applied sciences

MDPI

Article

Construction of Three-Dimensional Semantic Maps of Unstructured Lawn Scenes Based on Deep Learning

Xiaolin Xie [1,2], Zixiang Yan [2,*], Zhihong Zhang [2], Yibo Qin [2], Hang Jin [2], Cheng Zhang [2] and Man Xu [2]

[1] Longmen Laboratory, Luoyang 471003, China; xiexiaolin@haust.edu.cn
[2] College of Agricultural Equipment Engineering, Henan University of Science and Technology, Luoyang 471003, China; lyzzh@haust.edu.cn (Z.Z.); 210321041662@stu.haust.edu.cn (Y.Q.); 220320261796@stu.haust.edu.cn (H.J.); 230320261490@stu.haust.edu.cn (C.Z.); 220320261812@stu.haust.edu.cn (M.X.)
* Correspondence: 210321041670@stu.haust.edu.cn

Abstract: Traditional automatic gardening pruning robots generally employ electronic fences for the delineation of working boundaries. In order to quickly determine the working area of a robot, we combined an improved DeepLabv3+ semantic segmentation model with a simultaneous localization and mapping (SLAM) system to construct a three-dimensional (3D) semantic map. To reduce the computational cost of its future deployment in resource-constrained mobile robots, we replaced the backbone network of DeepLabv3+, ResNet50, with MobileNetV2 to decrease the number of network parameters and improve recognition speed. In addition, we introduced an efficient channel attention network attention mechanism to enhance the accuracy of the neural network, forming an improved Multiclass MobileNetV2 ECA DeepLabv3+ (MM-ED) network model. Through the integration of this model with the SLAM system, the entire framework was able to generate a 3D semantic point cloud map of a lawn working area and convert it into octree and occupancy grid maps, providing technical support for future autonomous robot operation and navigation. We created a lawn dataset containing 7500 images, using our own annotated images as ground truth. This dataset was employed for experimental purposes. Experimental results showed that the proposed MM-ED network model achieved 91.07% and 94.71% for MIoU and MPA metrics, respectively. Using a GTX 3060 Laptop GPU, the frames per second rate reached 27.69, demonstrating superior recognition performance compared to similar semantic segmentation architectures and better adaptation to SLAM systems.

Keywords: semantic segmentation; improved DeepLabv3+; ORB-SLAM3; attention mechanism; 3D semantic maps

check for updates

Citation: Xie, X.; Yan, Z.; Zhang, Z.; Qin, Y.; Jin, H.; Zhang, C.; Xu, M. Construction of Three-Dimensional Semantic Maps of Unstructured Lawn Scenes Based on Deep Learning. *Appl. Sci.* **2024**, *14*, 4884. https://doi.org/10.3390/app14114884

Academic Editor: Jose Machado

Received: 31 January 2024
Revised: 19 May 2024
Accepted: 3 June 2024
Published: 4 June 2024

1. Introduction

In recent years, with continuous advancements in computer vision, deep learning, and artificial intelligence technologies, the widespread application of depth sensors [1] and Light Detection and Ranging (LiDAR) [2] has propelled the rapid development of three-dimensional (3D) semantic maps [3]. Robots are playing an increasingly important role in daily life, undertaking various tasks to make our lives more convenient. To better fulfill these tasks, higher demands are placed on the performance of robots. In the construction of 3D semantic maps for unstructured scenes such as lawns [4,5], surrounding environment perception, autonomous navigation, localization, and scene understanding become crucial for gardening pruning robots, assisting them in achieving better lawn-mowing results.

In the field of deep learning, although object detection [6] can identify the positions and categories of objects in images, it cannot accurately detect the specific boundaries of large or irregular targets, such as lawns, streets, sky, and wall cracks. However, medical applications and applications in intelligent robots, drones, and other technologies often require the precise identification of the locations and boundaries of target areas, involving a region target recognition and boundary localization problem which can be generalized as a boundary detection problem [7]. When addressing the issue of lawn area and its boundary detection, it is necessary to identify the lawn and other elements by analyzing the semantics of the image scene. Based on this information, the boundaries of the lawn and the positions of obstacles must be located to support the robot's autonomous navigation and obstacle avoidance.

Typically, the maintenance of lush lawns is carried out by gardeners using hand-held mowers or riding mowers. In recent years, some smart lawn mowers have also emerged, but they often require the installation of electronic fences [8] before they begin working. These fences delineate the robot's maximum working range to prevent it from entering inappropriate areas and causing equipment damage. Overall, traditional lawn-mowing methods primarily rely on the autonomous responses of humans or robots, lacking a holistic awareness of the required working area. Lawn trimming is usually a manual, labor-intensive, and time-consuming task. However, with labor shortages, aging populations, and declining birth rates becoming common challenges faced by agriculture and other labor-intensive industries, the development of robotic technology becomes an optimal strategy for addressing this problem, enabling robots to replace gardeners in performing lawn trimming tasks.

In recent years, semantic image segmentation [9] has emerged as a key task in image processing. This rapid progress has led to a paradigm shift in the way many fields approach problem solving. Consequently, the benefits of utilizing semantic image segmentation in autonomous lawnmowers are evident. It enables better recognition of the surrounding environment, accurately identifying environmental information, thereby facilitating the creation of a global map of the environment for improved path planning in the future.

In this study, we created a lawn dataset using self-annotated images as ground truth. We employed the proposed Multi-Class MobileNetV2 ECA DeepLabv3+ (MM-ED) network to segment the lawn dataset and compared its performance with other deep learning algorithms. Experimental results show that compared to other algorithms, the architecture proposed in this paper is suitable for integration into ORB-SLAM3 and demonstrates higher accuracy relative to other algorithms, making it suitable for real-time, resource-constrained mobile robot applications. Moreover, the system framework proposed in this paper, which integrates MM-ED and ORB-SLAM3, can effectively generate 3D semantic point cloud maps, enabling the robust reconstruction of the working area and providing crucial technical support for the autonomous operation and navigation of gardening pruning robots.

2. Related Work

2.1. Technology for the Semantic Segmentation Recognition of Images

For gardening pruning robots, detecting a viable working area is a prerequisite for autonomous navigation. In the realm of semantic detection, scholars worldwide have proposed a plethora of detection algorithms for image semantic segmentation in recent decades, such as PSPNet, U-Net, and HRNet. Although research on the semantic segmentation of lawn scenes is relatively scarce, semantic segmentation networks have found widespread application in other areas.

In terms of reducing the complexity of the backbone network, Yan et al. proposed a method for tea bud segmentation and picking point localization based on a lightweight convolutional neural network, Multiclass DeepLabV3+ MobileNetV2 (MC-DM). By optimizing MobileNetV2 and introducing a densely connected spatial pyramid pooling module, they achieved high-precision tea bud segmentation [10]. Shi et al., aiming to achieve the automatic and effective extraction of stand-alone tree elements in images, proposed a lightweight network segmentation model, SENet and MobileNet embedded in DeepLabV3+ (SEMD). By choosing DeepLabV3+ as a base framework, integrating MobileNet to reduce complexity, and introducing SENet to capture feature information, they managed to achieve the high-precision segmentation of stand-alone tree images against complex backgrounds [11].

