remote sensing

MDPI

Article

Efficient Dual-Branch Bottleneck Networks of Semantic Segmentation Based on CCD Camera

Jiehao Li [1,2], Yingpeng Dai [2,3,*], Xiaohang Su [4] and Weibin Wu [1]

[1] Key Laboratory of Key Technology on Agricultural Machine and Equipment, Ministry of Education, College of Engineering, South China Agricultural University, Guangzhou 510642, China
[2] State Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081, China
[3] Tobacco Research Institute of Chinese Academy of Agricultural Sciences, Qingdao 266101, China
[4] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China
* Correspondence: daiyingpeng@caas.cn

Abstract: This paper investigates a novel Efficient Dual-branch Bottleneck Network (EDBNet) to perform real-time semantic segmentation tasks on mobile robot systems based on CCD camera. To remedy the non-linear connection between the input and the output, a small-scale and shallow module called the Efficient Dual-branch Bottleneck (EDB) module is established. The EDB unit consists of two branches with different dilation rates, and each branch widens the non-linear layers. This module helps to simultaneously extract local and situational information while maintaining a minimal set of parameters. Moreover, the EDBNet, which is built on the EDB unit, is intended to enhance accuracy, inference speed, and parameter flexibility. It employs dilated convolution with a high dilation rate to increase the receptive field and three downsampling procedures to maintain feature maps with superior spatial resolution. Additionally, the EDBNet uses effective convolutions and compresses the network layer to reduce computational complexity, which is an efficient technique to capture a great deal of information while keeping a rapid computing speed. Finally, using the CamVid and Cityscapes datasets, we obtain Mean Intersection over Union (MIoU) results of 68.58 percent and 71.21 percent, respectively, with just 1.03 million parameters and faster performance on a single GTX 1070Ti card. These results also demonstrate the effectiveness of the practical mobile robot system.

Keywords: semantic segmentation; CCD camera; lightweight network; neural networks

Citation: Li, J.; Dai, Y.; Su, X.; Wu, W. Efficient Dual-Branch Bottleneck Networks of Semantic Segmentation Based on CCD Camera. *Remote Sens.* **2022**, *14*, 3925. https://doi.org/10.3390/rs14163925

Academic Editor: Ayman F. Habib

Received: 29 June 2022
Accepted: 9 August 2022
Published: 12 August 2022

1. Introduction

Recently, the technology of machine learning and neural networks for instrumentation and measurement has been widely discussed [1–6], especially for robotics fields [7–11]. For example, a multisensor-based algorithm for the outdoor environment using RGB-image neural networks for the mobile robot was considered in [12]. Furthermore, the framework of light field imaging and convolutional neural networks was designed in [13] for the semantic segmentation task. Machine vision, a core part of the mobile robot, plays an essential role in the self-control system and achieves the transformation from automation to intelligence [14,15]. As the main content of the machine vision, environmental perception provides essential environmental information for the subsequent tasks of the unmanned robot [16–18]. At present, Convolutional Neural Networks (CNNs) have been developing rapidly, and they have pushed the performance of machine vision to an unprecedented height [19]. Most semantic segmentation methods are based on CNNs. Semantic segmentation is used to divide the image into semantically meaningful parts, and this technique is known as pixel-wise classification. It is widely used in fields such as semantic segmentation [20–22], classification [23], detection [24], and so on. These applications have a

high demand for the precision and speed of semantic segmentation techniques [25,26]. In contrast, simultaneously considering the real-time performance and accuracy is the main challenge for engineering applications.

For semantic segmentation tasks, three problems are considered for CNNs: (1) Accuracy and inference speed dilemma. The complex structure produces high precision but takes more time during inference. The fundamental issue is how to balance the conflict between accuracy and inference speed. (2) The presence of items of different levels. Different environmental information, such as buildings, roads, pedestrians, and lane-line, has different sizes and shapes. This requires the network to understand the spatial features in multiple feature spaces and extract abundant multi-scale information. (3) A large number of parameters. Embedded devices on unmanned mobile robots often have limited storage. Therefore, the algorithm with few parameters is suitable for application on unmanned mobile devices.

Consequently, this study aims to construct an effective lightweight network (EDBNet) by building a small network that meets both operating speed and accuracy with few variables. With regard to the accuracy, feature maps with high spatial resolution, wide receptive fields, and an increase in non-linear layers can extract significant and substantial feature representations. Furthermore, the flexible structure with channel pruning, dropping certain layers, and applying depth-wise convolution, can reduce the computational complexity in terms of inference speed. The following items are the main contributions:

- A shallow EDB package is suggested to capture a wealth of information from two aspects in the situation of instrument detection on mobile robots. Firstly, this module consists of two branches jointly extracting local and contextual information. Secondly, two-dimensional standard convolution is divided into two parallel one-dimensional convolutions in each branch, widening the non-linear layers and strengthening the non-linear relationship.
- The mobile robot system can accurately and quickly draw conclusions while interpreting a scene. Studies using the CamVid and Cityscapes datasets demonstrate the efficacy of two real-world experiments on mobile robot systems, as well as the high accuracy and rapid inference speed that EDBNet accomplishes while creating a few parameters.

Rlated works are described in Section 2 along with some fundamental and cutting-edge techniques and an overview of current issues. Finally, the planned EDBNet is introduced, and Section 3 goes into great depth on how to construct each component. To confirm the usefulness of the EDBNet, we compare the experimental findings with those from other methods in Section 4. The conclusions and future planning are presented in Section 5.

2. Related Works

2.1. Multi-Scale Strategies

Generally, the existence of multi-scale objects increases the difficulty of semantic segmentation tasks. How to describe multi-scale objects to improve accuracy becomes a key problem. Many researchers often address this problem by extracting multi-scale features. Early methods such as FCN [27] and Unet [28] adopt serial skip connections to fuse the high-level and low-level information. Segnet [29] introduces Pooling Indices to record detailed information. The above three early methods reuse the low-level information to optimize object details such as boundary location and achieve fine segmentation results. However, successive downsampling operations result in the loss of lots of information, thus adversely affecting the segmentation results. Spatial pyramid pooling is a helpful strategy to extract multi-scale features. For instance, PSPNet [30] proposes the Pyramid Pooling Module to fuse features under four different pyramid scales. DeepLab [31] uses Atrous Spatial Pyramid Pooling (ASPP), which can capture objects from the image at multiple scales. RefineNet [32], a generic multi-path refinement network, is also a good network with multi-scale architecture. PSPNet, Deeplab, and RefineNet have complex structures to achieve high accuracy. However, a complex structure introduces high computational

complexity, and then we need to spend more time computing during the inference. As a result, these algorithms often provide poor inference speeds and are ineffective for real-time segmentation applications. Some retrieve not just multi-scale features but also offer rapid inference speeds for lightweight neural networks. In order to obtain information on various scales, ICNet employs three branches [33], and these data are then fused using CFF (Cascade Feature Fusion). Additionally, ICNet speeds up inference by lowering input resolution. DAB unit fully utilizes depthwise separable and dilated convolution while simultaneously extracting local and factual information. DABNet [34] layers DAB blocks to compromise inference speed and accuracy. In order to extract both high-level semantic information and minimal specific details, BiseNet [35] employs two branches to fuse the information using the Feature Fusion Module.

