electronics

MDPI

Article

Road Scene Instance Segmentation Based on Improved SOLOv2

Qing Yang [1], Jiansheng Peng [1,2,*], Dunhua Chen [1] and Hongyu Zhang [1]

[1] College of Automation, Guangxi University of Science and Technology, Liuzhou 545000, China; 221068416@stdmail.gxust.edu.cn (Q.Y.); 221068337@stdmail.gxust.edu.cn (D.C.); 221077113@stdmail.gxust.edu.cn (H.Z.)
[2] Department of Artificial Intelligence and Manufacturing, Hechi University, Hechi 547000, China
* Correspondence: pengjs@hcnu.edu.cn; Tel.: +86-139-0778-6821

Abstract: Road instance segmentation is vital for autonomous driving, yet the current algorithms struggle in complex city environments, with issues like poor small object segmentation, low-quality mask edge contours, slow processing, and limited model adaptability. This paper introduces an enhanced instance segmentation method based on SOLOv2. It integrates the Bottleneck Transformer (BoT) module into VoVNetV2, replacing the standard convolutions with ghost convolutions. Additionally, it replaces ResNet with an improved VoVNetV2 backbone to enhance the feature extraction and segmentation speed. Furthermore, the algorithm employs Feature Pyramid Grids (FPGs) instead of Feature Pyramid Networks (FPNs) to introduce multi-directional lateral connections for better feature fusion. Lastly, it incorporates a convolutional Block Attention Module (CBAM) into the detection head for refined features by considering the attention weight coefficients in both the channel and spatial dimensions. The experimental results demonstrate the algorithm's effectiveness, achieving a 27.6% *mAP* on Cityscapes, a 4.2% improvement over SOLOv2. It also attains a segmentation speed of 8.9 FPS, a 1.7 FPS increase over SOLOv2, confirming its practicality for real-world engineering applications.

Keywords: instance segmentation; road scene; SOLOv2; VoVNetV2; FPN

check for updates

Citation: Yang, Q.; Peng, J.; Chen, D.; Zhang, H. Road Scene Instance Segmentation Based on Improved SOLOv2. *Electronics* **2023**, *12*, 4169. https://doi.org/10.3390/electronics12194169

Academic Editor: Hyunjin Park

Received: 15 September 2023
Revised: 4 October 2023
Accepted: 5 October 2023
Published: 8 October 2023

Copyright: © 2023 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

In today's society, with the accelerated pace of urbanization and the continuous growth in transportation demands, road scene segmentation has become a technology of paramount significance. Road scene segmentation involves the precise separation and identification of various objects on roads within digital images or videos from their surrounding environments. This technology holds extensive prospects in fields such as autonomous driving, traffic monitoring, and intelligent transportation systems. The objective of road scene segmentation is to achieve a comprehensive understanding and perception of the traffic environment by accurately segmenting elements like vehicles, pedestrians, and traffic signs on the road. By effectively separating all objects on the road, road scene segmentation provides autonomous vehicles with the essential environmental awareness to ensure safe driving. Simultaneously, in traffic monitoring systems, road scene segmentation can be employed to monitor the traffic flow in real time, detect violations, and optimize the traffic signal control, thereby enhancing the road safety and traffic efficiency. However, due to the complexity and diversity of road scenes, road scene segmentation tasks encounter a series of challenges. For instance, the variations in the lighting conditions, the influence of weather conditions, vehicle obstructions, and overlapping factors all impact the accuracy and stability of scene segmentation. Therefore, effectively addressing these challenges and enhancing the precision and robustness of road scene segmentation have become the current focus and hotspots of research.

To address these challenges, researchers have continuously advanced the research and development of road scene segmentation using techniques such as threshold-based

methods, edge detection, and deep learning. The threshold-based methods like histogram-based and Otsu's thresholding [1] primarily consider the grayscale values of individual pixels, making them sensitive to noise. Moreover, when dealing with images containing complex backgrounds and numerous objects, a single threshold often fails to extract comprehensive features, resulting in a poor mask quality. On the other hand, the edge-based segmentation algorithms aim to solve segmentation issues by detecting edges within the regions of interest. However, due to the often-dramatic grayscale variations at different edges, these algorithms may fail to ensure that the extracted contours possess continuity and closure, limiting their effectiveness. In response to these limitations, researchers have turned to deep learning approaches to overcome these challenges and improve the accuracy and robustness of road scene segmentation. The deep learning methods leverage neural networks to learn complex features from the data, allowing for more accurate and adaptable segmentation, even in the presence of noise and complex backgrounds.

Diverging from the traditional image segmentation algorithms based on thresholds and edge detection, the deep learning-based image segmentation methods exhibit significant advantages. Full Convolutional Networks (FCNs) [2], for example, replace fully connected layers with a defined number of convolutional and pooling layers, facilitating the neural network's transition from object detection to object segmentation. As deep learning has evolved, segmentation has been further refined into semantic and instance segmentation. Semantic segmentation involves labeling each pixel in an image with class labels, assigning the same label to all detected objects of the same class. Consequently, semantic segmentation can only distinguish between the different types of objects and not between individual instances of the same type [3]. In contrast, instance segmentation predicts labels for each pixel belonging to an object, providing distinct labels for the different individuals within the same class. It simultaneously addresses both the object detection and semantic segmentation challenges. In busy traffic scenarios, the mutual occlusion between vehicles or pedestrians is common. When two vehicles partially overlap or are in close proximity, semantic segmentation networks struggle to separate their individual parts. In contrast, the instance segmentation techniques excel at achieving precise separation of such vehicles, offering more detailed road scene information.

