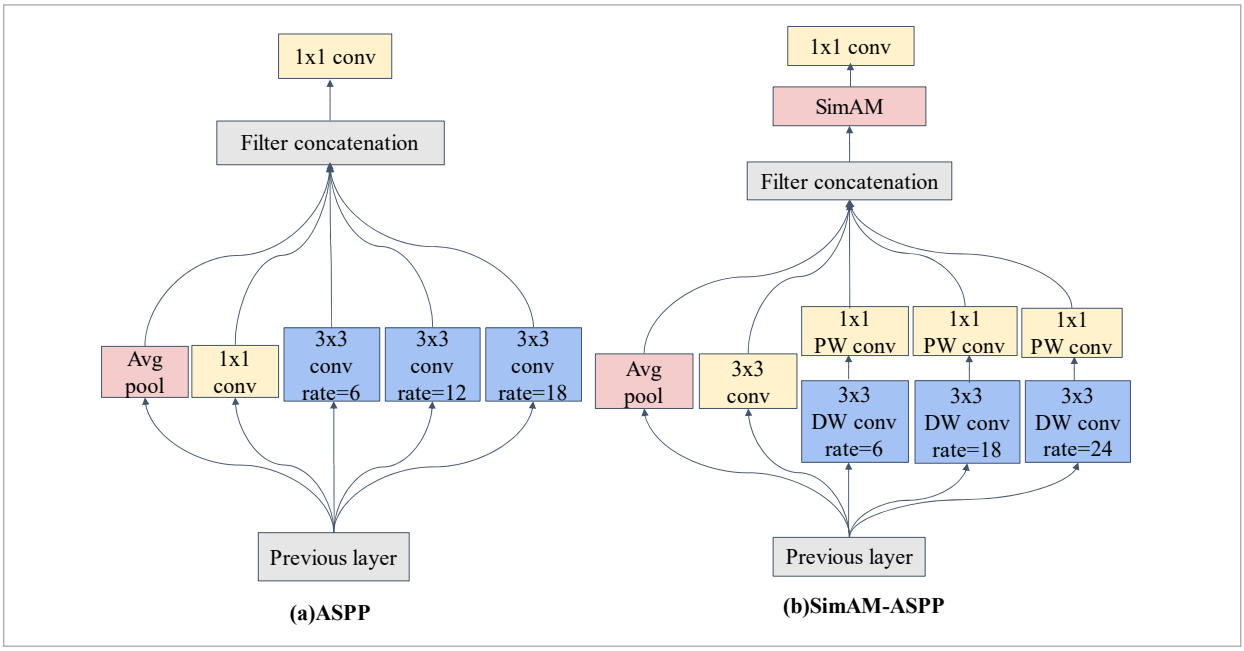


Figure 3
ASPP vs. SimAM-ASPP



undergo a dropout operation to focus on potentially useful features related to the multiscale objects in the 3D dimension. Finally, the information between channels is reduced and interacted using 1×1 convolution to obtain the output feature map.

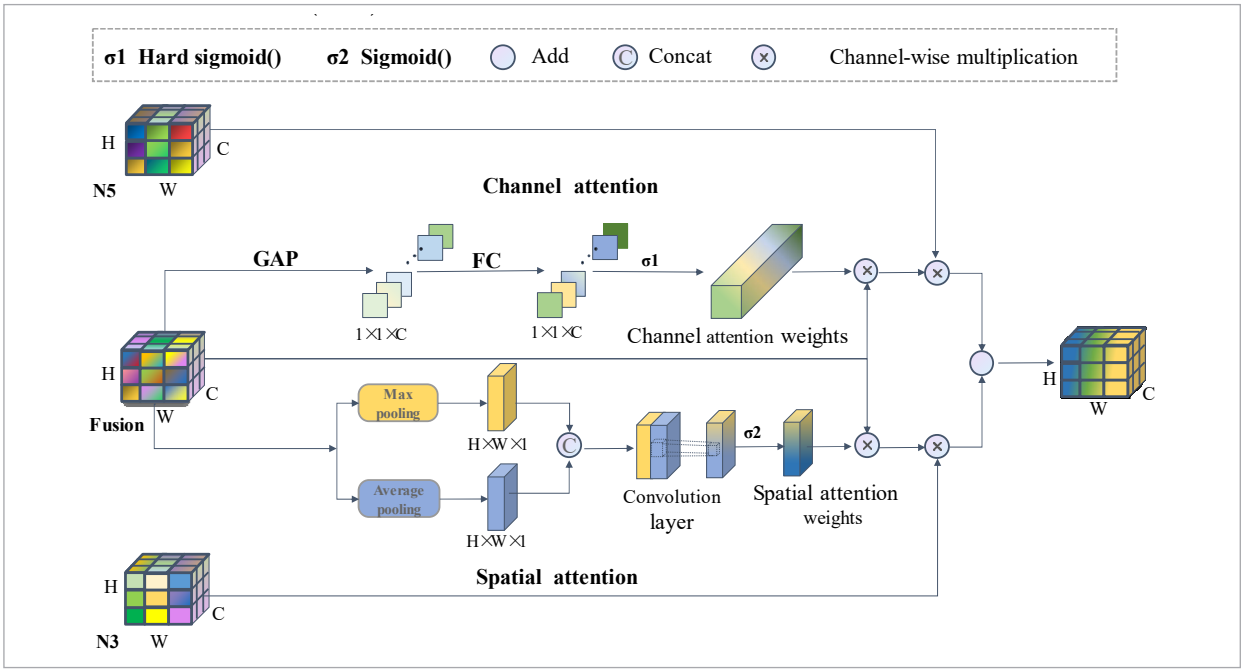
3.2. Feature Enhancement Module

BiFPN [25] fuses feature of different scales in its output feature map, but features of different scales have different semantic differences, directly fusing them will bring redundant information and interfere with the model's performance. Therefore, we design the Feature Enhancement Module (FEM) module to reassess the importance of the multi-scale fused features, further enhance key features, filter the interference information generated by the fusion, and alleviate the dispersion problem of the small object features. The details are shown in Figure 4, FEM consists of two parallel branches, which we call the channel purification and spatial purification modules. These branches generate adaptive weights in the channel and spatial dimensions, respectively, guiding the features to learn in a more critical direction. SE [16] a representative channel-attention method commonly used in CNN architectures, explicitly models the interdependencies between feature mapping channels to enhance their representation. However, the SE module has a limitation: the loss of channel information due to dimensionality reduction. To avoid excessive model complexity, the two fully-connected layers of the SE module need to reduce the channel dimensions, leading to the loss of channel information. CenterMask [19] demonstrated that channel information can be preserved by using only one fully connected layer with C channels. Therefore, in the channel purification part, we adopt the eSE module and omit the first FC layer that compresses the channels. Specifically, the input features $X_{fusion} \in \mathbb{R}^{H\times W\times C}$ are compressed in the spatial dimension to aggregate the spatial information that represents the global features of the image. ESE process can be defined as Equation (1).