2.2. Lightweight Networks

Real-time applications have a strict requirement for inference speed. Unmanned mobile robot often has limited storage, which requires the algorithm to have a small number of parameters. Thus, a lightweight network, such as ENet [23], SqueezeNet [36], ShuffleNet [37], is an effective way to improve the inference speed and parameters in some applications. For instance, a lightweight network based on multi-scale context fusion [38] is proposed to improve the accuracy, inference speed, and model size. A lightweight deep neural network [39] is designed to exploit surgical instrument semantic segmentation that meets real-time conditions, accuracy, and robustness. Regarding practical engineering applications, balancing accuracy, inference speed, and model size is the main concern.

In the case of inference speed, four methods are often used to decrease the computational time during the inference. Firstly, reducing the resolution and pruning channels are common and effective methods, which greatly reduce the calculation and thus decrease the computational time. Secondly, lots of lightweight models adopt effective convolution methods such as depthwise separable convolution [22], one-dimensional convolution [23], and group convolution [40]. Those convolution methods change the computational strategy of the convolution and dramatically reduce the computational complexity, thus accelerating the inference speed. Thirdly, some algorithms abandon some convolutional/downsampling layers to obtain a shallow/tight structure. Fourthly, Binary Ensemble Neural Networks [41] replace most arithmetic operations with the bit-wise operation, which substantially accelerates the inference speed. The above methods accelerate the inference speed effectively. Nevertheless, lowering the resolution, adopting a good convolution technique, and using BNNs would result in significant spatial information loss. The non-linear connection between the input and the output will deteriorate if channels are pruned and layers are abandoned, and those operations also affect precision.

In terms of parameters, three methods are mainly used to decrease the parameters. Firstly, effective convolution methods not only accelerate the inference speed but also decrease parameters. For example, MobileNet [42] replaces the Standard Convolution with the Depthwise Separable Convolution, and the parameters are reduced by about 8/9. Secondly, model compression is an effective method. For example, Xception [43] simplifies the Inception module [44] and uses the group convolution to extract the features. GhostNet [45] and SqueezeNet [36] design the compression module named "Fire module" and "Ghost module", respectively. Thirdly, some models change the type of parameters. The author transforms the data type of BNNs from float-32 to 1-bit, which decreases the parameters substantially.

3. Proposed Network

We go into further detail on the design of the EDBNet in this part. The EDB module is first described in detail as the main element of EDBNet, as seen in Figure 1. The design of the EDBNet architecture is then discussed, including the network depth, the staged extraction of features, and the arrangement of the EDB module. Finally, Figure 2 illustrates the general structure of the proposed EDBNet.

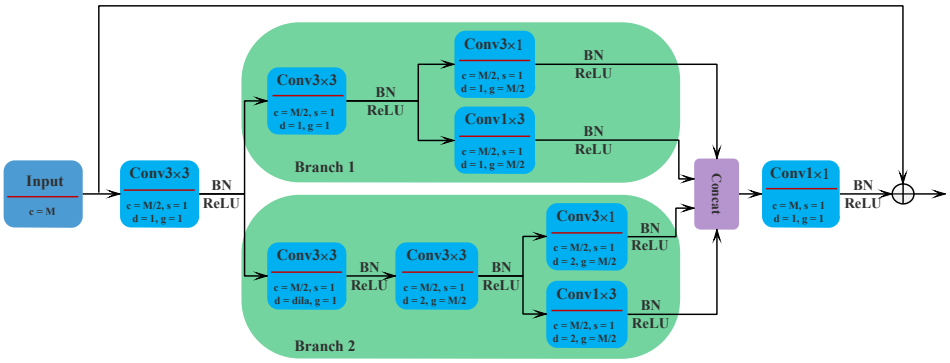


Figure 1. EDB module. (BN: Batch normalization; ReLU: Activation function, rectified linear unit; c: Channel number; colorblueM: A positive number divisible by 2; s: Convolution stride; d: Dilation rate; g: Group number).

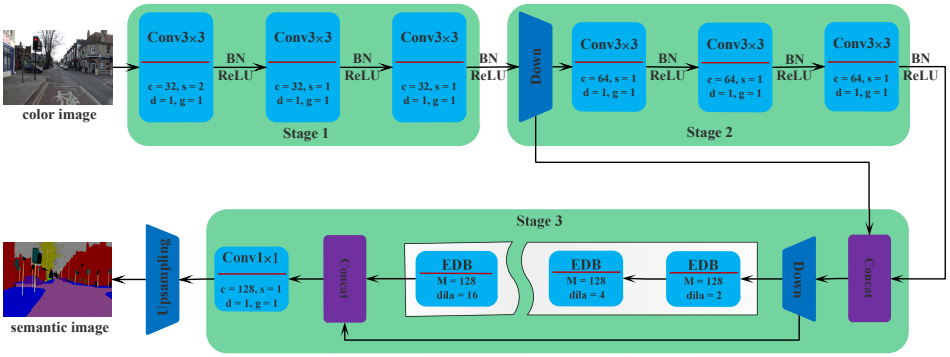


Figure 2. Structure of EDBNet. (c: Channel number; s: Convolution stride; d: Dilation rate; g: Group number).

3.1. Edb Module

Bottleneck structure. Inspired by the Inception module in GoogLeNet [46], a similar bottleneck structure is applied to the EDB module. To extract local feature information and contextual information while preserving the small-scale structure, the EDB module utilizes dual branches when dealing with complete feature information. As seen in Figure 1, the input is first filtered using a 3×3 conventional convolutional structure, and then a dual-branch structure is employed to extract both local and contextual feature information. Then, new features are created by merging the information from the original features using 1×1 standard convolution. The EDB module's complexity is adjusted dynamically using residual connections. Overall, the EDB module can extract multi-scale information. Multiple depthwise convolutions could effectively decrease the parameters and computational time.