Instance segmentation is a computer vision task that closely mirrors human visual perception, representing a deep understanding of image scenes. Applying the instance segmentation techniques to road scene segmentation holds immense practical value in the field of autonomous driving. However, the existing road scene instance segmentation algorithms exhibit shortcomings such as poor segmentation of small objects, low-quality mask edge contour delineation, slow processing speeds, and limited model generalization capabilities. In light of these challenges, this paper introduces a novel instance segmentation algorithm based on SOLOv2 [4], aiming to achieve real-time and precise segmentation of road scenes. Our primary contributions are outlined as follows:

(1) A new feature extraction network is proposed. Insert the Bottleneck Transformer (BoT) block [5] into VoVNetV2 [6], and replace the traditional convolution in VoVNetV2 with Ghost Conv (Ghost Convolution) [7]. The BoT block obtains the global dependency of the image by aggregating the local interaction information of the image, allowing the network to obtain long sequences of associated features. Ghost Conv can significantly reduce the network computing costs and memory usage by replacing some traditional convolutions with cheap linear operations. The improved backbone network not only enhances the feature extraction capabilities of the model, but also reduces the calculation amount and parameter amount of the model.
(2) Use Feature Pyramid Grids (FPG) [8] to replace the Neck part of the original network. FPG is a deep multi-path feature pyramid that fuses the feature in multiple directions in multi-scale space to obtain fine-resolution features with semantic information.
(3) Add Convolutional Block Attention Module (CBAM) [9] to the Head part of the original network. The CBAM attention mechanism first learns feature information

from the two main dimensions of the space and channel, and then fuses it to obtain the refined feature information.

2. Related Work

Currently, the deep learning-based instance segmentation methods can be broadly categorized into two-stage and one-stage approaches. The two-stage instance segmentation methods can be further divided into top-down and bottom-up approaches. The top-down methods involve object detection to locate the boxes containing each instance, followed by semantic segmentation within these boxes. The bottom-up methods, on the other hand, begin with semantic segmentation to identify pixels and then use clustering or other metric learning methods to distinguish the instances of the same class. The one-stage instance segmentation methods can generally be classified into two categories: those based on global image information and those based on local image information. The global image-based methods do not require cropping or ROI alignment processes. Instead, they initially form a feature map for the entire instance and then use various operations to combine features to obtain the final mask for each instance. The local region-based methods directly output instance segmentation results based on local information. In a sense, the bounding box serves as a rough mask, approximating the contour of the mask by using the smallest bounding rectangle.

2.1. Two-Stage

SDS [10] was one of the earliest instance segmentation algorithms, obtaining the segmentation results through recommendation generation, feature extraction, region classification, and region refinement. However, due to relying solely on CNN technology for feature extraction, it resulted in masks with coarse details and imprecise positional information. DeepMask [11] approached the image segmentation as an extensive binary classification problem, while SharpMask [12] enhanced the output of DeepMask, producing higher fidelity masks that accurately delineated the object boundaries. Fast RCNN [13] performed global feature extraction on the entire image, replacing the final max-pooling layer with RoI (Region of Interest) pooling. It also incorporated parallel different fully connected layers at the end of the network. Faster RCNN introduced the Region Proposal Network (RPN) to replace the selective search algorithm used in Fast RCNN, significantly improving network efficiency. Mask RCNN [14] extended Faster RCNN by adding a semantic segmentation branch, achieving better segmentation results. However, it heavily relied on the accuracy of the bounding boxes and exhibited a poorer performance in detecting small-scale objects. SCNet [15] leverages the global context information to reinforce the relationships between the classification, detection, and segmentation subtasks. FASA [16] dynamically generates virtual features, providing more positive samples for rare categories, and employs loss-guided adaptive strategies to prevent overfitting. RefineMask [17] and ISTR [18] combine the fine-grained features in the segmentation process, yielding high-quality masks for both large and small objects, although they come at a slower processing speed.

2.2. One-Stage

YOLACT [19] introduced a mask branch into the single-stage object detection models, achieving instance segmentation through two parallel branch tasks. However, it struggled to suppress the external noise outside the RoIs, leading to potential mask leakage. YOLACT++ addressed these issues by introducing deformable convolutional networks (DCN) into the model's backbone network, optimizing the prediction heads, and adding a Mask R-scoring network branch to enhance mask prediction quality. ContrastMask [20] made effective use of the training data, allowing data from new categories to contribute to the segmentation model's optimization process. It achieved this by transferring the segmentation ability of base categories to new ones through a unified pixel-level contrastive learning framework. However, it couldn't guarantee the correctness of the foreground and background partitioning for new categories. PolarMask [21] reformulated the in-

stance segmentation problem by predicting the instance contours through center-based classification and dense distance regression in polar coordinates. This greatly simplified the model training process. SOLO [22] defined instance segmentation as a dual problem involving the category-aware predictions and instance-aware mask generation. It predicted the instance categories through the Category Branch and obtained corresponding instance masks through the Mask Branch. SOLOv2, an improvement over SOLO, enhanced the Mask Branch by dividing it into the mask kernel prediction and mask feature learning components. These components were responsible for predicting the convolutional kernels and feature masks, respectively, and introduced the concept of Matrix Non-Maximum Suppression (Matrix NMS). While SOLOv2 addressed the issues of mask inefficiency, low resolution, and imprecise mask prediction, it still faced challenges in segmenting small-scale objects and achieving faster processing speeds. Tensormask [23] leveraged structured 4D tensors to represent the masks in spatial and elevation dimensions. BlendMask [24], a fusion of Mask RCNN and YOLACT, incorporated not only the object detection results but also the information from the FPN (Feature Pyramid Network). The PointRend method [25] by the team led by He K optimized the image segmentation at the object edges, improving the performance in challenging edge areas. E2ec [26], proposed by Zhang T and colleagues, is a multi-stage, efficient end-to-end contour-based instance segmentation model suitable for high-quality instance segmentation.