$$A_{ca}\left(X_{fusion}\right)=\sigma_1\left(W_c\left(F_{gap}\left(X_{fusion}\right)\right)\right), \tag{1}$$

where $X_{fusion} \in \mathbb{R}^{H\times W\times C}$ is the output of BiFPN module and is multi-scale fusion feature that computed from splicing operation and 1×1 convolution. And

Figure 4
Feature Enhancement Module (FEM)



$F_{gap}(X) = \frac{1}{WH}\sum_{i,j=1}^{W,H} X_{i,j}$ is channel-wise global average pooling operation, $W_C \in \mathbb{R}^{C\times1\times1}$ is weight of fully-connected layer, σ_1 is the Hardsigmoid activation function. Subsequently, $A_{ca}(X_{fusion}) \in \mathbb{R}^{1\times1\times C}$ is used as the weight coefficient of the channel attention, applied X_{fusion} to enhance its features representation via a matrix multiplication operation, as shown in Equation (2).

$$X_{ca} = A_{ca}(X_{fusion}) \otimes X_{fusion}. \quad (2)$$

Furthermore, to enhance the model's ability to focus on local details and highlight crucial spatial regions, we introduce a spatial purification branch in parallel. Specifically, the fused feature maps are subjected to maximum pooling and global average pooling to generate feature descriptors, respectively, as shown in Equation (3).

$$C(X) = Concat\left(Avg(X), Max(X)\right). \quad (3)$$

The two are spliced together to generate spatial attention maps using 5×5 convolution.

$$A_{sa}(X_{fusion}) = \sigma_2\left(Conv^{5\times5}\left(C(X_{fusion})\right)\right). \quad (4)$$

This attention map is then normalized by a sigmoid activation function to obtain the spatial attention weight $A_{ca}(X_{fusion}) \in \mathbb{R}^{1\times1\times C}$ vector. Finally, this weight vector X_{fusion} is associated with a matrix multiply $\otimes$ operation to get a spatially weighted feature map. This ensures that important spatial locations are amplified while unimportant regions are suppressed. The specific details are shown in Equation (5),

$$X_{sa} = A_{sa}(X_{fusion}) \otimes X_{fusion}. \quad (5)$$

Among them, σ_2 is the sigmoid activation function. Finally, we perform softmax operations on N5 and N3 to obtain the channel attention weighted value W_5 and spatial attention weighted value W_3, where $W_3 + W_5 = 1$. Subsequently, matrix products are conducted for and to obtain refined channel and spatial dimensions for the weighted fusion of feature maps $F_{out}(X_{fusion}) \in \mathbb{R}^{H\times W\times C}$. The specific details are depicted in Equation (6):

$$F_{out} = W_5 \otimes X_{ca} \oplus W_3 \otimes X_{sa}. \quad (6)$$

4. Experiments

We evaluate AMF-SparseInst's accuracy and inference speed on the COCO and Cityscapes dataset.

4.1. Experimental Environment

The model proposed in this paper has been studied experimentally and the experimental environment are shown in Table 2.

Table 2
Experimental environment

Name	Version
CPU	Intel(R) Xeon(R) Gold 5220 CPU @ 2.20GHz
RAM	252 G
Operating System	Ubuntu 18.04.6
GPU	GeForce RTX 2080Ti 8 GPUs
CUDA	CUDA 11.0
Language Frameworks	Python 3.8.12 detectron2 0.3 PyTorch 1.7.1

4.2. Evaluation Metrics

Currently, Average Precision (AP) is commonly sed to measure the performance of the instance segmentation model. AP denotes the area enclosed by the P-R curve with recall (R) and precision(P) as horizontal and vertical coordinates, and the formulas for recall and precision are as follows in Equation (7):

$$\begin{cases} P = \dfrac{TP}{TP+FP}, \\ R = \dfrac{TP}{TP+FN} \end{cases} \quad (7)$$

where TP (True Positive) is the number of samples detected as positive, FP (False Positive) is the number of samples misdetected as positive, FN (False Negative) is the number of samples missed as positive. For multiple samples of a certain classification, it is assumed that it has m positive samples, each positive sample corresponds to a recall rate R value (1/m, 2/m, ..., 1), calculate the maximum accuracy P for each re-

call rate, and then calculate the average value of these m P-values, as shown in the Equation (8):

$$AP = \frac{1}{m}\sum_{i=1}^{m} P_i = \int P(R)\,dR. \quad (8)$$

The above AP is for a particular category, while a dataset often contains multiple categories. Assuming that a dataset has C categories of samples, the mAP is obtained by averaging the AP for all categories in the dataset, as shown in Equation (9):

$$mAP = \frac{1}{C}\sum_{j=1}^{c} AP_j . \quad (9)$$

The evaluation indexes of the whole model are shown in Table 3. This paper uses the official evaluation standard of COCO dataset. AP is defined using the intersection-over-union (IOU) criterion, which is the overlap of two instance masks. The AP used in this paper is defaulted to the average accuracy of the dataset, which is also known as mAP. In Table 2, the segmentation accuracy of objects of different scales (large, small and medium) is calculated according to the area size. AP_S, AP_M, and AP_L correspond to the AP values of three different scales: small, medium, and large, respectively.