Dual-branch structure. The dual-branch structure, which makes up the majority of the EDB module, is crucial to the combined collection of local and contextual information. As shown in Figure 1, two branches make up this dual-branch arrangement. The upper one, named "branch 1", mainly extracts the local information to describe the detail of the object. It consists of a 3×3 standard convolution and two depthwise asymmetric convolutions. The first is to use the 3×3 standard convolutions to filter the input. Then, considering that asymmetric convolution can reduce the computational complexity and parameters, a 1×3 convolution and a 3×1 convolution are applied in our EDB. Unlike factorized convolutions in ERFNet [47] and DABNet [34], the 1×3 convolution and the 3×1 convolution are parallel rather than cascade, which can widen the non-linear layers and strengthen the non-linear relationship between the input and the output while maintaining the function of cascade asymmetric convolutions. The bottom one, named "branch 2", is used to extract the contextual information. It consists of a dilated convolution, a depthwise convolution, and two depthwise asymmetric convolutions. The dilated convolution is used to enlarge the re-

ceptive field. Compared with the large convolutional kernel, it produces fewer parameters and quicker computational speed while enlarging the receptive field. Then, a depthwise convolution and two parallel depthwise asymmetric convolutions are used to widen and deepen the non-linear layers, benefiting from extracting sufficient contextual information. Overall, this dual-branch structure strengthens the non-linear relationship to extract as much multi-scale feature information as possible. Meanwhile, effective convolutions such as dilated convolution and depthwise convolution decrease parameters and accelerate computational speed.

3.2. EDBNet Architecture Design

We will go into more detail about the EDBNet's architecture in this part, including how to strike a balance between accuracy, inference speed, and parameters; Figure 2 depicts the EDBNet framework.

Since repeated downsampling operations result in feature maps with significantly reduced spatial resolution and produce sparse features in the last layer [48,49], the EDBNet structure will lose a significant amount of spatial information, making semantic segmentation tasks more difficult. In order to achieve a tighter architecture, the EDBNet uses three downsampling processes. This technique generates high spatial resolution feature maps and preserves adequate spatial feature information, but its receptive field is insufficient to include the objects. In order to solve this issue, we thus introduce dilated convolution to extend the receptive field.

We categorize the EDBNet into three phases based on the variations in spatial resolution. Three 3×3 conventional convolution processes make up the first stage's extraction of the original feature data. In this case, the downsampling procedure uses a 3×3 convolution with stride 2 as the input. Compared to max-pooling, keeping a lot of information is advantageous. On the other hand, it speeds up inference by reducing resolution. The second stage consists of three 3×3 conventional convolutional networks and a downsampling operation. This step likewise harvests sufficient image features, similar to the first stage. Following the second stage, we combine the feature data from the downsampling operation with the convolution layers from the final layer of the second section and input them into the third stage. Four EDB modules and a downsampling process make up the third stage. Thus, the EDBNet acquires sufficient receptive field to cover the object and collect multi-scale feature data by progressively increasing the dilation rates in the EDB units. In the end, we combine that data and submit it to the classifier. After the bilinear interpolation, the output pictures restore the same resolution as the original image.

In conclusion, the EDBNet can significantly increase speed, accuracy, and parameter adaptability. Only a few downsampling processes result in feature maps with great spatial resolution and additional spatial data. The non-linear layers are widened by the EDB module, which simultaneously gathers local and contextual information. The receptive field may be increased, and multi-scale feature data can be extracted using dilated convolutions in EDB modules with various dilation rates. These processes frequently result in great accuracy. The computational complexity is decreased by efficient convolutions like depthwise and 1D convolution. A compact and shallow network topology is also advantageous, which speeds up calculation during inference. The two processes mentioned above help to increase the speed of inference.

4. Experiments

In this part, the accuracy, speed of inference, and parameters of the EDBNet are assessed using the CamVid dataset and the Cityscapes dataset [50]. The implementation process is first outlined in detail. The EDBNet's usefulness is then shown through a series of tests we design. Then, we assess the accuracy, speed of inference, and parameters of EDBNet compared to big models and lightweight neural networks.

4.1. Implementation Details

Trials are conducted on the computer equipped with a GTX 1070Ti GPU and Pytorch. To improve the training procedure, we use an Adam optimizer. The epoch is set to 1000, while the batch size is 8. The loss function used to determine how much the forecasted output (Y_{ij}) differs from the label (T_{ij}) in this case is the cross-entropy.

$$J = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log(p_{ij}) \tag{1}$$

where N is the number of test samples. C is the number of classifications. p_{ij} is the predicted probability and is dependent on the output results of EDBNet. y_{ij} is the indicator variable and can be expressed by:

$$y_{ij} = \begin{cases} 0, & Y_{ij} \neq T_{ij} \\ 1, & Y_{ij} = T_{ij} \end{cases} \tag{2}$$

In the early stage of training, it needs to learn quickly with a large learning rate. In the later stage of training, it needs a small learning rate to find the optimal solution. The pow function is used to dynamically adjust the learning rate, as follows:

$$\eta = baseLR(1 - \frac{cur\_epoch}{max\_epoch})^{0.9} \tag{3}$$

where *baseLR* denotes the original learning rate. *cur_epoch* represents the current number of epoch. *max_epoch* is the maximum number of iterations.

There are 367 training images, 101 validation images, 233 testing images, and 11 semantic categories in the CamVid dataset. Additionally, the Cityscapes dataset includes 1525 test photos, 500 validation images, and 2975 training images that are divided into 19 semantic categories.

4.2. Ablation Experiment

We conduct ablation experiments to verify the effectiveness of the proposed algorithm. The EDB module plays an important role in real-time semantic segmentation tasks. Here, the EDBNet structure is used as the baseline, and we verify the effectiveness of the EDB module from three aspects. When without branches 1 or 2, the networks are named "EDBNet without branch1" and "EDBNet without branch 2", respectively. "EDBNet with extended Stage 2" represents that EDB modules in the second stage replaces standard convolutions. "EDBNet with extended Stage 3" represents that six EDB modules are used in the third stage with the results of Table 1.

When all EDB modules are replaced with conventional convolutions, according to the EDBNet design, the MIoU drops from 68.58 to 58.68 percent. This demonstrates how accurately the EDB module is intended to work. The correlation between pixels, high-level feature information that indicates an object's characteristic, is described by contextual information. Incorrect object categorization will result in a lack of contextual information. This substantially impacts the accuracy of semantic segmentation.

The related structure has a faster inference time and fewer variables when the EDB module forgoes branches 1 or 2, but accuracy suffers since the EDB module cannot extract contextual information. Branch 2 yields feature maps with great spatial resolution when abandoned, according to Table 1. The accuracy is only 61.77 percent MIoU, and the receptive field is insufficient to cover the objects. Branch 2 employs a high dilation rate to expand the receptive area when branch 1 is abandoned, increasing accuracy from 61.77 percent to 66.7 percent. In addition, when the dilation rate in Stage 3 is fixed, the ability of the EDB module to extract multi-scale information is limited. So EDBNet with a fixed dilation rate achieves 67.26% accuracy, which is less than 1% compared to EDBNet with a gradually increased dilation rate. Compared with standard convolution, dilated convolution, or depthwise separable convolution, the EDB module can extract complete information and better describe objects. The accuracy could be improved when standard,

dilated, or depthwise separable convolution is replaced with an EDB module. However, more time is spent during the inference. Too many EDB modules will slow down the inference speed. It is crucial to design the amount of EDB modules such that accuracy and inference speed may be traded off. Four EDB modules in the third stage, developed after extensive testing, improve the link between inference speed and accuracy in EDBNet.