While two-stage instance segmentation methods offer a high segmentation accuracy, they struggle with poor real-time performance. Conversely, the single-stage instance segmentation methods have greatly improved the real-time capabilities but often struggle with segmenting small objects effectively. Therefore, the focus of this work is to enhance the mask representation of small objects and improve the network's accuracy in segmenting them.

3. Materials and Methods

To balance the accuracy and real-time performance of multi-object instance segmentation in complex backgrounds, this paper builds upon the single-stage instance segmentation network SOLOv2. We enhance the feature extraction capabilities of the backbone network, addressing the issue of information loss caused by the FPN's direct fusion of the high and low-level information. This strengthens both the spatial and channel features in the feature map, resulting in a more comprehensive and improved instance segmentation algorithm. The overall structure of the improved algorithm is as shown in Figure 1.

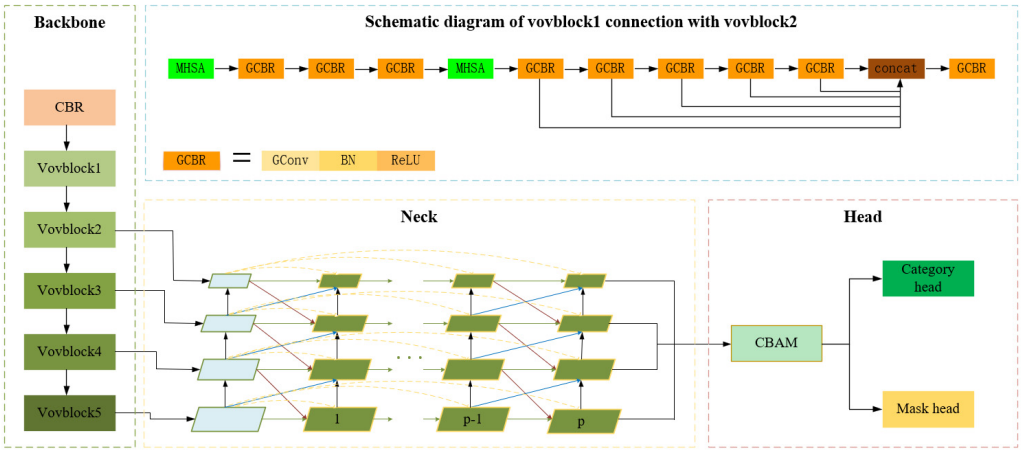

Figure 1. Schematic diagram of the improved algorithm structure.

The input image is first divided into $S \times S$ grids, and the features are extracted through the backbone network. The Neck part fuses the features and then enters the Head module to predict the instance categories and masks, respectively. Each grid in the Category Branch is responsible for predicting an instance. For the instance located at grid (i, j), the k-th channel represents its category probability, $k = i * S + j$. In order to reduce the amount of

network parameters and calculations, the Mask Branch is decoupled into the Kernel Branch and the Feature Branch, which are respectively responsible for generating the convolution kernel G and the feature map F that requires convolution. After convolution, the instance mask is obtained. For an instance located at grid (i, j), its mask is as shown in Equation (1).

$$M_{i,j} = G_{i,j} * F \tag{1}$$

3.1. Backbone

VoVNetV2 is an efficient lightweight backbone network. It first consists of three 3×3 convolutional layers to form a stem block to complete the downsampling operation. Then it goes through a four-stage One-Shot Aggregation (OSA) module. It is composed of convolutional layers, and the subsequent feature maps are immediately aggregated, fused with the Effective Squeeze-Excitation (eSE) module, and then the residual connections are added to obtain the final output. The eSE module aims at the problem of channel information loss due to size reduction in the Squeeze-and-Excitation (SE) attention mechanism. It uses two Full Connection (FC) layers that do not reduce the channel size to retain channel information, and improves the interaction between the feature map channels. The dependencies are explicitly modeled to enhance the representation ability of feature maps. The addition of the eSE module improves the network's ability to interact with information between image channels, but it lacks the impact of global information on the features. Therefore, this article adds a BoT block to the original OSA module to obtain global dependencies by aggregating the local interactions.

The BoT block moves self-attention into computer vision tasks, adding 1×1 convolutions before and after the Multi-Head Self-Attention (MHSA) structure. It has been confirmed in the literature [5] that BoT block greatly improves the small object detection. The OSA module that adds the BoT structure not only focuses on the aggregation of the local information, but also improves the network's attention to the global information, effectively combining the local information and global information, making the features of small objects in the image more prominent. The MHSA structure used in the BoT block is as shown in Figure 2.

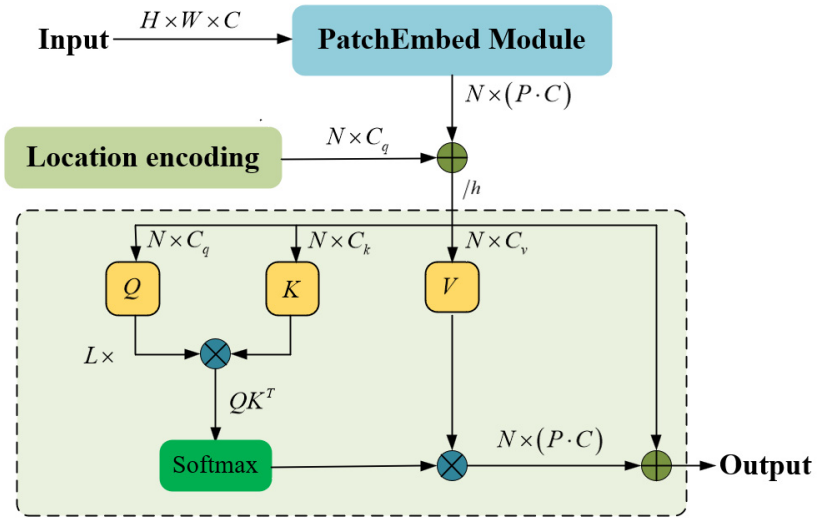

Figure 2. Schematic diagram of MHSA module structure.