In terms of enhancing model accuracy, Feng et al. improved the DeepLabV3+ network model by embedding the CBAM module and utilizing SENet to optimize the decoding part, significantly enhancing the accuracy and localization precision of semantic segmentation for parts of Tilapia, effectively addressing the issues of target edge and small object segmentation [12]. Cai et al., addressing the issue of the inaccurate segmentation of strawberries at different maturity stages, proposed a method based on an improved DeepLabV3+ model. By introducing an attention mechanism to adjust the weights of neural network feature channels, they significantly enhanced the feature information of strawberry images, markedly improving segmentation precision [13]. Zheng et al. proposed a Lateral Hierarchically Refining Network (LHRNet) for the precise detection of salient objects, effectively integrating multi-level features and merging coarse semantics with fine details to produce more reliable predictions [14].

Regarding fusion with other algorithms, Chen et al. proposed a non-contact measurement method for concrete plane structure cracks based on binocular vision, using an improved DeepLabV3+ model for crack area segmentation combined with SIFT feature-point-matching principles to calculate crack width, proving that this method solves the problem of the rapid and convenient measurement of concrete crack width in outdoor environments [15].

It is evident that semantic segmentation has provided rich solutions for addressing various issues, hence the motivation to apply this approach to constructing lawn environments with the intention to achieve effective lawn semantic segmentation results.

2.2. Three-Dimensional Semantic Map Technology

Unstructured 3D point cloud maps are crucial for perceiving surrounding environments, autonomous navigation, and autonomous scene understanding. For gardening pruning robots, unstructured 3D point cloud semantic segmentation technology is a key technology for achieving autonomous operation and navigation. Three-dimensional semantic maps can assist gardening robots in recognizing and detecting obstacles, thereby providing obstacle avoidance and motion path-planning capabilities, significantly improving the operation and production efficiency of gardening robots. After performing semantic segmentation on captured images to obtain the semantic information of different objects, directly mapping the semantic mask images to actual physical 3D coordinates can establish a 3D point cloud map of the working environment.

In terms of integrating the construction of 3D semantic maps with deep learning methods, Cui et al. integrated an improved YOLOV3 algorithm to propose a new method for constructing 3D semantic maps for mobile robots, combining it with a SLAM system for pose estimation and three-dimensional environment construction [16]. Chen et al., by improving the neural network structure of RandLA-Net, constructed a deep 3D point cloud semantic segmentation model for large-scale unstructured agricultural scenes [17]. Yajima et al. utilized Structure from Motion (SFM) to create 3D point clouds and employed deep learning for the segmentation and classification of highway assets, addressing the issues of labor-intensive and time-consuming processes in highway maintenance and infrastructure monitoring [18].

Regarding research on constructing semantic maps of different scales, Koch et al. focused on small-scale 3D reconstruction to generate detailed, brief, and safe flight plans for unmanned aerial vehicles (UAVs) in environments, considering semantic properties through discrete optimization [19]. Zhang et al. proposed a method for reconstructing dense, large-scale outdoor 3D semantic maps based on monocular vision, achieving semantic segmentation through visual odometry, depth estimation, and a deep learning-based conditional random field (CRF) image segmentation system, ultimately producing dense urban environment 3D semantic maps with global consistency [20].

It is clear that in many instances, establishing 3D semantic maps can significantly aid in improving work processes and solving many previously unresolved issues, making the construction of 3D semantic maps for lawn environments both necessary and meaningful.

3. Materials and Methods

3.1. Semantic Segmentation Module

3.1.1. Basic Principles of Deeplabv3+

DeepLabv3+ [21] is a representative architecture in the field of semantic segmentation, having evolved from DeepLabv1, which was released by Google in 2014. It enhances the speed and accuracy of semantic segmentation by introducing concepts such as conditional random fields (CRFs), Atrous Spatial Pyramid Pooling (ASPP), and the encoder–decoder architecture. It mainly consists of two parts: an encoder module and a decoder module. In the encoding stage, the input image is first passed through a backbone network; some researchers use residual networks (ResNet) as backbone networks to obtain feature maps which are downsampled 16 times. Then, these downsampled feature maps are fed into the ASPP module. The ASPP module consists of a 1 × 1 convolution, an average pooling layer with global information, and three 3 × 3 dilated convolution layers with dilation rates of 6, 12, and 18, respectively. Finally, the feature maps obtained from the ASPP module are concatenated, and the channel number is compressed to 256 through a 1 × 1 convolution. In the decoding part, the output feature maps from the encoding part are upsampled four times using a bilinear interpolation algorithm. Then, they are concatenated with same-resolution feature maps extracted by the backbone network. Lastly, a 3 × 3 convolution refines the concatenated features, resulting in the segmentation output. The overall architecture of DeepLabv3+ is illustrated in Figure 1.

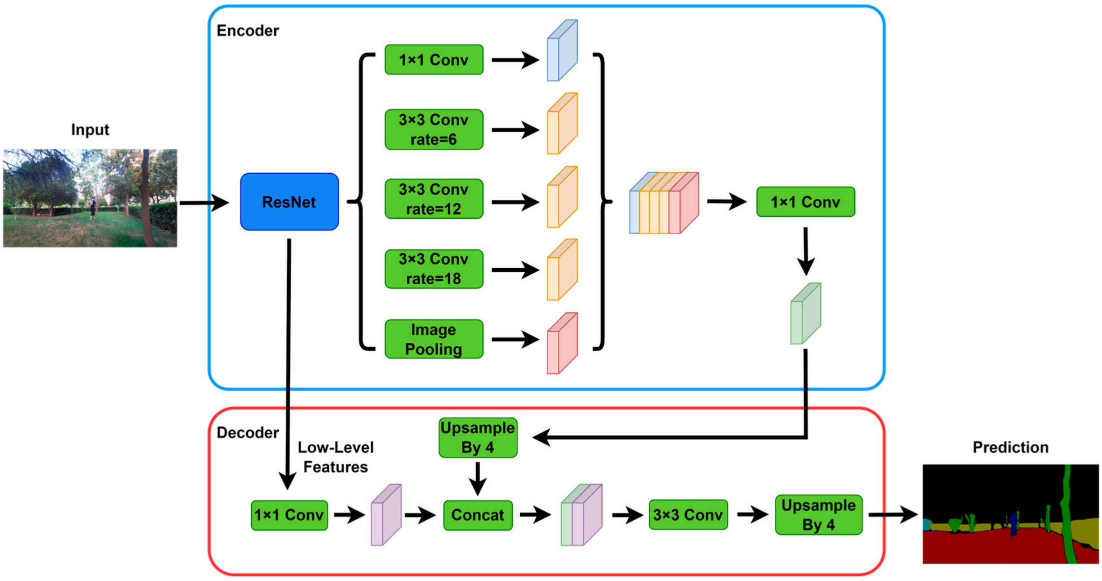


Figure 1. DeepLabv3+ architecture.