Table 1. Ablation Experiments on CamVid dataset.

Models	MIoU(%)	FPS	Parameters
EDBNet without Branch 2	61.77	78.13	0.80 M
EDBNet without Branch 1	66.70	68.49	0.81 M
EDBNet with extended Stage 2	68.45	46.08	1.40 M
EDBNet with extended Stage 3	67.88	35.97	1.06 M
EDBNet with fixed dilation rate	67.26	61.73	1.03 M
EDBNet(ours)	68.58	61.73	1.03 M

4.3. Performance Evaluation of the Accuracy and Parameters

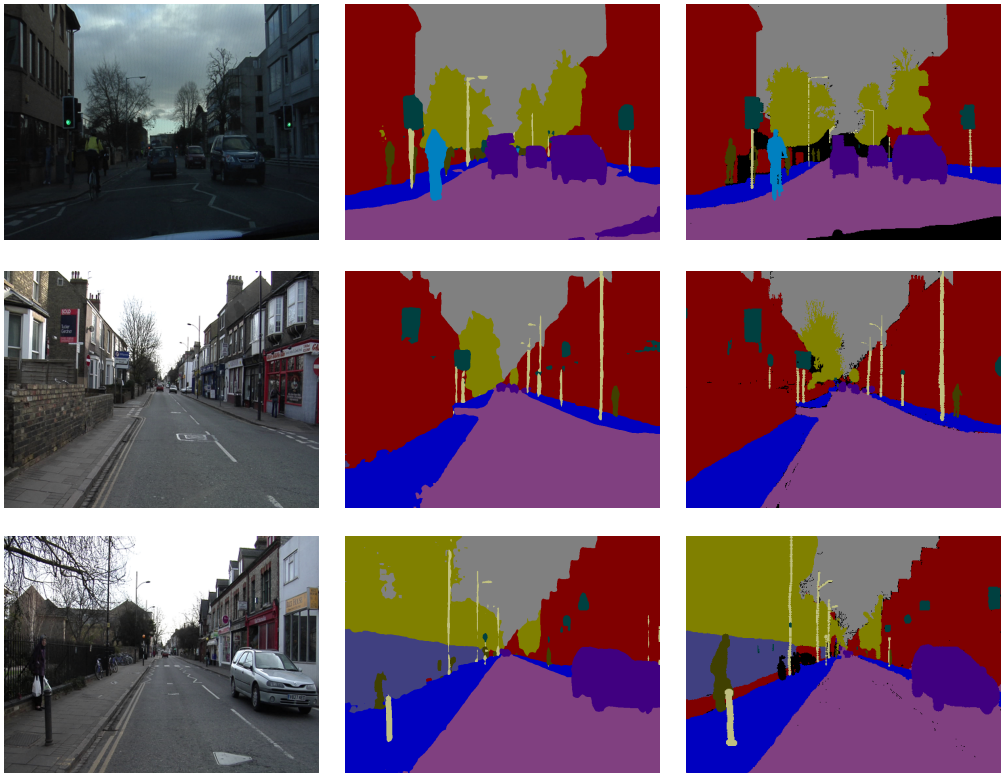
The EDBNet's network design performance is examined using the CamVid and Cityscapes datasets. In terms of accuracy and parameters, we compare the experimental findings of our EDBNet with those of commonly used and cutting-edge networks. Table 2 displays the findings. Refer to Figures 3 and 4.

Large models. In addition to cutting-edge algorithms like PSPNet, Deeplab, SVCNet, and CGBNet, large models include the frequently used FCN, Segnet, Dilation 10, and DeconvNet. The framework with iterative downsampling processes has poor spatial resolution feature maps and loses a great deal of spatial information. This negatively impacts semantic segmentation tasks. To address the issue, FCN-8s offers skip connection, which fuses the low-level and high-level characteristics. Segnet introduces the pooling indices to describe the location information accurately. DeconvNet adds unspooling and deconvolution layers to restore details such as location and shape. Those operations mitigate the above problem, but their role is limited. Large models that are often used typically perform poorly in terms of accuracy. Large models using cutting-edge algorithms typically create a new sophisticated module to extract all the information and therefore achieve great accuracy. For example, DeepLab abandons the last downsampling layers to create feature maps with high spatial resolution and adds the ASPP module to extract the multi-scale sort. These procedures can increase accuracy. PSPNet creates PPM, a multi-scale module that can extract distinct features in multiple feature spaces, to increase detailed and semantic information representation. The inference speed of big state-of-the-art models is drastically slowed, making them unsuitable for real-time semantic segmentation tasks and having good accuracy overall. The suggested technique clearly outperforms commonly used big models in terms of accuracy, inference speed, and parameters, as shown in Table 2. Large, cutting-edge models yield great accuracy; on the Cityscapes dataset, Deeplab, SVCNet, and CGBNet each reaches more than 80 MIoU. However, the inference speed is severely compromised by such techniques. Compared to big state-of-the-art models, the proposed technique better balances accuracy and inference speed.

Table 2. Testing Results of Accuracy MIoU).