In order to process the two-dimensional images, the PatchEmebd module is used to adjust the spatial dimension of the two-dimensional image $x \in R^{H \times W \times C}$ to a one-dimensional sequence $x_p \in R^{N \times (P \cdot C)}$, where (H, W) is the resolution of the input image, C is the number of channels, (P, P) is the resolution of each image block, and $N = HW/P^2$ is the number of image blocks, which is the effective input sequence length of the MHSA module [27]. In order to enable the MHSA module to utilize the sequence order information,

the position information about the sequence is added to the feature sequence, and three identical feature matrices Q, K, and V are generated. The feature matrix is projected h times to the C_q, C_k and C_v dimensions through linear projection to calculate the dot product attention in parallel [28]. Its calculation is as shown in Equation (2).

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{C_k}})V \tag{2}$$

Since the addition of the BoT block will inevitably lead to an increase in the model parameters and calculation amount, in order to solve this problem, this article uses Ghost Conv to replace the traditional convolution to reduce the model parameters and calculation amount. Ghost Conv is a method of compressing models that can generate more feature maps with fewer parameters, reducing the network parameters and calculation volume while ensuring the network accuracy. In order to reduce the amount of network calculations, Ghost Conv divides the traditional convolution into two steps as shown in Figure 3. First, a feature map with a smaller channel is generated through the traditional convolution, and then based on the obtained feature map, a cheap linear transformation operation is performed (depthwise conv) to generate a new feature map, and finally the two sets of feature maps are spliced together to obtain the final output feature map.

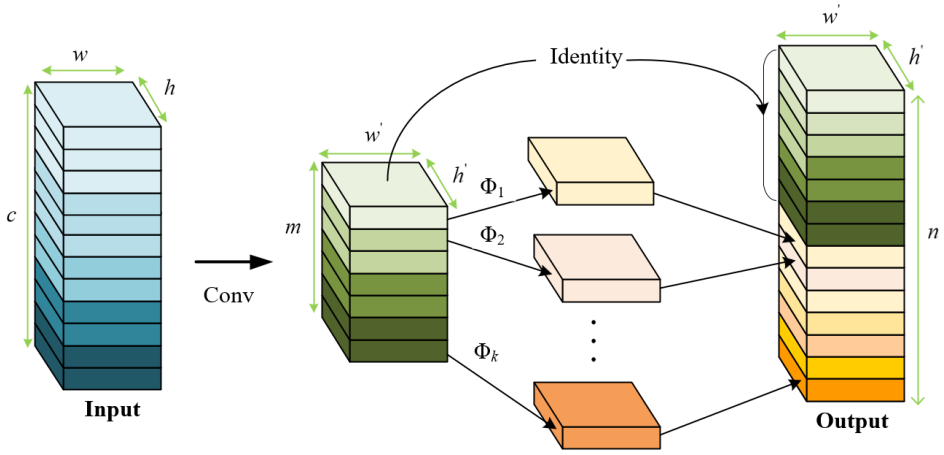

Figure 3. Schematic diagram of Ghost Conv structure.

Where c, h, and w respectively represent the channel, height, and width of the input image, m, h', and w' respectively represent the channel, height, and width of the intrinsic feature maps obtained after traditional convolution, n represents the final output feature map channel, the traditional convolution kernel size is k, Φ represents the depthwise conv, the depthwise convolution kernel size is d, after s transformations, the speed up ratio (r_s) between the calculation amount of the traditional convolution and the calculation amount of Ghost Conv is as shown in Equation (3).

$$\begin{aligned} r_s &= \frac{n * h' * w' * c * k * k}{m * h' * w' * c * k * k + (s-1) * m * h' * w' * d * d} \\ &= \frac{c * k * k}{\frac{1}{s} * c * k * k + \frac{s-1}{s} * d * d} \approx \frac{s * c}{s + c - 1} \approx s \end{aligned} \tag{3}$$

After s transformations, the compression ratio (r_c) between the parameter amount of the traditional convolution kernel and the parameter amount of the Ghost Conv convolution kernel is as shown in Equation (4).

$$r_s = \frac{n * c * k * k}{m * c * k * k + (s-1) * m * d * d} \approx \frac{s * c}{s + c - 1} \approx s \tag{4}$$

Since n is the number of channels of the final output feature map $n = m * s$, $m = \frac{n}{s}$, $s - 1$ is because the identify part does not need to be calculated. It can be seen from the

above calculation formula that compared with the traditional convolution, Ghost Conv can reduce the calculation amount and parameter amount of the model and achieve a faster calculation speed.

3.2. Neck

After SOLOv2 uses the backbone network to extract the image features, it obtains feature maps of different sizes at all levels through the Feature Pyramid Network (FPN) type Neck. Each level of the feature map enters the prediction head to predict the semantic categories and instance masks. In the feature map output of each size of the feature pyramid network, the small-size feature map has a larger receptive field and rich semantic information, but the resolution of the image is lower, the target position is rough, and the small target information is missing; while the large-size feature map has a large receptive field and rich semantic information. The size feature map has a high resolution and accurate target location, but the receptive field is small and lacks semantic information. In order to solve this problem, this paper builds fine resolution features by using an FPG instead of an FPN.