3.1.2. Improved DeepLabv3+ Architecture

In this paper, an improvement scheme is proposed for the original DeepLabv3+ model, which has problems such as a large number of model parameters, inconvenient portability, and poor real-time performance. The improved network structure is shown in Figure 2, and the improved lightweight real-time multi-object semantic segmentation detection network is called Multiclass MobileNetV2 ECA DeepLabv3+ (MM-ED). The improvement is mainly reflected in the following two aspects:

1. The lightweight architecture MobileNetV2 [22] is introduced in the backbone network. This effectively reduces redundant computations and memory accesses, decreases model size, and speeds up inference.
2. The Efficient Channel Attention Network (ECANet) attention mechanism [23] is introduced at the end of the backbone network, incorporating channel attention. This greatly enhances the feature extraction capability of the model.

The model improvements proposed in this paper were optimized considering the limited computational capabilities and small storage spaces often found in the embedded platforms carried by gardening pruning robots devices. Reducing the size of the backbone network is conducive to operating efficiently on such platforms. Additionally, incorporating attention mechanisms can enhance the model's recognition accuracy within constrained computational resources.

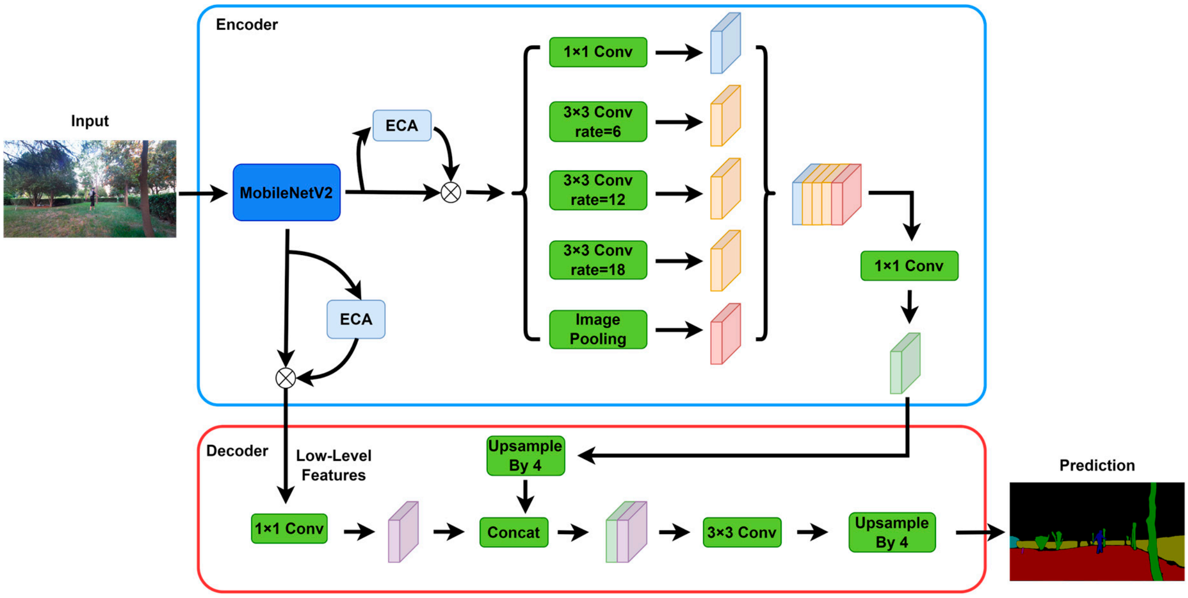


Figure 2. Improving the DeepLabv3+ architecture.

3.1.3. Lightweight Backbone Network MobileNetV2

MobileNetV2 is a lightweight neural network model proposed by the Google team in 2018. Compared to the traditional residual structure of ResNet, MobileNetV2 has fewer parameters, a simpler model structure, and a faster network training speed.

The MobileNet architecture introduces depthwise separable convolution to replace standard convolution, reducing computational complexity and model parameters. Depthwise separable convolution decomposes a regular convolution into depthwise convolution and pointwise convolution. It first decomposes the input features into multiple single-channel features and applies a 3 × 3 convolution kernel on each channel, which is known as depthwise convolution. Then, pointwise convolution convolves the result of the depthwise convolution with a 1 × 1 convolution, assembling the output features. Depthwise separable convolution can significantly reduce computational complexity compared to standard convolution with a relatively modest decrease in prediction accuracy.

Building upon depthwise separable convolution, MobileNetV2 introduces an inverted residual structure and employs a linear layer in the final layer, further enhancing network performance. The inverted residual structure consists of three main parts. First, it increases the dimension of input features using a 1 × 1 convolution. Then, it extracts features using a 3 × 3 depthwise separable convolution, followed by a dimension reduction using a 1 × 1 convolution. Unlike the residual structure, the inverted residual structure avoids the drawback of the network performing well only on low-dimensional features when the input feature channel count is low, leading to the loss of high-dimensional feature information. To prevent the destruction of features by the ReLU function, the inverted residual structure employs 1 × 1 convolutions for dimensionality expansion before the 3 × 3 network structure and for dimensionality reduction after the 3 × 3 network structure. The Rectified Linear Unit (ReLU) activation function is replaced with a linear function to minimize the loss of useful network information. The network architecture of MobileNetV2 is shown in Table 1.

Table 1. MobileNetV2 network architecture.

Input	Network	t	c	n	s
224 × 224 × 3	Conv2d	-	32	1	2
112 × 112 × 32	Bottleneck	1	16	1	1
112 × 112 × 16	Bottleneck	6	24	2	2
56 × 56 × 24	Bottleneck	6	32	3	2
28 × 28 × 32	Bottleneck	6	64	4	2
14 × 14 × 64	Bottleneck	6	96	3	1
14 × 14 × 96	Bottleneck	6	160	3	1
7 × 7 × 160	Bottleneck	6	320	1	2
7 × 7 × 320	Conv2d 1 × 1	-	1280	1	1
7 × 7 × 1280	Avgpool 7 × 7	-	-	1	-
1 × 1 × 1280	Conv2d 1 × 1	-	k	-	-

In Table 1, “Input” represents the size of the input feature map for the current layer. “Network” shows the network structure of each layer in MobileNetV2, including regular convolution layers, inverted residual structures, and average pooling layers. “t” is the expansion factor for channels in the inverted residual structure, “n” is the repetition count for the current layer, “c” is the number of output channels, and “s” is the convolution stride for the current layer.

The method proposed in this paper ultimately needs to ensure that the neural network model can be invoked in ORB-SLAM3; in practical applications, it must be deployed on embedded devices. Typically, algorithms running on embedded devices have a faster recognition speed and utilize less storage space while achieving approximate recognition accuracy. MobileNetV2 is a lightweight network suitable for embedding in mobile devices. Compared to the ResNet series network originally used as the backbone feature extraction network in DeepLabv3+, MobileNetV2 is better suited for deployment on edge devices, offering more efficient parameters and a faster speed.

3.1.4. Attention Mechanism ECA

In the original DeepLabv3+ model, after passing through the backbone network, the model directly enters the ASPP layer for processing, with the importance of each channel’s semantic information weighted equally by default. The attention mechanism can be considered a dynamic weight adjustment process based on input features. In the visual domain, humans can extract information from areas of interest in complex scenes using the brain. Inspired by this observation, attention mechanisms have been introduced in computer vision to mimic this aspect of the human visual system.