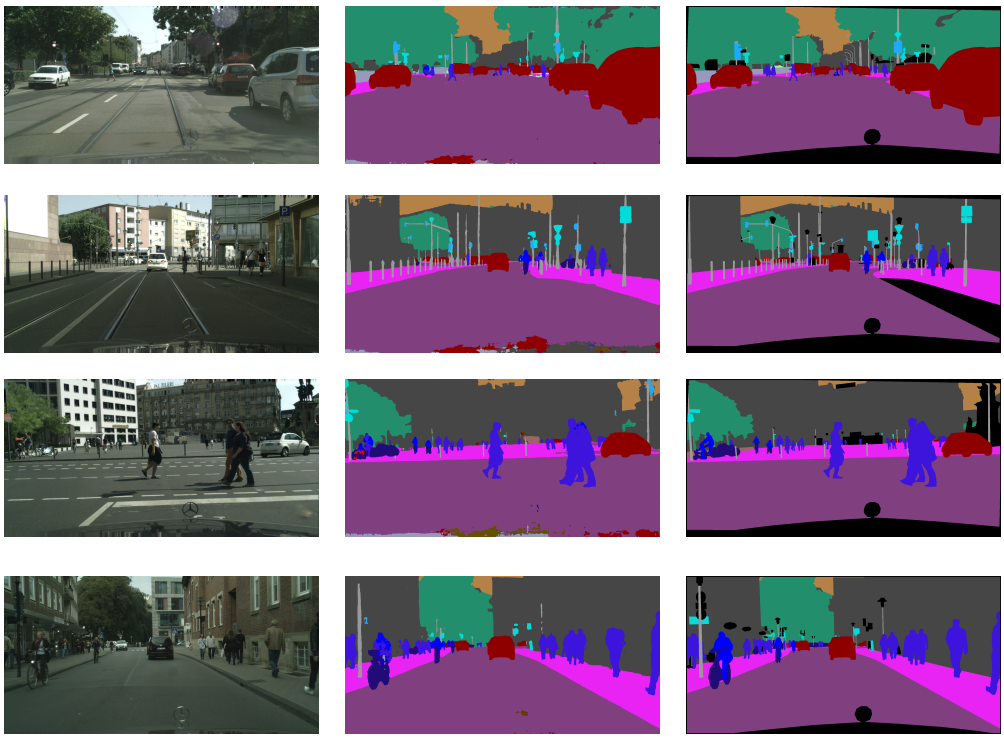
Models		GTX 1070Ti		Parameters
		CamVid	Cityscapes	
Large Models	FCN-8s [27]	57.0	65.3	134.5 M
	SegNet [29]	60.1	-	29.45 M
	Dilation10 [51]	65.3	67.1	140.5 M
	PSPNet [30]	69.1	78.4	65.7 M
	DeepLab v3 [31]	-	81.3	>30 M
	SVCNet [52]	75.4	81.0	-
	CGBNet [53]	-	81.2	-
Lightweight Models	ENet [23]	51.3	58.3	0.37 M
	ICNet [33]	67.1	69.5	26.6 M
	BiseNet [35]	65.5	68.4	12.5 M
	ERFNet [47]	-	68.0	2.1 M
	ESPNet V2 [54]	-	66.2	<10 M
	FSSNet [55]	58.6	58.8	0.2 M
	DABNet [34]	66.4	70.1	0.76 M
	DFANet [56]	64.7	70.3	7.8 M
	BiseNet v2 [57]	72.4	72.6	49 M
EDBNet (proposed)		68.6	71.2	1.03 M

Lightweight models. ENet is renowned for its fast inference speed and small parameter requirements. In order to achieve a compact structure while simultaneously losing accuracy, the last downsampling layers are abandoned. On the CamVid and Cityscapes datasets, ENet only achieves 51.3 and 58.3 percent MIoU, respectively. Lightweight algorithms like ICNet, Bisenet, DABNet, and DFANet emphasize the trade-off between accuracy and inference speed due to the demands of real-time applications. These algorithms provide an effective module that extracts all information rapidly. As an illustration, DABNet creates a DAB module that quickly pulls local and contextual data. BiseNet creates a straightforward dual-branch unit to capture both high-level semantic and low-level detail information. Those methods comprehensively balance computational complexity and structural complexity. From Table 2, ENet and FSSNet stress the inference speed and parameters, their accuracy less than most widely used large models. Other lightweight structures achieve more than 68 percent MIoU while maintaining quick inference speed. Even DABNet and DFANet produce quicker inference speeds than ENet and FSSNet. Our proposed EDBNet achieves 68.6% MIoU and 71.2% MIoU with 1.03 M parameter and quick inference speed on the CamVid and Cityscapes datasets. Compared with lightweight models, EDBNet produces competitive results.



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

Figure 3. Semantic segmentation results on the CamVid. (**a**) Original images. (**b**) EDBNet. (**c**) Ground truth.



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

Figure 4. Semantic segmentation results on the Cityscapes. (**a**) Original images. (**b**) EDBNet. (**c**) Ground truth.

4.4. Performance Evaluation of the Inference Speed on a Single GTX 1070Ti Card

In the original studies, certain lightweight networks are run on various computing devices. For instance, Titan X, GTX 1080Ti, and GTX 1070Ti are used to test ENet, DABNet, and EDBNet (planned), respectively. Since the computing equipment impacts only the inference speed, we assess the inference speed under identical circumstances to make a fair comparison. On a single GTX 1070Ti card, we compare our method to cutting-edge algorithms with rapid inference speeds, including ENet, DABNet, and BiseNet. Table 3 displays the comparative findings.

Table 3. Testing Results of Inference Speed.

Models	**512 × 1024**	
	ms	**fps**
SegNet	80.6	12.4
ENet	18.2	54.9
ICNet	15.0	67.2
DABNet	14.6	68.5
ESPNet	12.7	78.7
DFANet	12.6	79.4
BiseNet v2	9.7	103.1
EDBNet (proposed)	12.3	81.3

In contrast to conventional convolution, the EDB function includes two branches to extract additional feature information. The complicated structure requires more time to deduce the non-linear relationship between the input and output. We lower the computational complexity on two fronts to speed up inference while obtaining a wealth of feature information. (1) Regarding the EDB module design, some standard convolutions are replaced with depthwise convolutions. (2) Some last layers are abandoned in terms of the EDBNet architecture design. In addition, because the main role of the EDB module is to fuse multi-scale feature information, the initial features are extracted by standard convolutions, and EDB modules are used in the last stage. From Table 3, EDBNet has a quicker inference speed than Segnet, ENet, ICNet, and DABNet, while achieving better accuracy. Compared with ESPNet and DFANet, EDBNet produces a similar inference speed. On the other hand, BiseNet v2 achieves the quickest inference speed compared to the lightweight networks listed in Table 3.

4.5. Results on a Practical Mobile Robot in the Real World

In order to evaluate the semantic performance of the practical mobile robot system, two robot platforms are used to operate the real-time segmentation task. Firstly, for the standard four-wheel drive mobile robot platform, the embedded device with Nvidia Jetson AGX Xavier is used to run the EDBNet, and the CCD camera is applied to the image acquisition equipment. The environment perception system comprises RS-LIDAR-32 Lidar, XW-GI7660 integrated navigation, and a CCD camera. The graphics card memory of the robot system is 8 GB, as well as the six wheel-legged robot. For fast segmentation of road information, we only set two categories so that the mobile robot can quickly find the way and effectively avoid obstacles simultaneously.

The semantic segmentation performance of road edge information using the EDBNet method on the mobile robot is shown in Figure 5. In the case of different conditions such as direct sunlight, back-lighting, and shadow, the EDBNet method can obtain a better segment of road regions from the background. In other words, if the mobile robot quickly finds a passable area through the result of semantic segmentation, the network result has effective processing speed and segmentation effect. When the embedded device with Nvidia Jetson AGX Xavier is at peak processing speed, the real-time operating speed of the EDBNet can perform tasks under 30 FPS on the mobile robot.