Table 3
Evaluation index of COCO dataset

Evaluation index	Meaning
FPS	Frames per second
AP	IOU=0.50:0.05:0.95
AP_{50}	IOU=0.50
AP_{75}	IOU=0.75
APs	area < 32^2
AP_M	32^2 < area < 96^2
AP_L	area > 96^2

4.3. Datasets and Implementation Details

The COCO 2017 dataset [21] is a generalized dataset for object detection and instance segmentation tasks, containing a training set (118,287), a test set (5,000), a test set (40,670) images and 80 foreground object categories, and one background category. In our experiments, we used a "polygon" annotation, where the short sides of the images were randomly sampled from 416 to 640 pixels with an interval size of 32, while the long sides were less than or equal to 864 pixels. Unless otherwise stated, we use the short-edge size 640 to evaluate speed and accuracy. The initial learning rate is set to 0.00005, the weight decay is 0.01, the model optimizer is Adam, the batch size is 64, and the iterations are trained for 270,000 times.

Cityscapes [5] includes street scenes taken from 50 different cities. It provides high-quality annotations for 5,000 pixel-level frames, of which 2,975 frames are used for training, 500 for validation, and 1,525 for testing. In our experiments, we cropped the short edges of the images from 800 to 1024 pixels with a spacing of 32, while the long edges were less than or equal to 2048 pixels. Unless otherwise stated, we used the short-edge size of 1024 to evaluate speed and accuracy. The initial learning rate was set to 0.000006, the batch size was 8, and 72,000 iterations were trained.

4.4. Ablation Experiments on COCO 2017 Validation Set

To investigate the performance of the SimAM-ASPP and FEM modules in AMF-SparseInst, ablation experiments are designed as shown in Table 4. The experimental results show that when SparseInst's PPM module is replaced by SimAM-ASPP module, the model speed is increased by 6.9 FPS, the model can recognize more objects, and the overall segmentation accuracy is increased by 1.5%. In addition, after changing the FPN connection, although the model speed is reduced by 2.2 FPS, it can recognize more small objects, and APs is increased by 1.2 percent. In addition, by adding FEM without changing the FPN connection, the model speed is 1 FPS slower and the AP_{50} is improved by 1.4%, but the small target accuracy is not improved much. Therefore, in order to balance the segmentation speed and improve the segmentation accuracy of small targets, we combine the three modules to get the final experimental results.

Table 5 shows the effect of modifying the spatial pyramid pooling module on the model. Using ResNet-50-d-DCNv2 as the backbone, these three different methods are sequentially added to SparseInst, trained and evaluated in the same way on the COCO dataset. Ordinary

Table 4
Contribution of each component under the backbone of ResNet-50-d-DCNv2

baseline	SimAM-ASPP	Lite-BiFPN	FEM	FPS	AP	AP50	AP75	APS	APM	APL
√	-	-	-	37.3	36.1	56.7	37.7	16.3	38.3	55.7
√	√	-	-	**44.2**	37.6	57.9	37.8	17.3	38.6	56.3
√	√	√	-	42.0	38.3	58.8	38.3	17.8	38.2	57.3
√	√	-	√	43.2	38.9	59.3	38.4	17.4	38.7	58.1
√	√	√	√	40.1	**39.6**	**60.8**	**38.9**	**18.7**	**39.5**	**59.4**

Table 5
Ablation on the SimAM-ASPP

ASPP	SimAM	AP	AP_{50}	AP_{75}	AP_S	T(ms)
conventional conv	-	36.6	57.3	37.1	16.0	29.2
conventional conv	√	37.0	57.2	37.4	16.8	26.4
depthwise conv	√	**37.6**	**57.9**	**37.8**	**17.3**	**22.6**

encoder [20] are not capable of one-stage prediction. We use ASPP to expand the receptive field to extract different context information, and then extract more multi-scale features through feature fusion. Notably, small objects have less feature information in the high level of CNN, and using SimAM helps to focus on small target features (AP_S +0.8%). In addition, the use of depthwise convolution not only greatly reduces the inference latency, but also improves the AP, especially for AP_S.

Based on the SimAM-ASPP and Lite-BiFPN, we do ablation experiments on various parts of the FEM. Table 6 demonstrates the impacts of the modifications to FEM module on the model. The output of BiFPN [25] includes low-level and high-level feature information, and different levels of features contain different semantic information. Thus, we design FEM to effectively fuse the two levels features. The FEM consists of channel attention (CA) and spatial attention (SA), and we explore how the two types of attention can be connected to get better results. As the table shows, using CA or SA alone causes the model to focus more on large objects and ignore small ones. It is worth noting that parallel attention can not only improve the efficiency of the model, but also allow the model to focus on small and large objects.

To better explain the features of different layers of the network, we drew the features extracted from the five stages of the backbone network (from C1 to C5), as shown in Figure 5. It can be seen that from C1 to C3, networks mainly focus on low-level features such as the outline and position of the target, while C4 and C5 networks focus on high-level features such as the con-

Table 6
Ablation on the FEM

CA	SA	connection	AP	AP_{50}	AP_S	AP_L	T(ms)
√	-	-	38.4	58.9	17.3	57.4	25.8
-	√	-	38.5	59.4	17.5	57.4	25.5
√	√	serial	39.1	59.6	18.5	59.3	26.7
√	√	parallel	**39.6**	**60.8**	**18.7**	**59.4**	**24.9**