ECANet is an image attention mechanism focused on enhancing the model’s attention to edge information, aiming to improve the performance of image-processing tasks. Here are some key features of ECANet:

1. ECANet introduces a contextual attention mechanism to allow the model to better understand relationships between pixels in an image. This helps the model capture contextual information in the image more effectively.
2. By enhancing attention to edge information, ECANet improves the model’s perception of details. This is crucial for many image-processing tasks such as object detection and image segmentation.
3. ECANet introduces an adaptive pooling mechanism to dynamically adjust the size of attention, enabling the model to adapt better to targets or details of different sizes.

The design of ECANet aims to enhance the model’s perception of important information in the image, particularly when dealing with complex images. This mechanism has demonstrated good performance in many image-processing tasks. Figure 3 provides a schematic diagram of the ECANet attention mechanism’s network structure.

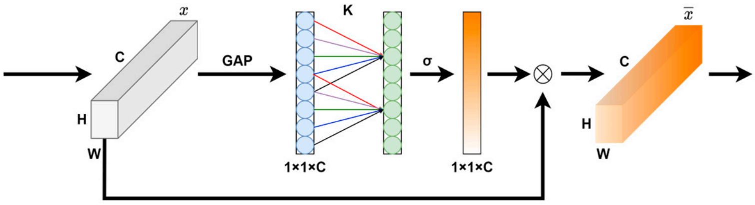


Figure 3. ECANet attention mechanism.

In Figure 3, "*k*" represents the optimal range for channel information interaction, which is the kernel size of the one-dimensional convolution, calculated using Formula (1).

$$k = \left| \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right|_{odd} \tag{1}$$

In Formula (1), "*C*" is the number of feature channels, and in this paper, γ and b are set to 2 and 1, respectively. These parameters are used to adjust the ratio between the number of channels "*C*" and the kernel size "*k*". When there are many channels, the kernel size "*k*" increases with "*C*" and vice versa when there are fewer channels. This approach enables effective interactions between different channels, promoting fusion.

In the context of image semantic segmentation tasks, low-level feature maps play a crucial role as they contain the original details and texture information of the image. In the structure presented in this paper, the ECANet attention mechanism is introduced before the high-level information is fed to the ASPP and is also added to the low-level features provided to the decoding part. The inclusion of the ECA module enhances the network's ability to express local features, expands the receptive field, and facilitates the capture of more diverse feature information. The adaptive channel weighting of feature maps enhances correlation between different channels, allowing for a better exploration of details and texture information and thus improving the recognition of object boundaries and shapes.

3.1.5. Model Deployment

Due to the fact that the lightweight, real-time semantic segmentation network MM-ED is implemented based on PyTorch [24] and the ORB-SLAM3 system cannot directly invoke PyTorch models, it is necessary to export this architecture in a form that can be used in the ORB-SLAM3 system. To achieve this, we chose to use TorchScript, a tool in PyTorch for converting models into an efficiently executable intermediate representation, to enhance model performance and deployment efficiency. In this paper, we converted the inference model into the TorchScript format and invoked it using LibTorch in C++, thereby implementing the inference functionality.

This deployment approach offers flexibility and high performance, enabling the MM-ED network to be seamlessly integrated into the ORB-SLAM3 system, providing an efficient solution for real-time segmentation tasks. By utilizing TorchScript and LibTorch, we can leverage the powerful capabilities of PyTorch and achieve high-performance inference using the model in different environments. This integrated approach provides a convenient and effective way of applying the architecture, making the deployment of deep learning models in practical systems more feasible. Figure 4 illustrates the deployment and invocation process of our model.

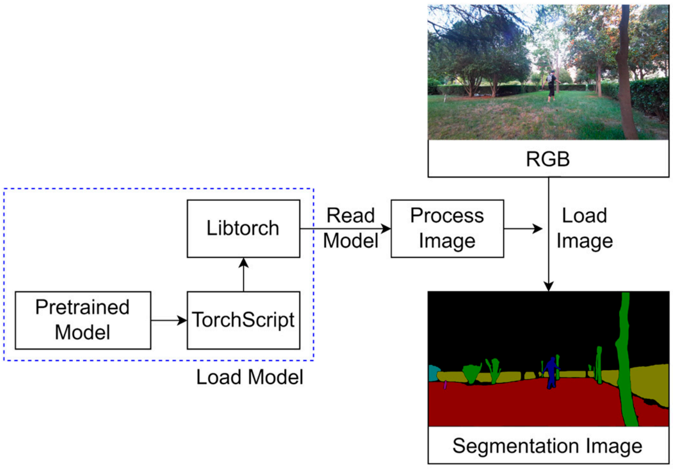


Figure 4. Deployment and invocation process of MM-ED model.

3.2. Visual SLAM System Framework

In the exploration of unknown environments, mobile robots utilize various sensors to gather environmental information. Currently, there are two mainstream SLAM methods: Visual SLAM [25] and LiDAR-based SLAM [26]. They each have their advantages and find widespread application in their respective domains. LiDAR, which is used for constructing 3D point clouds, operates under various lighting conditions and provides precise measurements and localization. However, LiDAR devices are relatively expensive compared to the cameras used in Visual SLAM. Therefore, the lawn semantic map construction system designed in this paper utilizes Visual SLAM technology. The main visual sensors are categorized into three types, monocular cameras [27], stereo cameras [28], and RGB-D depth cameras, corresponding to different types of Visual SLAM technologies.

Compared to monocular and stereo cameras, RGB-D cameras have significant advantages in building 3D point cloud maps. Firstly, RGB-D cameras can provide depth information for each pixel simultaneously while capturing color images, whereas stereo cameras require stereo-vision-matching calculations to obtain depth information. This allows RGB-D cameras to avoid additional and complex depth calculations during the mapping process, reducing the computational burden and enhancing real-time mapping efficiency. Secondly, RGB-D cameras can comprehensively capture the three-dimensional information of a scene. Monocular cameras provide only two-dimensional image information, and although stereo cameras can obtain depth information, their disparity calculations and matching process are susceptible to factors like lighting and texture. RGB-D cameras, by directly measuring depth, overcome these issues, resulting in a more accurate and stable construction of the 3D point cloud map. This is crucial for the navigation and localization of mobile robots in unknown environments. In summary, RGB-D cameras are superior at building 3D point cloud maps, providing essential support for the efficient and reliable perception and navigation of mobile robots in complex environments. Since the method proposed in this paper requires the construction of a 3D point cloud map, using an RGB-D camera is more efficient.

SLAM technology mainly consists of four key components: sensor data acquisition, motion estimation and localization optimization, loop closure detection, and map construction. A typical Visual SLAM framework, as illustrated in Figure 5, is roughly divided into two main modules: a frontend and a backend. The frontend is responsible for sensor data collection and visual odometry, while the backend includes nonlinear optimization and map-building loop closure detection to correct the results obtained by the frontend. This system enables robots to explore unknown and unstructured environments, acquiring rich and specific information within their unstructured surroundings. This information can be utilized for future analysis, localization, and navigation within the environment.