An FPG is a deep multi-path feature pyramid. The structure is as shown in Figure 4. The feature scale space is represented as a regular grid of parallel bottom-up paths and fused by multi-directional lateral connections. Different from an FPN, all independent pathways of an FPG are built from the bottom up, similar to the backbone pathway from the input image to the predicted output. To achieve information exchange at all levels of the image, an FPG interweaves the pyramid paths across scales and within scales with various lateral connections to form a deep network of feature pyramids.

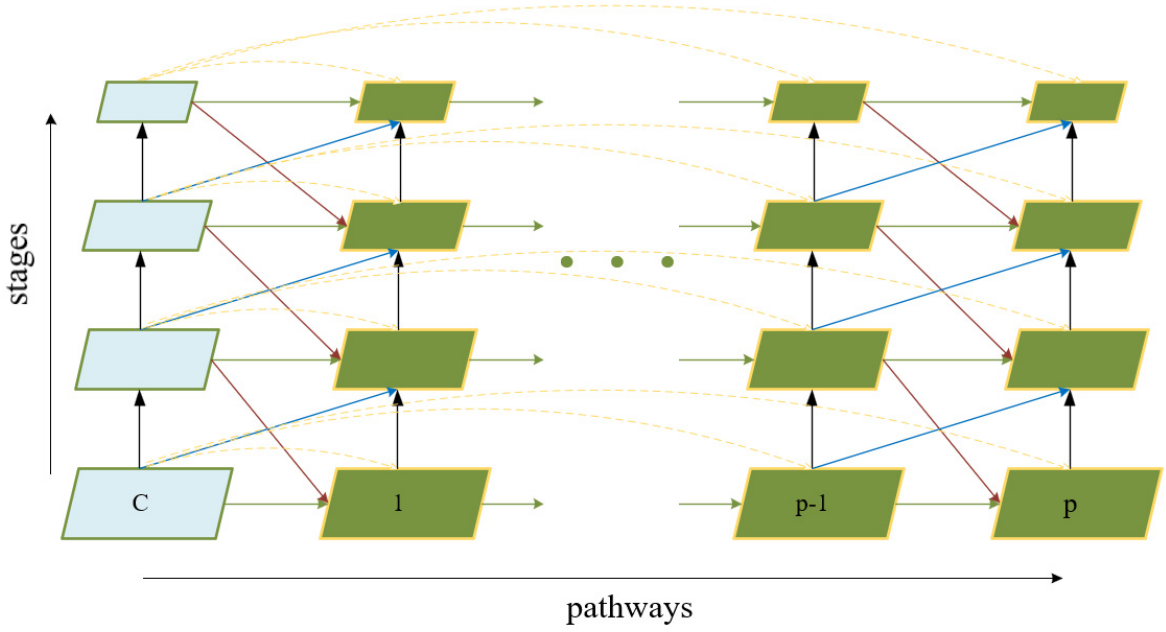

Figure 4. Schematic diagram of FPG module structure.

In Figure 4, the green arrow realizes feature fusion in the adjacent channels, the blue arrow shortens the path from the low-level features to the high-level features, the purple arrow fuses the upsampling features and downsampling features together, and the red arrow directly connects and shortens the network training time.

3.3. Head

A CBAM is an efficient lightweight attention module. Its structure is as shown in Figure 5. It obtains the image feature weights from two dimensions: channel and space, and then refines the feature map to enhance the representation ability of the feature map. The feature map undergoes global max pooling and global average pooling, respectively, to

obtain two weight vectors and let the weight vectors enter the same Multilayer Perceptron (*MLP*), respectively. The mapping weights output by the *MLP* are added at the pixel level and then activated by sigmoid to output the channel. Attention characteristics M_C, the calculation is as shown in Equation (5).

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F)) \\ &= \sigma\left(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1(W_0(F^c_{max}))\right) \end{aligned} \tag{5}$$

where σ is the sigmoid function, W_0 and W_1 are the *MLP* shared weights, and W_0 follows the ReLU activation function. Channel-multiply M_C and the input feature map to obtain F', and use it as the input of the Spatial Attention Module (SAM).

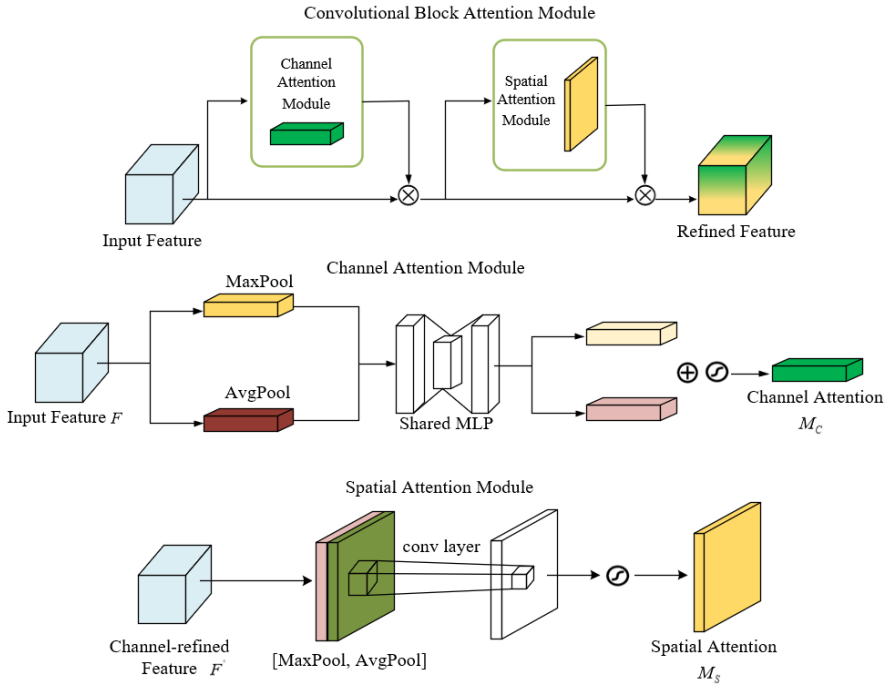

Figure 5. Schematic diagram of CBAM module structure.