Figure 5

The heatmap of each stage obtained by Grad-CAM

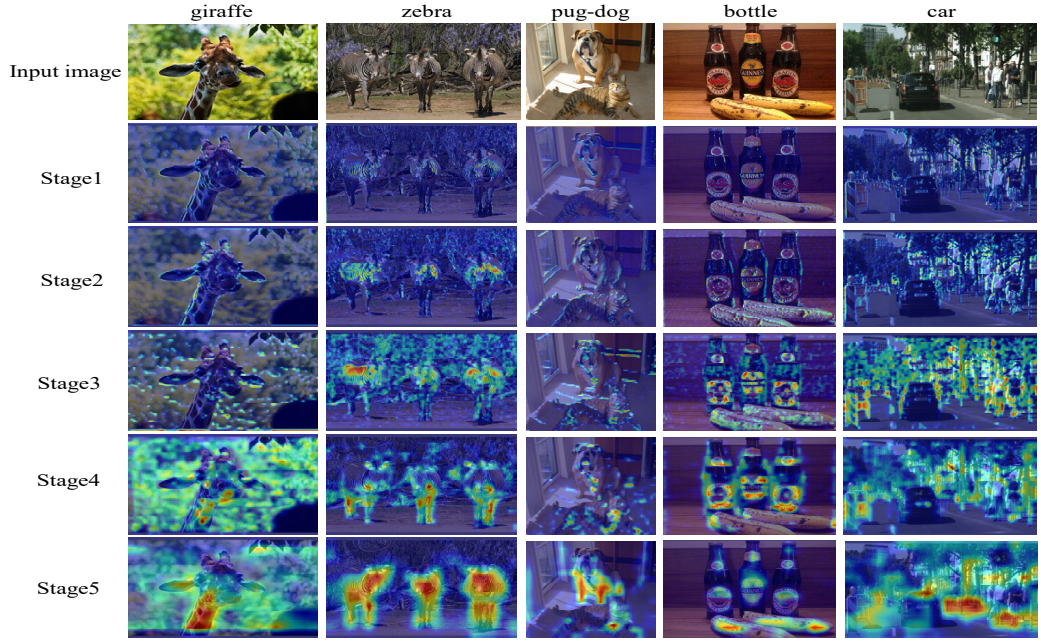
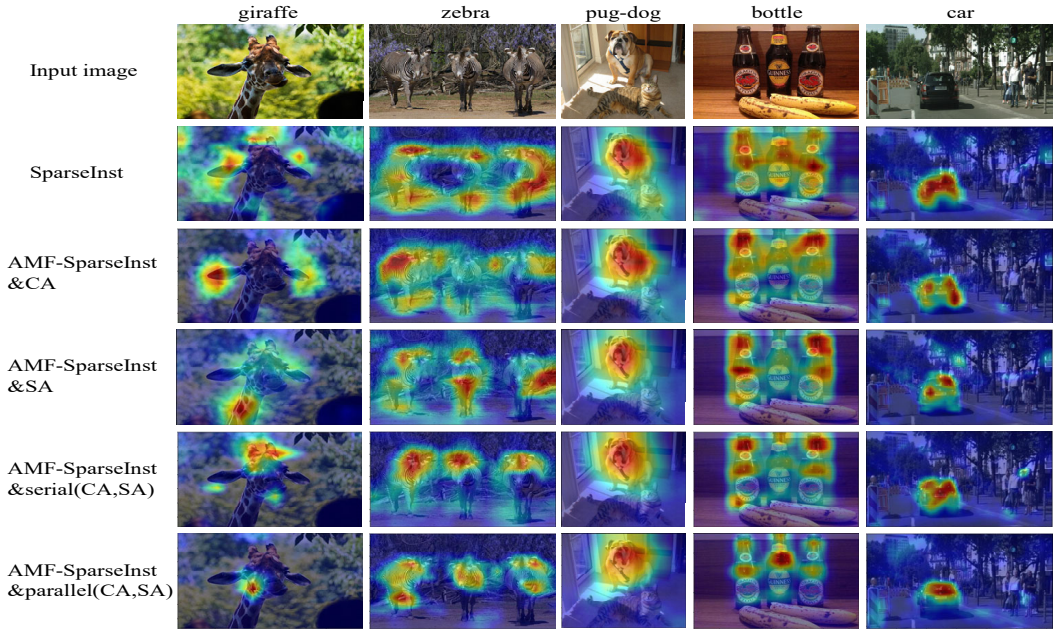


Figure 6

The attention heatmap comparison of models adding attention mechanisms or not obtained by Grad-CAM



textual semantics of the target. In order to more intuitively understand the capabilities of our proposed feature enhancement module, we use Grad-CAM to make a visual attention heat map from the COCO 2017 and Cityscapes dataset, as shown in Figure 6. From top to bottom: the input images, SparseInst without adding any attention, AMF-SparseInst with CA, SA, serial (CA, SA), objects into different methods and get the concerns of several methods. Obviously, using channel attention and spatial attention in parallel works best. In feature fusion, it can accurately locate objects of different scales without being affected by the background. For large scale objects, the resulting attention focuses on highlighting particularities, such as the head, legs, or center of the object. Small scale objects usually contain entire regions.

In the hyperparameter settings of the SimAM-ASPP module, we compare the effectiveness of several sets of atrous convolution kernels with different expansion rate sizes, as shown in Table 7. It is found that (6,18,24) achieves the best balance of speed and accuracy.

In the FEM, we set two hyperparameters, W_3 and W_5, which represent the weight sizes of the N3 and N5 features, respectively. Regarding the choice of the two parameter values, we designed five combinations, and found that (0.6,0.4) has the least latency and highest accuracy, as shown in Table 8.

Table 7
Speed-accuracy trade-off of expansion rates in SimAM-ASPP

Expansion rates in SimAM-ASPP	GFLOPs	Latency (ms)	AP (%)
(6,12,18)	94.6G	23.7	36.8
(6,18,24)	95.4G	**22.7**	**37.6**
(6,18,36)	96.6G	23.1	37.2

Table 8
Speed-accuracy trade-off of FEM

W3, W5	GFLOPs	Latency	AP (%)
(0.2,0.8)	96.9G	26.3	38.9
(0.4,0.6)	95.2G	25.5	39.0
(0.5,0.5)	94.2G	25.2	39.3
(0.6,0.4)	93.9G	24.9	**39.6**
(0.7,0.3)	94.7G	**24.4**	38.9

4.5. Timing

To more easily understand the efficiency of our proposed method, we evaluated the inference latency of three modules in AMF-SparseInst for two types of backbone. To accurately record the time, we disabled asynchronous execution in the GPU, which reduces the model inference speed. As is shown in Table 9, the backbone and encoder-decoder consume most of the inference time, while post-processing takes only 2~3ms to process the final segmentation and recognition results in evaluation.