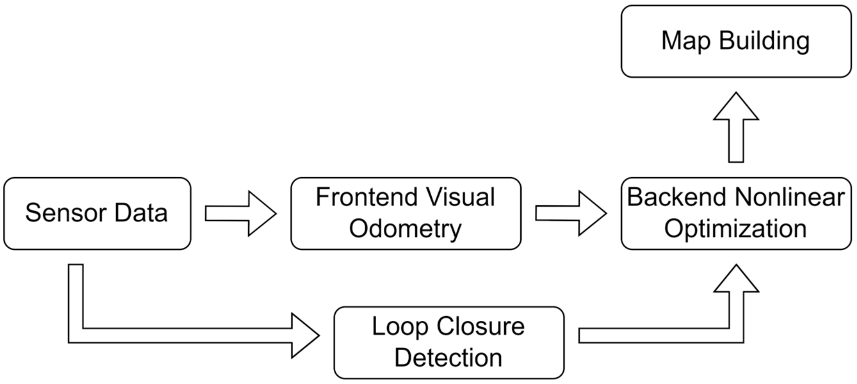


Figure 5. Visual SLAM system framework.

The ORB-SLAM [29], proposed in 2015, stands out as one of the most sophisticated and user-friendly SLAM systems in contemporary robotics and computer vision applications. ORB-SLAM3, an enhanced version of its predecessor ORB-SLAM2, integrates state-of-the-art Oriented FAST and Rotated BRIEF (ORB) feature description and matching techniques. This system provides robust and efficient visual odometry and mapping capabilities. We will further explore the realm of 3D semantic maps of unstructured lawn scenes by combining our MM-ED model with ORB-SLAM3.

3.3. Mapping Module

To obtain more comprehensive 3D map information, we extended ORB-SLAM3 to achieve semantic map construction for lawn scenes. Originally, ORB-SLAM3 provided a sparse map, but we aimed to combine feature points and distance information from RGB-D cameras to generate a denser point cloud map. Building upon ORB-SLAM3, we introduced an additional thread dedicated to constructing a dense point cloud map. To avoid unnecessary calculations and redundant information, we chose to pass RGB images and depth information into the point cloud construction thread upon keyframe recognition, efficiently generating a dense point cloud map.

Figure 6 illustrates how we improved the processing pipeline for point cloud maps. In the original dense point cloud construction process, the camera captures video and ORB-SLAM3 extracts keyframe information from the video. By combining RGB images and depth information from the keyframes, a dense point cloud is obtained. In our new processing pipeline, RGB images are inputted into our proposed MM-ED network model to obtain masked images. These masked images are then combined with depth information to obtain a 3D semantic map.

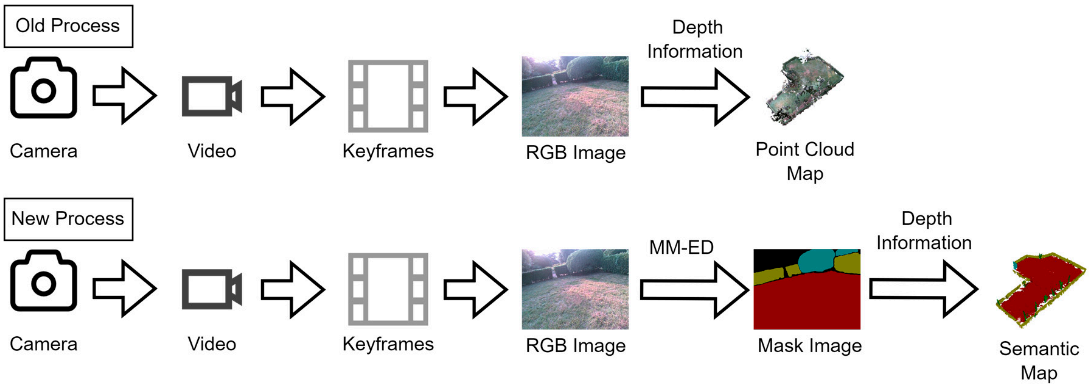


Figure 6. Improvements in point-cloud-map-processing workflow.

Our point cloud construction system not only considers the geometric structure of the map but also incorporates semantic information by processing RGB images using the

MM-ED neural network. This step aids in identifying regions in the image with semantic information, generating corresponding mask images. Subsequently, we construct a point cloud map based on this semantic information, enabling the map to not only include geometric structures but also possess semantic information. This enhances the map's expressive capabilities, significantly reduces point cloud storage space, and optimizes map representation efficiency by eliminating many irrelevant points.

In this paper, we fully consider the semantic map construction module and propose the simultaneous construction of two forms of maps. By transforming point cloud information into an OctoMap [30] and an occupancy grid map [31], we comprehensively express the features and semantic information of the environment.

OctoMap is a highly efficient three-dimensional spatial data structure. It implicitly represents and manages objects in space by recursively dividing three-dimensional space into eight subnodes. It does not require the explicit storage of triangular meshes, greatly saving memory. OctoMap can efficiently and finely manage and represent objects in three-dimensional space, which is a significant advantage of its use in three-dimensional applications. In terms of navigation, the octree structure efficiently represents hierarchical information in space, allowing the map to describe the details of the environment more finely, including objects and structures at different levels, providing good assistance for three-dimensional navigation.

The advantage of occupancy grid maps lies in their ability to perform more accurate path planning. By partitioning a map into grids, obstacles and passable areas can be clearly represented, allowing for more precise calculations of optimal paths. Grid maps can also help a robot avoid obstacles and choose shorter, safer paths when selecting routes. Additionally, grid maps can be used for real-time updates, adjusting paths based on actual conditions, making path planning more flexible and adaptive to changes. By fully leveraging the advantages of grid maps, we can improve the accuracy and efficiency of path planning, achieving superior navigation and path guidance.

The corresponding semantic map construction framework is illustrated in Figure 7. The map models obtained through this method serve as relative position references for the robot when it needs to reposition itself in similar environments.

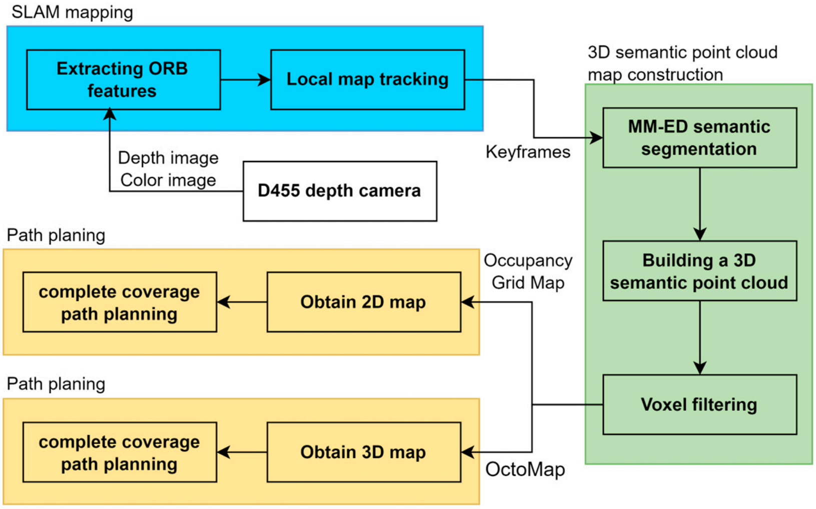


Figure 7. Map model construction framework.