In the SAM module, the global max pooling and global average pooling are first performed based on the channel to obtain two feature maps with a channel number of 1. After channel splicing, a 7×7 convolution operation is performed to reduce the channel to 1. After sigmoid obtains the spatial attention feature $M_S(F')$, and the calculation is as shown in Equation (6).

$$M_s(F') = \sigma\left(f^{7\times7}\left(\left[AvgPool(F'); MaxPool(F')\right]\right)\right) = \sigma\left(f^{7\times7}\left(\left[F'^s_{avg}; F'^s_{max}\right]\right)\right) \tag{6}$$

where σ is the sigmoid function, $f^{7\times7}$ represents 7×7 convolution. Multiply $M_S(F')$ and F' to get the final output.

4. Results

4.1. Dataset

This article uses the Cityscapes data set for the experiments. The Cityscapes dataset is a road scene object segmentation dataset that focuses on providing the training and performance testing for autonomous driving environment perception models. The images in this dataset come from video sequences of different road scenes in 50 cities in Germany and its neighboring countries, covering different street scenes, road scenes, and seasons. There are a total of 5000 finely annotated images and 2000 roughly annotated images. Since

instance segmentation requires a high level of data annotation, this experiment only uses finely annotated images, including 2975 finely annotated images for training, 1525 images for testing, and 500 images for validation. This article studies the instance segmentation algorithm in road scenes, so we selected the eight most common categories of road scenes in the Cityscapes dataset that contain instance segmentation labels, namely, person, rider, car, truck, bus, train, motorcycle, and bicycle. The number of images and instances for each category is shown in Table 1.

Table 1. The number of images and instances of each category in the dataset.

Categories	Number of Images	Number of Instances
Person	2343	18,406
Rider	1023	1762
Car	2832	27,963
Truck	359	482
Bus	274	379
Train	142	168
Motorcycle	513	743
Bicycle	1646	4157

4.2. Experimental Environment Configuration

The experimental environment for this experiment is shown in Table 2. Hyperparameter setting: Optimizer, AdamW; Weight decay, 0.0001; Learning rate (Lr), 0.0001; Epoch, 300; Batchsize, 4; Lr is adjusted at 20th, 30th, 40th epochs; Image scale is randomly sampled from 1024–2048.

Table 2. Experimental environment configuration.

Software and Hardware	Version and Model
CPU	Intel(R) Xeon(R) Silver 4214R CPU @ 2.40 GHz
GPU	Nvidia GeForce RTX 3080Ti
Operating System	Ubuntu18.04
Frame	Pytorch1.13.0, CUDA11.7

4.3. Experimental Evaluation Indicators

As this article is based on SOLOv2 for improvement, we will continue to use the original evaluation indicators of SOLOv2, that is, using the mean average precision (*mAP*) and mean average recall (*mAR*) to comprehensively evaluate the algorithm. The calculation is as shown in Equations (7) and (8).

$$mAP = \frac{1}{C}\sum_{c}\left(\frac{1}{|T|}\sum_{t}\frac{TP(t)}{TP(t)+FP(t)}\right) \tag{7}$$

$$mAR = \frac{1}{C}\sum_{c}\left(\frac{1}{|T|}\sum_{t}\frac{TP(t)}{TP(t)+FN(t)}\right) \tag{8}$$

Among them, C represents the total number of categories, c represents the current category, T represents the threshold, t represents the current threshold, true positive (*TP*) represents a positive sample that is correctly predicted as a positive sample by the model; false positive (*FP*) represents a positive sample that is incorrectly predicted by the model. Negative samples of positive samples; false negative (*FN*) represents positive samples that are incorrectly predicted as negative samples by the model.

The segmentation speed of the algorithm was evaluated using Frames Per Second (FPS).

4.4. Comparison of Different Algorithms

In order to verify the effectiveness of the algorithm proposed in this article, it is compared with the Mask RCNN, YOLACT, PolarMask, SOLO, and SOLOv2 algorithms. The experimental results are shown in the Table 3. Compared with Mask RCNN, the *mAP*, AP_S, AP_M, and AP_L of the algorithm proposed in this article decreased by 3.2%, 8.4%, 3.1%, and 0.5%, respectively, but the FPS increased by 5.1; compared with YOLACT, the *mAP* of the algorithm proposed in this article, the AP_S, AP_M, and AP_L increased by 13.3%, 3.5%, 15%, and 19%, respectively, and FPS also increased by 2.4; compared with PolarMask, the *mAP*, AP_S, AP_M and AP_L of the algorithm proposed in this article increased by 8.9%, 2.6%, 9.7%, and 11.4%, respectively. FPS decreased by 0.6; compared with SOLO, the *mAP*, AP_S, AP_M, and AP_L of the algorithm proposed in this article increased by 6.8%, 2.7%, 9.6%, and 5.2%, respectively, and FPS increased by 3; Compared with SOLOv2, the *mAP*, AP_S, AP_M, and AP_L of the algorithm proposed in this article have increased by 4.2%, 2.6%, 6.9%, and 2.5%, respectively, and the FPS has increased by 1.7. It can be seen that, except for the two-stage instance segmentation algorithm Mask RCNN, which has a higher segmentation accuracy than the algorithm proposed in this article, the segmentation accuracy of the other single-stage instance segmentation algorithms is lower than the algorithm proposed in this article, especially in the segmentation of small objects. In terms of the segmentation speed, the FPS of the algorithm proposed in this article is higher than the other algorithms except PolarMask.