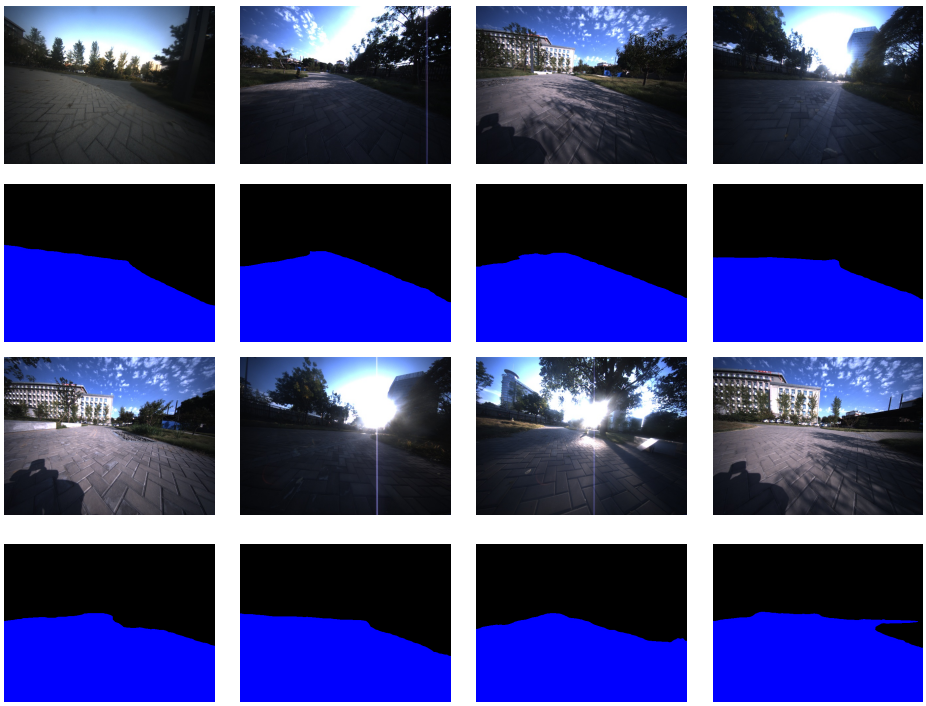


Figure 5. Semantic segmentation results of road edge information on the campus environment.

Furthermore, the six wheel-legged robot [58,59] is utilized to evaluate the proposed algorithm in the real-world environment, as shown in Figure 6. There are the visual perception system, central control system, motion control system, and energy system in the robot system. The mobility system consists of six-wheeled Elmo servo motors with the VxWorks environment and 36 GSM20-1202 electric cylinders with a 300 mm stroke. The energy system includes 24 V storage and 300 V power batteries. The robot is equipped with attitude, force, and motor encoder sensors for real-time feedback control. As a bonus, the visual system integrates navigation (XW-GI7660), Lidia (Velodyne), an infrared sensor (Gobi640), and a CCD sensor (Blackfly). The image engine is a TX2 board, and the control environment is ROS and Ubuntu. The CCD camera is employed as the image acquisition apparatus, and the Nvidia Jetson AGX Xavier is used to control the EDBNet.

The semantic segmentation results are shown in Figure 7. Firstly, (b) EDBNet, (c) DABNet, and (d) ENet are trained on the Cityscapes dataset. Then, the trained results of ENet, DABNet, and EDBNet are deployed on the six wheel-legged robot for the semantic segmentation task. In the real-world environment, the environment information includes roads, trees, buildings, cars, sidewalks, and so on. It can be seen in Figure 7 that three detailed comparison results can be drawn, including three pedestrians, bicycles, and cars. The proposed EDBNet algorithm can identify three pedestrians without a box behind them. In contrast, the ENet network basically loses the semantic information of the box, and the segmentation of the three pedestrians sticks together without clearly separating the three characters. Furthermore, in the case of the semantic segmentation task for bicycle groups and telephone poles, the EDBNet algorithm can effectively identify all bicycles and telephone poles in the image, while the DABNet and ENet networks lose part of the object information, and the ENet network does not express the pole information well. At the same time, for the identification task of car groups, the proposed algorithm can also complete the task well, while the other two methods have information loss in identification accuracy. In summary, compared with more advanced algorithms, the proposed algorithm has effective application performance in mobile robot systems, and it is an optional method that can be generalized and applied to engineering applications.

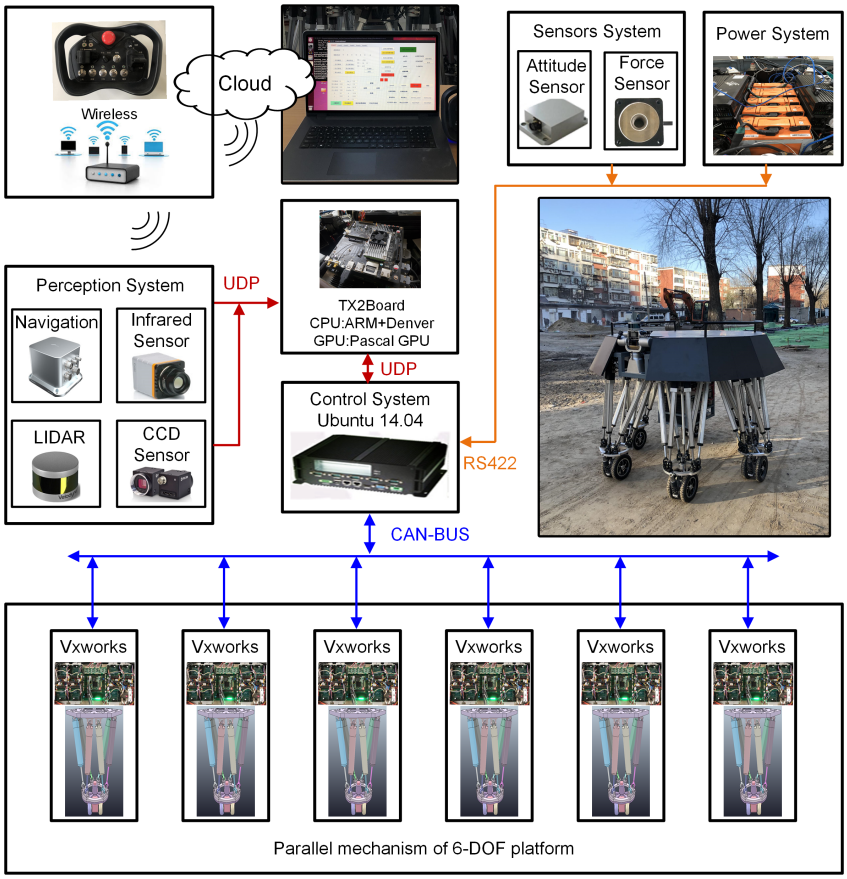


Figure 6. Hardware and software architecture of the robotic distributed system.