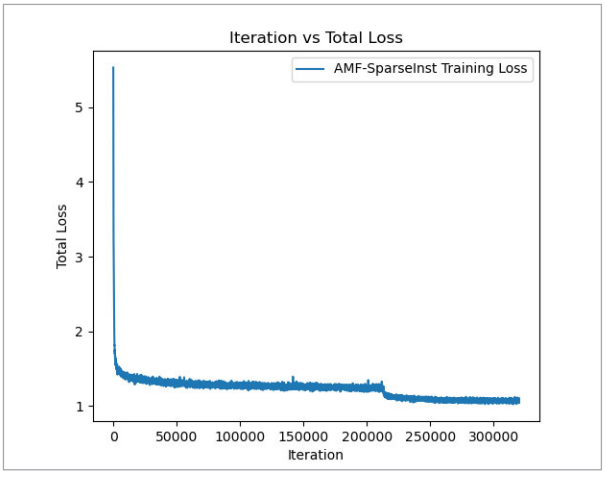
Table 9
Inference time (ms)

Type	Backbone	Encoder-decoder	Post processing
R-50	8.1 (45.3%)	7.5 (42.2%)	2.2 (12.5%)
R-50-d-DCNv2	11.8 47.2%)	10.1 (40.6%)	3.1 (12.2%)

4.6. Training Loss

The model training Settings include ADAMW optimizer, WarmupMultiStepLR learning rate scheduling, the initial learning rate is linearly adjusted by Warmup to 0.00005, the maximum iteration is 300,000 times, and the momentum is 0.9. Weight attenuation is 0.05, bias and normalized layer weight attenuation is 0.0001 and 0.0, respectively. Each batch processes 64 images. Figure 7 shows that as

Figure 7
Changes in the number of iterations and total loss during AMF-Sparseinst training



the number of iterations increases, both the total loss and the AMF-Sparseinst training loss gradually decrease. The model learns features quickly in the early stage, and the learning rate decreases ten times after 210,000 iterations, achieving loss reduction. After 250,000 iterations, the training becomes stable. This process reflects the optimization and steady improvement of model performance.

4.7. Comparison with State-of-the-Art Methods

We compare AMF-SparseInst to some of the most advanced real-time instance segmentation methods in terms of accuracy and inference speed, primarily using the COCO test-dev dataset for validation on a 2080Ti GPU with cuda version 11.0 and PyTorch version 1.7.1. We provide different backbones for AMF-SparseInst to realize the balance between speed and accuracy. We use ResNet-50 [11] to achieve higher inference speed and ResNet-d [13] to achieve better accuracy but with higher latency. In addition, for better comparison with YOLACT [2], we used simple randomized cropping and larger weight decay (0.05). For other methods, we chose the experimental results with the similar input size and backbone. Due to the limitations of the experimental environment and time, we did not achieve exactly the same parameter Settings as this method for training and repetition.

At Figure 8, we can see that the proposed AMF-SparseInst with R50-d and DCN [44] obtains better balance compared with the counterparts and achieves 39.7 FPS and 39.8 AP with 640× input, which outperforms most real-time methods (>=30FPS).

The validity of the proposed AMF-SparseInst is further verified by comparing the instance segmentation effects of AMF-SparseInst and other representative and real-time algorithms on the COCO 2017 test-dev and the Cityscapes dataset. Note that the speeds of all models are tested on a NVIDIA RTX 2080Ti without data augmentation. In addition, † means data augmentation (crop in "absolute_range"). If we use data augmentation, we can improve the accuracy of our model by 0.6% or so. Moreover, SparseInst's experimental data was trained and validated in our environment, and there are some differences. The experimental results are shown in Table 10, and it can be seen that AMF-SparseInst obtains better segmentation accuracy and faster speed on the two datasets. On COCO, although we outperform all real-time methods on the AP metric, AMF-SparseInst outperforms the other real-time methods by a larger margin on the

Figure 8
Intuitive comparison of speed and accuracy of AMF-SparseInst and some state-of-the-art methods

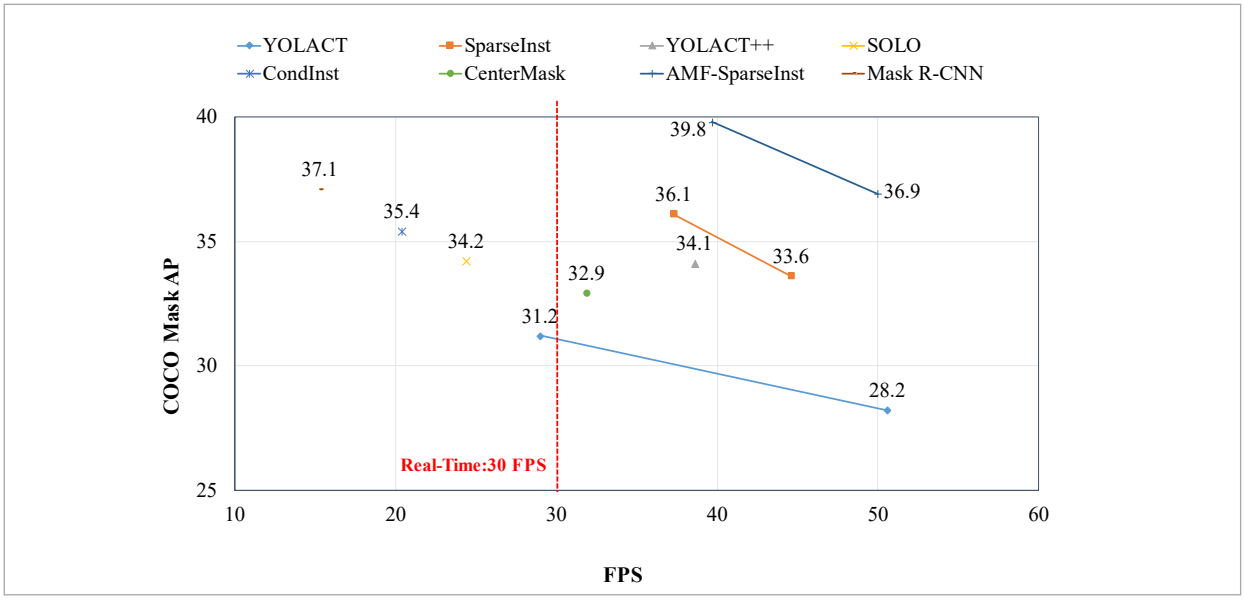


Table 10
Comparisons with state-of-the-art methods of Instance Segmentation on COCO and Cityscapes