Table 3. Comparison of performance of different algorithms.

Model	*mAP*	AP_S	AP_M	AP_L	FPS
Mask RCNN_r50	30.8	12.5	28.3	50.6	3.8
YOLACT_r50	14.3	0.6	10.2	31.1	6.5
PolarMask_r50	18.7	1.5	15.5	38.7	9.5
SOLO_r50	20.8	1.4	15.6	44.9	5.9
SOLOv2_r50	23.4	1.5	18.3	47.6	7.2
Our	27.6	4.1	25.2	50.1	8.9

This article also compares the segmentation accuracy between the data set classes between the proposed algorithm and the original algorithm. The results are shown in the Figure 6. Among the eight categories, the algorithm proposed in this article improved by 9.9%, 9.9%, 2.3%, 8.1%, and 7.5%, respectively, compared with the original algorithm in the five categories of person, rider, truck, motorcycle, and bicycle. In the three categories of car, bus, and train, there is a slight decrease compared with the original algorithm.

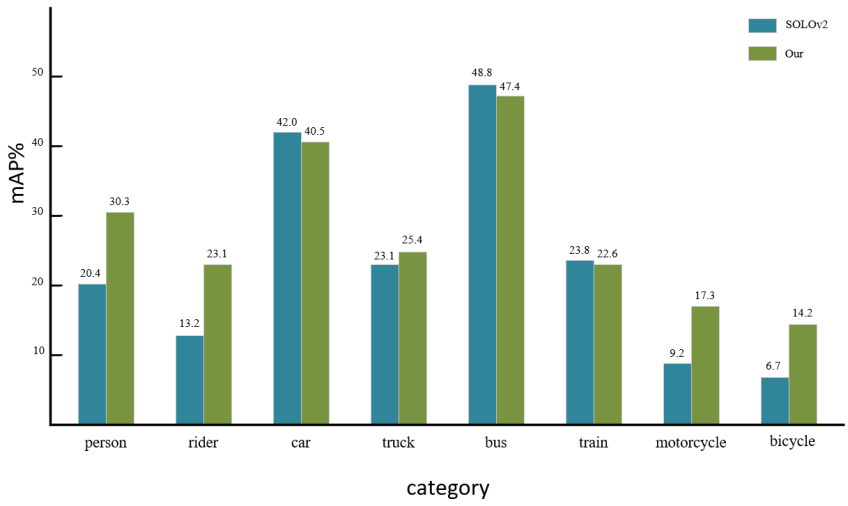

Figure 6. Segmentation accuracy of different categories.

4.5. Ablation Experiment

In order to further verify the effectiveness of the proposed structure, it is compared with the detection effect of the original module, and the results are shown in Table 4. After replacing the backbone network with VoVNetV2, the *mAP* dropped by 2.9% compared with the original model, but FPS increased by 2.9. Although the addition of VoVNetV2 reduced the model segmentation accuracy, it accelerated the model's processing speed. When the FPN is replaced by an FPG, the model's *mAP* increases by 1.9% and the FPS decreases by 0.3. After adding the BoT module, the *mAP* of the model increased by 2.2%, but the FPS decreased by 1.4. When the convolution is replaced by Ghost Conv, although the *mAP* value of the model only increases by 0.2%, the FPS increases by 1.1. Finally, after adding the CBAM module, the *mAP* increased by 4.2% and the FPS increased by 1.7 compared with the original model, which verified the effectiveness of the algorithm proposed in this article.

Table 4. Ablation Experiment Performance Comparison.

Model	Backbone		Neck		Modules			*mAP*	*mAR*	FPS
	Res50	VoVNetV2	FPN	FPG	BoT	Ghost Conv	CBAM			
SOLOv2	√		√					23.4	27.6	7.2
SOLOv2-V		√	√					20.5	25.2	10.1
SOLOv2-VG		√		√				22.4	26.7	9.8
SOLOv2-VGB		√		√	√			24.6	27.2	8.4
SOLOv2-VGBG		√		√	√	√		24.8	28.0	9.5
SOLOv2-VGBGC		√		√	√	√	√	27.6	30.3	8.9

To provide a more intuitive depiction of the improved algorithm's effectiveness, we conducted a visual comparison between the segmentation results of the improved algorithm and the original SOLOv2 algorithm, as illustrated in Figure 7. In Figure 7, from top to bottom, you can observe the original image, the SOLOv2 segmentation result image, and the improved segmentation result image. The red boxes indicate areas where SOLOv2 failed to detect or achieved incomplete segmentation compared to the improved algorithm. It's evident that, when compared to the original algorithm, the improved algorithm excels in terms of the segmentation accuracy and mask quality. Particularly with respect to smaller objects, the original algorithm often exhibits missed detections and incorrect segmentations in such cases.

Figure 7. Visualization results of improved algorithm and original algorithm on Cityscapes validation set. From top to bottom 1-Input RGB image; 2-SOLOv2; 3-Our. The red boxes indicate areas where SOLOv2 failed to detect or achieved incomplete segmentation compared to the improved algorithm.