4. Experiments and Evaluation

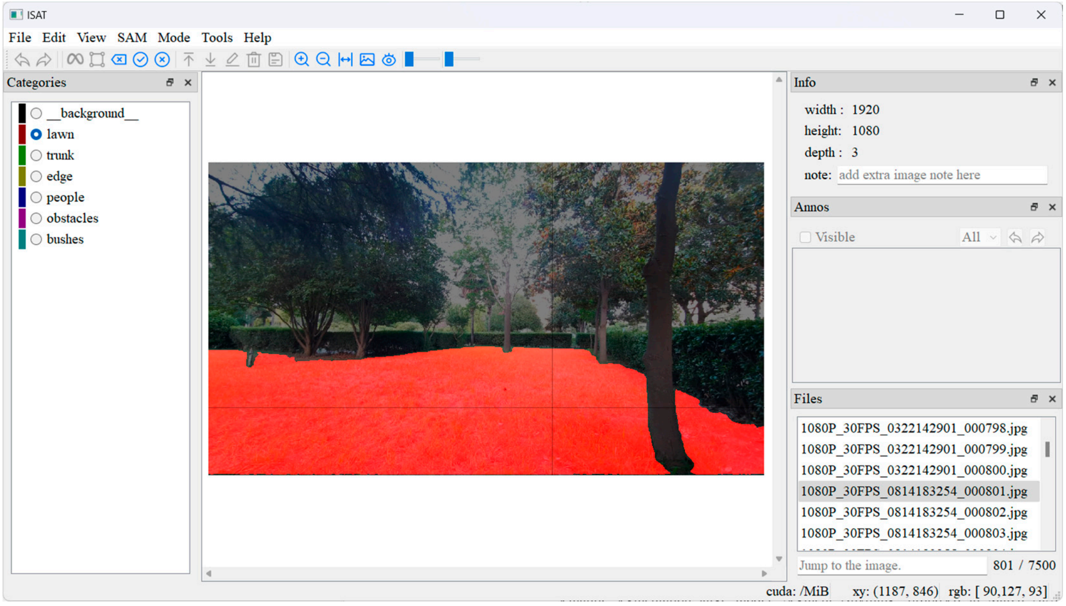
In this section, we conduct experiments with and validate the proposed MM-ED network model. In Section 4.1, we introduce the dataset used and its annotation details. Section 4.2 provides a detailed description of our experimental platform and parameter settings. Subsequently, in Section 4.3, we present the metrics used to evaluate the improved model. In Section 4.4, we conduct a series of comparative experiments and ablation studies aimed at demonstrating the effectiveness of the improvements in the backbone and attention mechanism on model performance. We compare the proposed method with representative methods in the segmentation field, including U-Net [32], HRNet [33], FCN [34], and DeepLabv3+. In Section 4.5, we integrate the improved MM-ED network into the ORB-SLAM3 system for the establishment of a 3D point cloud semantic map, generating occupancy grid maps and OctoMaps. This step aims to validate the effectiveness of the mapping method proposed in this paper.

4.1. Dataset Descriptions

4.1.1. Image Annotation

We used the open-source semantic segmentation image labeling tool ISAT_with_segment_anything [35] for image annotation. This tool is built on top of the latest semantic segmentation model, Segment Anything [36], which allows users to utilize the model's prediction results to assist in labeling, significantly improving efficiency and effectively reducing our workload. The model achieved higher accuracy on multiple datasets compared to previous models, with higher segmentation accuracy and generalization capability, thereby enhancing the quality of the annotations. Additionally, the tool employs interactive logic, allowing for iterative improvements on selected areas to ensure the desired accuracy. Specifically, when labeling lawn areas, for example, users only need to click on any point within the lawn area and the tool will invoke the model for prediction and generate a draggable polygon based on the prediction results. The accuracy of whether objects are correctly labeled primarily depends on whether the labeled contours match the outer boundaries of the objects. If the generated prediction results do not align with our desired outcome, corrections can be made within the tool through smudging and dragging operations to ensure accuracy in labeling. Compared to traditional tools like Labelme (where labeling blocks requires outlining their entire contours), using this tool for annotation is much faster. The final annotation results for each object are manually verified by us to ensure accuracy and are then used as ground truth images. In the experiments, the dataset was divided into a 70:30 ratio for training and testing.

The interface of the annotation tool is as shown in Figure 8, where the red area represents the lawn region automatically predicted by the model Segment Anything. We can further adjust it to meet our desired objectives.



○ __background__
● lawn
○ trunk
○ edge
○ people
○ obstacles
○ bushes

width : 1920
height: 1080
depth : 3
note: add extra image note here

□ Visible All

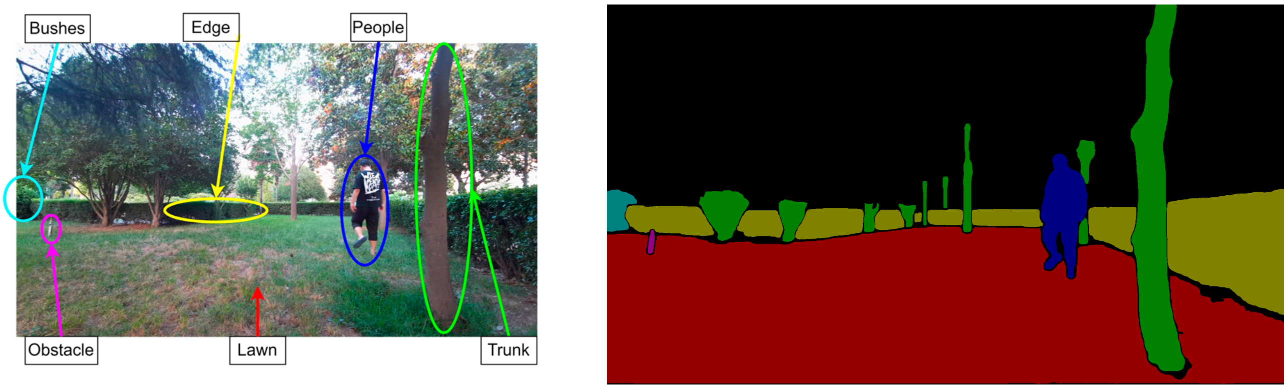
1080P_30FPS_0322142901_000798.jpg
1080P_30FPS_0322142901_000799.jpg
1080P_30FPS_0322142901_000800.jpg
1080P_30FPS_0814183254_000801.jpg
1080P_30FPS_0814183254_000802.jpg
1080P_30FPS_0814183254_000803.jpg

Jump to the image. 801 / 7500

cuda: /MiB xy: (1187, 846) rgb: [90,127, 93]

Figure 8. ISAT_with_segment_anything tool.

We manually divided the collected images into seven categories: Background (cls0, black), Lawn (cls1, red), Trunk (cls2, green), Edge (cls3, yellow), People (cls4, blue), Obstacle (cls5, purple), and Bushes (cls6, cyan). The original image was a 24-bit RGB image, and its corresponding visual labels are illustrated in Figure 9.



(**a**) Original image.