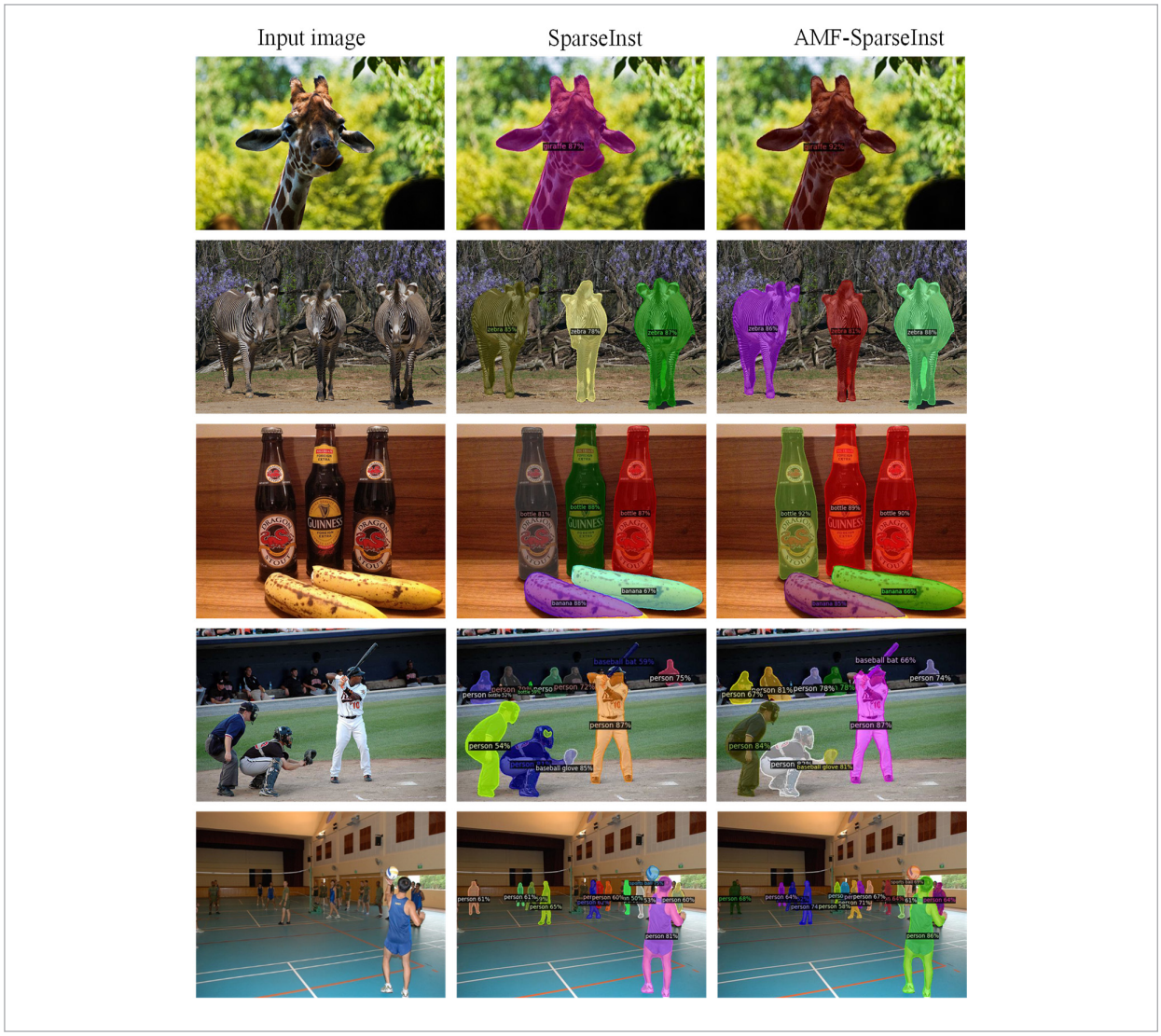
method	backbone	size	FPS	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Results on COCO 2017(the size refers to the shorter side size)									
CenterMask [19]	R-50-FPN	600	31.9	32.9	-	-	12.9	34.7	48.7
CondInst [27]	R-50-FPN	800	20.4	35.4	56.4	37.6	18.4	37.9	46.9
SOLO [30]	R-50-FPN	512	24.4	34.2	55.9	36.0	-	-	-
YOLACT [2]	R-50-FPN	550	50.6	28.2	46.6	29.2	9.2	29.3	44.8
YOLACT++ [1]	R-50-DCN-FPN	550	38.6	34.1	53.3	36.2	11.7	36.1	53.6
SparseInst [4]	R-50	608	44.6	34.2 †	55.3	36.6	14.3	36.2	50.7
AMF-SparseInst	R-50	608	**50.1**	36.9 †	-	-	-	-	-
SparseInst [4]	R-50	608	44.6	33.6	-	-	-	-	-
SparseInst [4]	R-50-d-DCNv2	608	37.6	36.0	56.6	37.8	16.4	38.1	55.6
SparseInst [4]	R-50-d-DCNv2	640	37.3	36.1	56.7	37.7	16.3	38.3	55.7
AMF-SparseInst	R-50-d-DCNv2	608	40.1	39.6	60.8	38.9	**18.7**	**39.5**	59.4
AMF-SparseInst	R-50-d-DCNv2	640	39.7	**39.8**	**60.9**	**39.0**	18.4	39.4	**59.6**
Results on Cityscapes									
CenterPoly [23]	R-50-DCNv2	512×512	62.5	9.2	22.0	-	-	-	-
CenterPoly [23]	R-50-DCNv2	1024×512	43.4	15.4	36.9	-	-	-	-
SparseInst [4]	R-50-d-DCNv2	1024×512	22.6	24.3	45.1	-	-	-	-
SparseInst [4]	R-50-d-DCNv2	1024×2048	12.7	33.5	55.1	-	-	-	-
AMF-SparseInst	R-50-d-DCNv2	1024×512	27.7	26.4	46.2	25.1	3.1	19.1	56.3
AMF-SparseInst	R-50-d-DCNv2	1024×2048	16.7	**36.7**	**60.7**	**37.2**	**8.7**	**33.3**	**63.4**

AP_{50} metric. This indicates that our mask is very accurate for coarse segmentation, but slightly inaccurate for very fine segmentation, which can be explained by the properties of polygons. On Cityscapes, AP metrics and real-time speed were significantly improved, 5.1% AP and 2.1 FPS higher than SparseInst, respectively, but the real-time speed was much lower than centerpoly [23].