5. Conclusions and Future Work

This paper proposes a new instance segmentation algorithm based on SOLOv2. The BoT module is added to the backbone network VoVNetV2 and the traditional convolution is replaced by Ghost Conv. The improved backbone network is used to replace the ResNet network to solve the problem of the poor edge contour segmentation quality of the model mask, and the problem of slow segmentation speed. A PFG is used to replace the FPN in the Neck part of the algorithm to solve the problem of the poor segmentation of small objects. The CBAM module is added to the Head part to obtain more detailed feature maps to improve the quality of the mask. The experimental results show that the improved algorithm has a *mAP* of 27.6% in the eight common road scene categories of the Cityscapes dataset, especially in the five categories with a relatively large number of small targets: person, rider, truck, motorcycle, and bicycle. The *mAP* improvement is particularly obvious, which is better than the current mainstream single-stage instance segmentation algorithm and only slightly weaker than the two-stage instance segmentation algorithm, but the segmentation speed is much faster than the two-stage instance segmentation algorithm. In the next step, we will use knowledge distillation, channel pruning and other methods to reduce the size of the algorithm while ensuring the accuracy of the algorithm, and fully integrate it into the embedded devices. At the same time, we will also consider how to effectively integrate various sensor data (such as RGB images, LiDAR, infrared images, etc.) into instance segmentation tasks to improve the understanding of complex scenes and the accuracy of object detection. This will help to better cope with various complex road scenarios.

Author Contributions: Conceptualization, J.P. and Q.Y.; methodology, Q.Y.; software, Q.Y.; validation, Q.Y., D.C. and H.Z.; formal analysis, D.C.; investigation, H.Z.; resources, J.P.; data curation, Q.Y.; writing—original draft preparation, Q.Y.; writing—review and editing, J.P.; visualization, D.C.; supervision, J.P.; project administration, J.P.; funding acquisition, J.P. All authors have read and agreed to the published version of the manuscript.

Funding: The authors are highly thankful to the National Natural Science Foundation of China (NO. 62063006), the Natural Science Foundation of Guangxi Province (NO. 2023GXNSFAA026025), to the Innovation Fund of Chinese Universities Industry-University-Research (ID: 2021RYC06005), to the Research Project for Young and Middle-aged Teachers in Guangxi Universities (ID: 2020KY15013), and to the Special research project of Hechi University (ID: 2021GCC028). This research was financially supported by the project of outstanding thousand young teachers' training in higher education institutions of Guangxi, Guangxi Colleges and Universities Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

Acknowledgments: The authors are highly thankful to the National Natural Science Foundation of China, to the Innovation Fund of Chinese Universities Industry-University-Research, to the Research Project for Young and Middle-aged Teachers in Guangxi Universities, and to the Special research project of Hechi University. This research was financially supported by the project of outstanding thousand young teachers' training in higher education institutions of Guangxi, Guangxi Colleges and Universities Key Laboratory of AI and Information Processing (Hechi University), Education Department of Guangxi Zhuang Autonomous Region.

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

References

1. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
2. 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.
3. 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.

4. Wang, X.; Zhang, R.; Kong, T.; Li, L.; Shen, C. Solov2: Dynamic and fast instance segmentation. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 17721–17732.
5. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 16519–16529.
6. Lee, Y.; Park, J. Centermask: Real-time anchor-free instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 7–10 September 2020; pp. 13906–13915.
7. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 7–10 September 2020; pp. 1580–1589.
8. Chen, K.; Cao, Y.; Loy, C.C.; Lin, D.; Feichtenhofer, C. Feature pyramid grids. *arXiv* **2020**, arXiv:2004.03580.
9. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
10. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous Detection and Segmentation. In Proceedings of the Computer Vision–ECCV 2014, 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Springer International Publishing: Berlin/Heidelberg, Germany, 2014; pp. 297–312.
11. O Pinheiro, P.O.; Collobert, R.; Dollár, P. Learning to segment object candidates. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1990–1998.
12. Pinheiro, P.O.; Lin, T.Y.; Collobert, R.; Dollár, P. Learning to Refine Object Segments. In Proceedings of the Computer Vision–ECCV 2016, 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 75–91.
13. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22 October 2017; pp. 2961–2969.
15. Han, K.; Rezende, R.S.; Ham, B.; Wong, K.Y.K.; Cho, M.; Schmid, C.; Ponce, J. Scnet: Learning semantic correspondence. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22 October 2017; pp. 1831–1840.
16. Zang, Y.; Huang, C.; Loy, C.C. Fasa: Feature augmentation and sampling adaptation for long-tailed instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 3457–3466.
17. Zhang, G.; Lu, X.; Tan, J.; Li, J.; Zhang, Z.; Li, Q.; Hu, X. Refinemask: Towards high-quality instance segmentation with fine-grained features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 6861–6869.
18. Hu, J.; Cao, L.; Lu, Y.; Zhang, S.; Wang, Y.; Li, K.; Huang, F.; Shao, L.; Ji, R. Istr: End-to-end instance segmentation with transformers. *arXiv* **2021**, arXiv:2105.00637.
19. Bolya, D.; Zhou, C.; Xiao, F.; Lee, Y.J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 29 October 2019; pp. 9157–9166.
20. Wang, X.; Zhao, K.; Zhang, R.; Ding, S.; Wang, Y.; Shen, W. Contrastmask: Contrastive learning to segment everything. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11604–11613.
21. 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/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 14–19 June 2020; pp. 12193–12202.
22. Wang, X.; Kong, T.; Shen, C.; Jiang, Y.; Li, L. Solo: Segmenting Objects by Locations. In Proceedings of the Computer Vision–ECCV 2020, 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XVIII 16; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 649–665.
23. Chen, X.; Girshick, R.; He, K.; Dollár, P. Tensormask: A foundation for dense object segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 29 October 2019; pp. 2061–2069.
24. 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/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 7–10 September 2020; pp. 8573–8581.
25. Kirillov, A.; Wu, Y.; He, K.; Girshick, R. Pointrend: Image segmentation as rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 28 July 2020; pp. 9799–9808.
26. Zhang, T.; Wei, S.; Ji, S. E2ec: An end-to-end contour-based method for high-quality high-speed instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 4443–4452.
27. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 6000–6010.

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