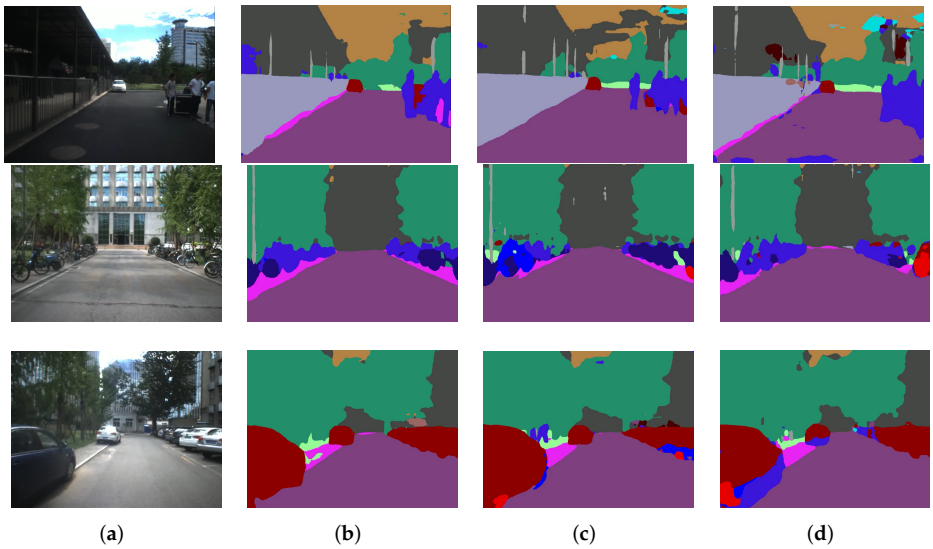


Figure 7. Semantic segmentation results in the real-world environment using the six wheel-legged mobile robot. (**a**) Original images. (**b**) EDBNet (proposed). (**c**) DABNet. (**d**) ENet.

5. Conclusions

This research attempts to enhance the real-time performance and semantic segmentation accuracy acceptable for robotic systems in engineering fields. A novel Efficient Dual-branch Bottleneck Network (EDBNet) is presented for real-time semantic segmentation tasks in intricate street sceneries. With minimal arguments, this module assists in concurrently extracting local and contextual information. Dual-branch structure, dilated

convolutions, and group convolutions are all fully utilized by the EDB module. While preserving a few settings, it may jointly extract local and contextual data. By substituting two parallel one-dimensional convolutions for one two-dimensional convolution to further eliminate feature information, the non-linear layers are widened, and the non-linear connection is strengthened. The accuracy, speed, and parameters of the EDBNet are intended to be highly effective. EDBNet delivers greater accuracy and has fewer parameters than other shallow networks, according to trials on the Cityscapes and CamVid datasets. Despite the items' various sizes, EDBNet could still characterize them more accurately. EDBNet delivers a faster inference performance on a single GTX 1070Ti when compared to DABNet. These results demonstrate the suitability of EDBNet for real-time semantic segmentation tasks on robot image processing. In the future, we will consider more applications on artificial intelligence frameworks to enhance the semantic segmentation performance.

Author Contributions: Conceptualization, J.L. and Y.D.; Data curation, J.L. and X.S.; Investigation, Y.D.; Project administration, W.W.; Software, Y.D.; Supervision, W.W.; Visualization, X.S.; Writing—original draft, Y.D.; Writing—review & editing, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Key Research and Development Program of Guangdong Province (2021B0101220003), Guangdong Laboratory for Lingnan Modern Agriculture Project (NT2021009), and National Key Research and Development Program of China under Grant 2019YFC1511401.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hasheminasab, S.M.; Zhou, T.; Lin, Y.C.; Habib, A. Linear Feature-Based Triangulation for Large-Scale Orthophoto Generation Over Mechanized Agricultural Fields. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5621718. [CrossRef]
2. Lin, Y.C.; Shao, J.; Shin, S.Y.; Saka, Z.; Joseph, M.; Manish, R.; Fei, S.; Habib, A. Comparative Analysis of Multi-Platform, Multi-Resolution, Multi-Temporal LiDAR Data for Forest Inventory. *Remote Sens.* **2022**, *14*, 649. [CrossRef]
3. Lin, Y.C.; Zhou, T.; Wang, T.; Crawford, M.; Habib, A. New Orthophoto Generation Strategies from UAV and Ground Remote Sensing Platforms for High-Throughput Phenotyping. *Remote Sens.* **2021**, *13*, 860. [CrossRef]
4. Chen, X.; Li, Y.; Fan, J.; Wang, R. RGAM: A novel network architecture for 3D point cloud semantic segmentation in indoor scenes. *Inf. Sci.* **2021**, *571*, 87–103. [CrossRef]
5. Tang, X.; Tu, W.; Li, K.; Cheng, J. DFFNet: An IoT-perceptive dual feature fusion network for general real-time semantic segmentation. *Inf. Sci.* **2021**, *565*, 326–343. [CrossRef]
6. He, J.; Gu, H.; Wang, Z. Multi-instance multi-label learning based on Gaussian process with application to visual mobile robot navigation. *Inf. Sci.* **2012**, *190*, 162–177. [CrossRef]
7. Li, L.; Dong, Z.; Yang, T.; Cao, H. Deep Learning-Based Automatic Monitoring Method for Grain Quantity Change in Warehouse Using Semantic Segmentation. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3056743.
8. Su, X.; Philip Chen, C.; Liu, Z. Adaptive fuzzy control for uncertain nonlinear systems subject to full state constraints and actuator faults. *Inf. Sci.* **2021**, *581*, 553–566. [CrossRef]
9. Peng, G.; Chen, C.L.P.; Yang, C. Neural Networks Enhanced Optimal Admittance Control of Robot-Environment Interaction Using Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–11.
10. Yang, C.; Peng, G.; Cheng, L.; Na, J.; Li, Z. Force Sensorless Admittance Control for Teleoperation of Uncertain Robot Manipulator Using Neural Networks. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 3282–3292. [CrossRef]
11. Li, J.; Li, R.; Li, J.; Wang, J.; Wu, Q.; Liu, X. Dual-view 3D object recognition and detection via Lidar point cloud and camera image. *Robot. Auton. Syst.* **2022**, *150*, 103999. [CrossRef]
12. Qiu, Z.; Zhuang, Y.; Yan, F.; Hu, H.; Wang, W. RGB-DI Images and Full Convolution Neural Network-Based Outdoor Scene Understanding for Mobile Robots. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 27–37.
13. Jia, C.; Shi, F.; Zhao, M.; Zhang, Y.; Cheng, X.; Wang, M.; Chen, S. Semantic Segmentation with Light Field Imaging and Convolutional Neural Networks. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3115204.
14. Li, J.; Wang, J.; Peng, H.; Hu, Y.; Su, H. Fuzzy-Torque Approximation-Enhanced Sliding Mode Control for Lateral Stability of Mobile Robot. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 2491–2500. [CrossRef]
15. Yang, C.; Wu, H.; Li, Z.; He, W.; Wang, N.; Su, C.Y. Mind Control of a Robotic Arm With Visual Fusion Technology. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3822–3830. [CrossRef]