Figure 9 shows a graphical comparison of the segmentation results of SparseInst and AMF-SparseInst on the COCO val. The central number annotated on each instance's segmentation mask is the confidence score, and it is evident that the confidence scores in the AMF-SparseInst visualizations are generally high. It can be seen that compared to SparseInst, AMF-SparseInst recognizes more objects and can segment more detail regions and multi-scale.

Meanwhile, since the CityPersons dataset is a subset of Cityscapes, which contains only personal annotations, we use the CityPersons dataset for visualizations as Figure 10. We can notice that AMF-SparseInst can accurately segment pedestrian legs, even very small ones. In addition, we can segment dense instances of very small objects, which shows that AMF-SparseInst does perform significantly better instance segmentation for small objects.

Figure 9
Comparison of SparseInst and AMF-SparseInst visualization on COCO 2017 validation set



5. Conclusion

We design the SimAM-ASPP consisting of depth-wise convolution and the SimAM attention, which is used to aggregate contextual information at different scales. In addition, in order to weaken the interference of redundant contextual information in multi-scale fusion features and alleviate the dispersion problem of small object features, we design the FEM, which generates adaptive weights in the channel and spatial dimensions to guide the features to learn in a more critical direction. Combining SimAM-ASPP and FEM, we present an efficient end-to-end framework called AMF-SparseInst, which solves the small object feature loss problems. The results on COCO 2017 and Cityscapes show that AMF-SparseInst excels in balancing segmentation accuracy and speed, outperforming other real-time and state-of-the-art methods. In the future, we will further explore a higher balance point for high-precision real-time instance segmentation from other perspectives, and customize models of different sizes for different object segmen-

Figure 10
Comparison of SparseInst and AMF-SparseInst visualization on CityPersons dataset

tation tasks according to the needs of industrial applications (e.g., autonomous driving and robotics).

5.1. Limitations

The framework proposed in this paper is real-time and can be used in practical industrial applications. However, compared with large and medium targets, the segmentation effect of this framework for small targets is poorer. We speculate that the lack of high-resolution images or high-resolution inputs in the dataset limits the performance of the framework. In the future, we will continue to study this problem in combination with super-resolution image tasks or generative adversarial network strategies such as Inst-GAN [18]. In addition, the focus of this paper is effectively combined and improved by SimAM and parallel CBAM to strengthen spatial and contextual semantic connections between different targets. However, other perspectives such as the latest attention mechanism (Transformer) [6], graph neural network or generative adversarial network may be more clever to establish the connection between the spatial layout and geometry of the target.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62262004) (http://www.nsfc.gov.cn). The funders had no role in study design, data collection and analysis, decision to publish, or preparation cpnof the manuscript.

Author Contributions

Conceptualization: Liang Wan.
Data curation: Shusheng Li.
Formal analysis: Yiyi Chen.
Investigation: Liang Liao.
Model design and training: Yiyi Chen.
Validation: Yiyi Chen, Liang Wan.
Writing original draft: Yiyi Chen.
Writing review & editing: Yiyi Chen, Liang Wan.

References

1. Bolya, D., Zhou, C., Xiao, F., Lee, Y. J. Yolact: Real-Time Instance Segmentation. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 9157-9166. https://doi.org/10.1109/ICCV.2019.00925
2. Bolya, D., Zhou, C., Xiao, F., Lee, Y. J. Yolact++: Better Real-Time Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. https://doi.org/10.1109/TPAMI.2020.3014297
3. Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. Encoder-Decoder with Atrous Depthwise Separable Convolution for Semantic Image Segmentation. Proceedings of the European Conference on Computer Vision (ECCV), 2018: 801-818. https://doi.org/10.1007/978-3-030-01234-2_49
4. Cheng, T., Wang, X., Chen, S., Zhang, W., Zhang, Q., Huang, C., Zhang, Z., Liu, W. Sparse Instance Activation for Real-Time Instance Segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, 4433-4442. https://doi.org/10.1109/CVPR52688.2022.00439
5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. IEEE, 2016. https://doi.org/10.1109/CVPR.2016.350
6. Fang, Y., Yang, S., Wang, X., Li, Y., Fang, C., Shan, Y., Feng, B., Liu, W. Instances as Queries. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 6910-6919. https://doi.org/10.1109/ICCV48922.2021.00683
7. Fu, B., He, J., Zhang, Z., Qiao, Y. Dynamic Sampling Network for Semantic Segmentation. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(07), 10794-10801. https://doi.org/10.1609/aaai.v34i07.6709
8. Hafiz, A. M., Bhat, G. M. A Survey on Instance Segmentation: State of the Art. International Journal of Multimedia Information Retrieval, 2020, 9(3), 171-189. https://doi.org/10.1007/s13735-020-00195-x
9. He, J., Deng, Z., Zhou, L., Wang, Y., Qiao, Y. Adaptive Pyramid Context Network for Semantic Segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 7519-7528. https://doi.org/10.1109/CVPR.2019.00770
10. He, K., Gkioxari, G., Dollár, P., Girshick, R. Mask R-CNN. Proceedings of the IEEE International Conference on Computer Vision, 2017, 2961-2969. https://doi.org/10.1109/ICCV.2017.322
11. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90
12. He, K., Zhang, X., Ren, S., Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9), 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824
13. He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M. Bag of Tricks for Image Classification with Convolutional Neural Networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 558-567. https://doi.org/10.1109/CVPR.2019.00065
14. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861, 2017. https://doi.org/10.48550/arXiv.1704.04861
15. Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q., Adam, H. Searching for Mobilenetv3. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 1314-1324. https://doi.org/10.1109/ICCV.2019.00140
16. Hu, J., Shen, L., Sun, G. Squeeze-and-Excitation Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745
17. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X. Mask Scoring R-CNN. Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition, 2019, 6409-6418. https://doi.org/10.1109/CVPR.2019.00657