(**b**) Visual labeling.

Figure 9. Lawn images and their corresponding visualization labels.

Our explanation of the annotation situation is based on the working conditions of common gardening pruning robots. "Lawn" represents the robot's working area, while "Edge" delineates the boundaries of the robot's operation. The weeding tasks of the robot are carried out within this boundary, and the yellow-bordered area serves as a traditional electronic fence. Within the robot's working area, it is not limited to only the lawn; sometimes, objects such as "Bushes," an "Obstacle," or a "Trunk" may appear in the lawn area. In this study, these three types of objects are taken as representatives of obstacles, forming prohibited areas for the robot. During the pruning process, there are always "People" appearing nearby. These individuals might be supervising the robot's work or simply passing by. Unlike other static objects, "People" are dynamic and somewhat special, so they need to be annotated separately.

4.1.2. Lawn Environment

We collected and established a dataset containing 7500 images of various lawn scenes. The lawn images used in the experiments were obtained on various lawns on the campus from mid-March to mid-September 2023. To validate the effectiveness of our semantic segmentation architecture, we typically used lawn environments such as the one shown in Figure 10. This image is presented from a bird's-eye view and depicts our envisioned lawn working scenario, which is not a completely flat area without obstacles. Instead, it often includes a variety of obstacles that hinder the operation of gardening pruning robots. It is precisely because of the presence of various obstacles that there is a need to establish 3D semantic maps to assist gardening pruning robots in working more efficiently.

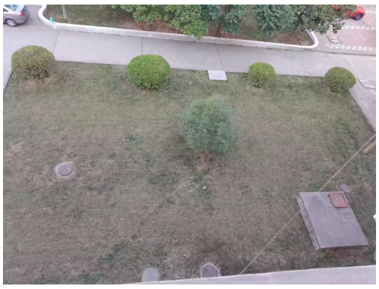
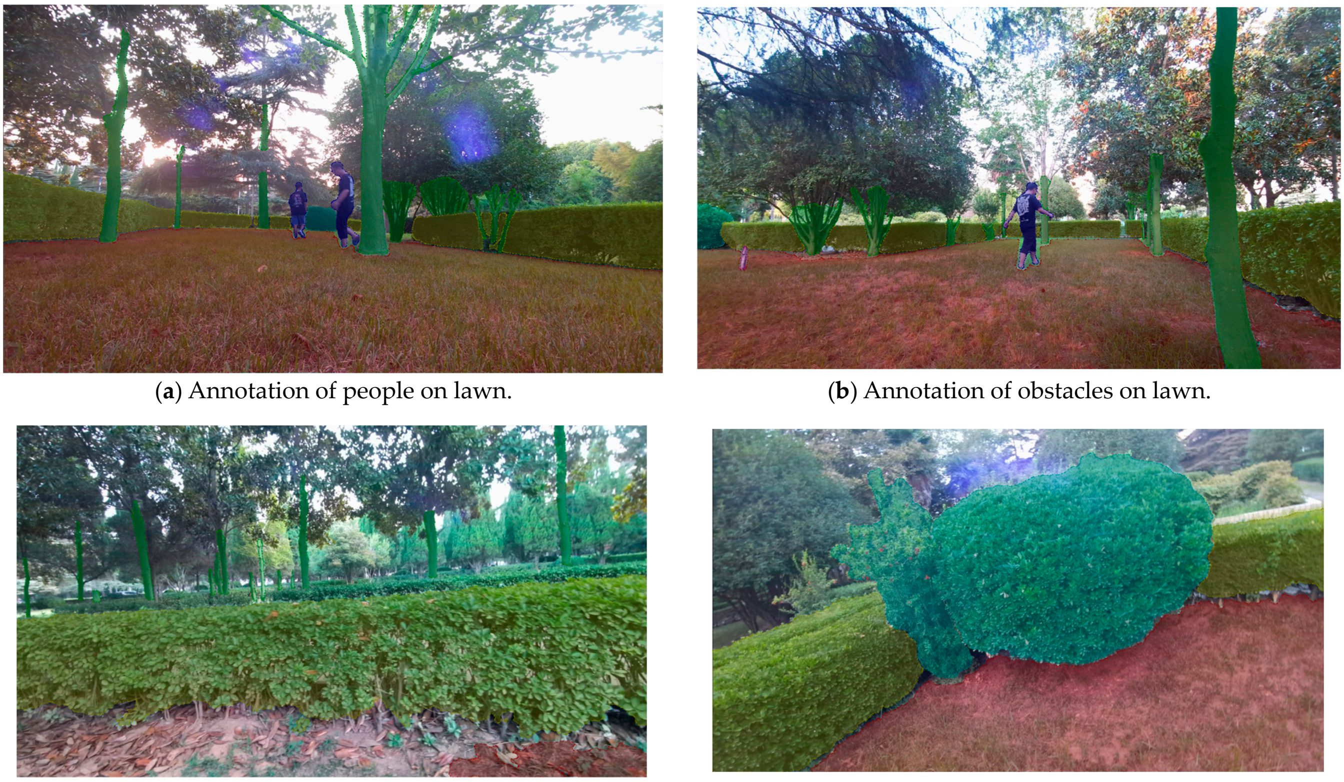


Figure 10. A schematic diagram of the experimental lawn, which often contains a variety of obstacles.

We captured some images with a resolution of 1280 × 1920 using an Azure Kinect DK and some with a resolution of 640 × 480 using an Intel Realsense D455 camera. High-resolution images were primarily used to aid with learning image information, while low-resolution photos were used for SLAM mapping. Generally, the shooting time was from 5:00 a.m. to 7:00 p.m. to ensure the diversity of the lawn environment, capturing lawns with different lighting conditions, locations, and obstacle situations.

Figure 11 selects some of the pictures from our dataset, displaying the annotated labels overlaid on the original images. Regarding the overall lawn environment, we have demonstrated in Figure 11a,b that the pictures contain all the elements we mentioned previously, including lawns, people, obstacles, bushes, tree trunks, and boundaries. However, capturing only pictures of types (a) and (b) would result in a large area of lawn in the pictures, making it difficult for other types of labels to be learned by the semantic segmentation network. Therefore, for other types of objects, we used pictures in which they appear extensively, such as in Figure 11c, where our picture focuses on an Edge, and similarly, in Figure 11d, which focuses on Bushes. For the labels People, Obstacles, and Trunk, we also obtained many pictures in which these objects appear extensively to aid in the learning of our network.


(**a**) Annotation of people on lawn.

(**b**) Annotation of obstacles on lawn.

(**c**) Annotation focused on Edge.

(**d**) Annotation focused on Bushes.

Figure 11. Annotations for different types of images in the lawn dataset.

4.2. Hardware and Software Configuration

The entire model training process was implemented by renting a "Tianchi Cloud" shared GPU server. The server's CPU is an Intel(R) Xeon(R) E5-2686 v4 @ 2.30 GHz with 60 GB of memory (Intel, Luoyang, China). The operating system used was Ubuntu 20.04, 64 bit, with Python 3.10, PyTorch 2.0.1, CUDA 11.8, and cuDNN 8 used for model training. A GPU (NVIDIA RTX A4000, NVIDIA, Santa Clara, CA, USA, 16 GB VRAM) was utilized to optimize the training speed. For experiments establishing the lawn working environment, a handheld depth camera and a laptop were used. The test model's frames per second (FPS) value was evaluated using an NVIDIA GeForce RTX 3060 Laptop GPU (6 GB) (NVIDIA, Santa Clara, CA, USA) and an Intel Core 12th i7-12700H CPU (Intel, Luoyang, China).