16. Li, J.; Wang, J.; Peng, H.; Zhang, L.; Hu, Y.; Su, H. Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction. *Neurocomputing* **2020**, *410*, 342–353. [CrossRef]
17. Li, J.; Wang, J.; Wang, S.; Yang, C. Human-robot skill transmission for mobile robot via learning by demonstration. *Neural Comput. Appl.* **2021**, 1–11. [CrossRef] [PubMed]
18. Li, J.; Zhang, X.; Li, J.; Liu, Y.; Wang, J. Building and optimization of 3D semantic map based on Lidar and camera fusion. *Neurocomputing* **2020**, *409*, 394–407. [CrossRef]
19. LeCun, Y. Back propagation applied to handwritten zip code recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
20. Siam, M.; Gamal, M.; Abdel-Razek, M.; Yogamani, S.; Jägersand, M. Real-time semantic segmentation comparative study. In Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1603–1607.
21. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
22. Howard, A.G.; Zhu, M.L.; Chen, B.; Kalenichenko, D.; Wang, W.J.; Weyang, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Networks for Mobile Vision Application. *arXiv* **2017**, arXiv:1704.04861.
23. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. ENet: A deep neural network architecture for real-time semantic segmentation. *arXiv* **2016**, arXiv:1606.02147.
24. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
25. Dai, Y.; Wang, J.; Li, J.; Li, J. MDRNet: A lightweight network for real-time semantic segmentation in street scenes. *Assem. Autom.* **2021**, *41*, 725–733. [CrossRef]
26. Li, J.; Qin, H.; Wang, J.; Li, J. OpenStreetMap-based autonomous navigation for the four wheel-legged robot via 3D-Lidar and CCD camera. *IEEE Trans. Ind. Electron.* **2022**, *69*, 2708–2717. [CrossRef]
27. Shelhamer, E.; Long, J.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 640–651. [CrossRef]
28. Ronneberger, O.; Fischer, P.; Brox, T. U-Net:Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; pp. 234–241.
29. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]
30. Zhao, H.S.; Shi, J.P.; Qi, X.J.; Wang, X.G.; Jia, J.Y. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6230–6239.
31. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
32. Lin, G.S.; Milan, A.; Shen, C.; Reid, I. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5168–5177.
33. Zhao, H.S.; Qi, X.J.; Shen, X.Y.; Shi, J.P.; Jia, J.Y. ICNet for Real-Time Semantic Segmentation on High-Resolution Images. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 418–434.
34. Li, G.; Yun, I.; Kim, J.; Kim, J. DABNet: Depth-wise Asymmetric Bottleneck for Real-time Semantic Segmentation. In Proceedings of the 30th British Machine Vision Conference (BMVC), Cardiff, UK, 9–12 September 2019; pp. 418–434.
35. Yu, C.Q.; Wang, J.B.; Peng, C.; Gao, C.X.; Yu, G.; Sang, N. BiseNet: Bilateral Segmentation Network for Real-time Semantic Segmentation. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 334–349.
36. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. In Proceedings of the 5th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
37. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
38. Gao, G.; Xu, G.; Yu, Y.; Xie, J.; Yang, J.; Yue, D. MSCFNet: A Lightweight Network With Multi-Scale Context Fusion for Real-Time Semantic Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–11. [CrossRef]
39. Sun, Y.; Pan, B.; Fu, Y. Lightweight Deep Neural Network for Real-Time Instrument Semantic Segmentation in Robot Assisted Minimally Invasive Surgery. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3870–3877. [CrossRef]
40. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
41. Zhu, S.L.; Dong, X.; Su, H. Binary Ensemble Neural Network: More Bits per Network or More Networks per Bit? In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4918–4927.

42. Sandler, M.; Howard, A.; Zhu, M.L.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
43. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
44. Szegedy, C.; Vanhoucke, V.; Loffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
45. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C. GhostNet: More Features From Cheap Operations. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1577–1586.
46. Szegedy, C.; Liu, W.; Jia, Y.Q.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
47. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 263–272. [CrossRef]
48. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
49. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1440–1448.
50. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
51. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the 2015 International Conference on Learning Representations (ICLR), Beijing, China, 6–9 December 2015; pp. 1–13.
52. Ding, H.; Jiang, X.; Shuai, B.; Liu, A.Q.; Wang, G. Semantic Correlation Promoted Shape-Variant Context for Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8877–8886.
53. Ding, H.H.; Jiang, X.D.; Shuai, B.; Liu, A.Q.; Wang, G. Semantic Segmentation With Context Encoding and Multi-Path Decoding. *IEEE Trans. Image Process.* **2020**, *29*, 3520–3533. [CrossRef]
54. Mehta, S.; Rastegari, M.; Shapiro, L.; Hajishirzi, H. ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9182–9192.
55. Zhang, X.T.; Chen, Z.X.; Jonathan, W.Q.M.; Cai, L.; Lu, D.; Li, X.M. Fast Semantic Segmentation for Scene Perception. *IEEE Trans. Ind. Inform.* **2019**, *15*, 1183–1192. [CrossRef]
56. Li, H.C.; Xiong, P.F.; Fan, H.Q.; Sun, J. DFANet: Deep Feature Aggregation for Real-Time Semantic Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 9514–9523.
57. Yu, C.; Gao, C.; Wang, J.; Yu, G.; Shen, C.; Sang, N. Bisenet v2: Bilated network with guided aggregation for real-time semantic segmentation. *arXiv* **2017**, arXiv:2004.02147.
58. Wang, S.; Chen, Z.; Li, J.; Wang, J.; Li, J.; Zhao, J. Flexible motion framework of the six wheel-legged robot: experimental results. *IEEE/ASME Trans. Mechatronics* **2021**, 1–9. [CrossRef]
59. Li, J.; Dai, Y.; Wang, J.; Su, X.; Ma, R. Towards broad learning networks on unmanned mobile robot for semantic segmentation. In Proceedings of the 2022 IEEE International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 9228–9234.