18. Lauenburg, L., Lin, Z., Zhang, R., Santos, M., Huang, S., Arganda-Carreras, I., Boyden, E. S., Pfister, H., Wei, D. Instance Segmentation of Unlabeled Modalities via Cyclic Segmentation GAN. arXiv preprint arXiv:2204.03082, 2022. https://doi.org/10.48550/arXiv.2204.03082
19. Lee, Y., Park, J. Centermask: Real-Time Anchor-Free Instance Segmentation. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 13906-13915. https://doi.org/10.1109/CVPR42600.2020.01392
20. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. Feature Pyramid Networks for Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 2117-2125. https://doi.org/10.1109/CVPR.2017.106
21. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L. Microsoft COCO: Common Objects in Context. Computer Vision - ECCV 2014, 2014, 740-755. https://doi.org/10.1007/978-3-319-10602-1_48
22. Lou, A., Loew, M. Cfpnet: Channel-wise Feature Pyramid for Real-time Semantic Segmentation. In: 2021 IEEE International Conference on Image Processing (ICIP). IEEE, 2021, 1894-1898. https://doi.org/10.1109/ICIP42928.2021.9506485
23. Perreault, H., Bilodeau, G A., Saunier, N., Héritier, M. Centerpoly: Real-time Instance Segmentation Using Bounding Polygons. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 2982-2991. https://doi.org/10.1109/ICCVW54120.2021.00333
24. Rashwan, A., Du, X., Yin, X., Li, J. Dilated SpineNet for Semantic Segmentation. arXiv preprint arXiv:2103.12270, 2021. https://doi.org/10.48550/arXiv.2103.12270
25. 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, 2020, 10781-10790. https://doi.org/10.1109/CVPR42600.2020.01079
26. Tian, Z., Shen, C., Chen, H., He, T. Fcos: Fully Convolutional One-Stage Object Detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 9627-9636. https://doi.org/10.1109/ICCV.2019.00972
27. Tian, Z., Shen, C., Chen, H. Conditional convolutions for instance segmentation. In: Computer Vision - ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I. Springer International Publishing, 2020, 282-298. https://doi.org/10.1007/978-3-030-58452-8_17
28. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B. Deep High-Resolution Representation Learning for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(10), 3349-3364. https://doi.org/10.1109/TPAMI.2020.2983686
29. Wang, K., Liew, J. H., Zou, Y., Zhou, D., Feng, J. Panet: Few-shot Image Semantic Segmentation with Prototype Alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 9197-9206. https://doi.org/10.1109/ICCV.2019.00929
30. Wang, X., Kong, T., Shen, C., Jiang, Y., Li, L. Solo: Segmenting Objects by Locations. In: European Conference on Computer Vision. Springer, Cham, 2020, 649-665. https://doi.org/10.1007/978-3-030-58523-5_38
31. Wang, X., Zhang, R., Kong, T., Li, L., Shen, C. Solov2: Dynamic and Fast Instance Segmentation. In: Advances in Neural Information Processing Systems, 2020, 33, 17721-17732. https://doi.org/10.48550/arXiv.2003.10152
32. Woo, S., Park, J., Lee, J.-Y., Kweon, I. S. CBAM: Convolutional Block Attention Module. In: Computer Vision - ECCV 2018, 2018, 3-19. https://doi.org/10.1007/978-3-030-01234-2_1
33. Xie, E., Sun, P., Song, X., Wang, W., Liang, D., Shen, C., Luo, P. Polarmask: Single Shot Instance Segmentation with Polar Representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 12193-12202. https://doi.org/10.1109/CVPR42600.2020.01221
34. Xu, W., Wang, H., Qi, F., Lu, C. Explicit Shape Encoding for Real-time Instance Segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 5168-5177. https://doi.org/10.1109/ICCV.2019.00527
35. Yang, L., Zhang, R Y., Li, L., Xie, X. Simam: A Simple, Parameter-free Attention Module for Convolutional Neural Networks. In: International Conference on Machine Learning. PMLR, 2021, 11863-11874. https://proceedings.mlr.press/v139/yang21o
36. Zhang, X., Zhou, X., Lin, M., Sun, J. Shufflenet: An Extremely Efficient Convolutional Neural Network for

Mobile Devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, 6848-6856. https://doi.org/10.1109/CVPR.2018.00716

37. Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J. Icnet for Real-time Semantic Segmentation on High-Resolution Images. In: Proceedings of the European Conference on Computer Vision (ECCV), 2018, 405-420. https://doi.org/10.1007/978-3-030-01219-9_25
38. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J. Pyramid Scene Parsing Network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, 2881-2890. https://doi.org/10.1109/CVPR.2017.660
39. Zheng, Q., Zhao, P., Zhang, D., Wang, H. MR-DCAE: Manifold Regularization-based Deep Convolutional Autoencoder for Unauthorized Broadcasting Identification. International Journal of Intelligent Systems, 2021, 36(10), 7204-7238. https://doi.org/10.1002/int.22586
40. Zheng, Q., Tian, X., Yu, Z., Jiang, N., Elhanashi, A., Saponara, S., Yu, R. Application of Wavelet-packet Transform Driven Deep Learning Method in PM2.5 Concentration Prediction: A Case Study of Qingdao, China. Sustainable Cities and Society, 2023, 92, 104486. https://doi.org/10.1016/j.scs.2023.104486
41. Zheng, Q., Tian, X., Yu, Z., Wang, H., Elhanashi, A., Saponara, S. DL-PR: Generalized Automatic Modulation Classification Method Based on Deep Learning with Priori Regularization. Engineering Applications of Artificial Intelligence, 2023, 122, 106082. https://doi.org/10.1016/j.engappai.2023.106082
42. Zheng, Q., Zhao, P., Li, Y., Wang, H., Yang, Y. Spectrum Interference-Based Two-Level Data Augmentation Method in Deep Learning for Automatic Modulation Classification. Neural Computing and Applications, 2021, 33(13), 7723-7745. https://doi.org/10.1007/s00521-020-05514-1
43. Zheng, Q., Zhao, P., Wang, H., Elhanashi, A., Saponara, S. Fine-grained Modulation Classification Using Multi-Scale Radio Transformer with Dual-Channel Representation. IEEE Communications Letters, 2022, 26(6), 1298-1302. https://doi.org/10.1109/LCOMM.2022.3145647
44. Zhu, X., Hu, H., Lin, S., Dai, J. Deformable ConvNets v2: More Deformable, Better Results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 9308-9316. https://doi.org/10.1109/CVPR.2019.00953
45. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. arXiv preprint arXiv:2010.04159, 2020. https://doi.org/10.48550/arXiv.2010.04159