Table 2 provides detailed parameters for model training, such as the maximum iterations, learning rate, and the number of classes.

Table 2. DeepLabv3+ training parameters.

Parameter	Value
Leaning Rate	0.01
Weight Decay	0.0005
Momentum	0.9
Epoch	140,000
Batch Size	4
Number of Classes	7
Crop Size	512 × 512

4.3. Evaluation Indicators

To objectively assess the semantic segmentation model's performance on the lawn segmentation dataset and facilitate comparisons with various methods, the following evaluation metrics were adopted: the MIoU (Mean Intersection over Union) and MPA (Mean Pixel Accuracy).

The MIoU is widely used to measure pixel-level overlap between model-predicted segmentation results and actual labels. It calculates the intersection of predicted values and actual values for individual pixel classes. The MIoU reflects a model's ability to accurately segment image pixels by averaging the IoUs for all categories. In lawn segmentation tasks, a higher MIoU indicates the model's ability to accurately capture the boundaries and shapes of objects in the lawn environment. The formula is as follows (2):

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{TP}{FN+FP+TP} \tag{2}$$

The CPA (Category Pixel Accuracy) measures how many pixels predicted to be positive by the model are true positives. The calculation formula is as follows (3):

$$class\ i : P_i = \frac{TP_i}{TP_i + FP_i'} \tag{3}$$

The MPA (Mean Pixel Accuracy) calculates the proportion of correctly classified pixels in each class, i.e., the CPA, and then computes the average through accumulation, as shown in Formula (4):

$$MPA = \frac{sum(P_i)}{number\ of\ classes} \tag{4}$$

The calculation of these two parameters is carried out through a pixel-level comparison between the model's predicted results and the actual labels. These metrics provide robust support for objectively evaluating the performance of semantic segmentation models. Here, the symbol "k" represents the number of segmentation classes, and the symbol "i" represents the predicted category. True Positive (TP), False Negative (FP), True Negative (TN), and False Positive (FN) parameters are used, and this series of parameters forms a solid foundation for the in-depth analysis of the performance of semantic segmentation models.

In this study, the backbone network was replaced, and the original model underwent lightweight processing. Therefore, it was necessary to evaluate the model's size and running speed to verify the effectiveness of the improvements. We used metrics such as Giga Floating-Point Operations Per Second (GFLOPs), Params, and FPS for a comprehensive comparison.

Floating-Point Operations Per Second (FLOPs) is a critical metric for measuring the computational complexity and performance of neural network models. It represents the number of floating-point operations performed per second. In neural networks, floating-point operations involve floating-point arithmetic operations on weights, inputs, and activation values. GFLOPs, indicating billions of floating-point operations per second, is a key indicator for assessing model complexity. For resource-constrained devices such as mobile or edge devices, lower GFLOPs values may be more suitable to ensure the model runs smoothly on these devices.

Params refers to the number of parameters that the model needs to learn. These parameters include weights and bias terms which are adjusted during the model's training process to enable it to learn from input data and produce appropriate outputs. This is also an important metric for evaluating the complexity of a neural network model.

The FPS metric is used to evaluate detection speed, indicating the number of images processed per second or the time required to process a single image. A shorter time implies a higher speed. In this paper, we use the FPS metric to objectively evaluate the actual running speed of the proposed model.

By comprehensively comparing these metrics, we can gain a comprehensive understanding of the proposed model's advantages and disadvantages in terms of its complexity, scale, and practical running speed.

4.4. Experiment and Analysis

4.4.1. Backbone Comparison Experiment

In this study, we conducted experiments on different backbone networks for the encoder, aiming to compare their performance in lawn environment segmentation. Taking DeepLabV3+ as the baseline, we substituted the backbone network and carried out comparative experiments involving major backbones such as MobileNetV2, ResNet18 [37], ResNet50, and ResNet101.

By observing Table 3, we can see that among various backbone networks, MobileNetV2 achieves satisfactory results owing to its depthwise separable convolutional structure. Not only does this model reduce to a comparable level with ResNet18 in terms of GFLOPs and its parameter count but also in the recognition metric of FPS; the recognition speed of MobileNetV2 is almost twice that of ResNet50. It is noteworthy that although ResNet18 has a relatively small parameter count, its recognition accuracy is lower than that of MobileNetV2. Although the MIoU of MobileNetV2 is only 0.33% higher than that of ResNet18, the FPS value of ResNet18 is 16 higher than that of MobileNetV2, which might lead to the misconception that ResNet18 performs better. However, in practical applications, we need to input semantic segmentation results into ORB-SLAM3 systems to generate 3D semantic maps, with the system being most suitable for processing speeds of around 30 FPS. Exceeding this speed would result in an excessive burden on the system, possibly leading to delays in processing. Therefore, when FPS requirements are met, MobileNetV2, which has a higher MIoU, is the more appropriate choice.

Table 3. Backbone comparative results: MobileNetV2 achieves the best balance between accuracy and model parameters.

Backbone	MIoU/%	MPA/%	GFLOPs	Params/M	FPS
ResNet18	90.24	93.74	54.27	11.75	44.85
ResNet50	92.28	95.52	176.52	39.31	14.86
ResNet101	91.91	95.30	254.31	57.42	10.40
MobileNetV2	90.55	94.66	68.24	14.32	28.39

Hence, in scenarios in which simplifying backbone networks, improving recognition speed, and adapting to ORB-SLAM3 systems are objectives, selecting MobileNetV2 as the backbone network is an extremely attractive option. Overall, the results of this experiment indicate that in the task of lawn environment segmentation, MobileNetV2 demonstrates excellent performance and efficiency as the backbone network, providing strong support for achieving model lightweighting and accelerating recognition speed.

4.4.2. Experiment Comparing Attentional Mechanisms

From Table 3, it can be observed that adopting MobileNetV2 as the backbone network undoubtedly reduces the model parameters and GFLOPs while improving the recognition speed. However, both the MIoU and MPA show a decreasing trend compared to the commonly used original backbone network in DeepLabv3+, ResNet50. In order to maintain the lightweight effect at the network level by retaining MobileNetV2 as the backbone network while improving the accuracy of model recognition, we conducted experiments by introducing various attention mechanisms at the same position in the network.

We selected representative attention mechanisms for integration, including Efficient Channel Attention (ECA), a Squeeze-and-Excitation (SE) block [38], a Convolutional Block Attention Module (CBAM) [39], and Coordinate Attention (CA) [40]. These attention mechanisms were embedded at the same location within the backbone network to explore their impact on model performance.

The SE mechanism, through the introduction of a "Squeeze-and-Excitation" block structure, explicitly learned relationships between channels, resulting in more effective attention allocation. ECA focused primarily on channel dimensions, allowing the network to concentrate more on crucial features in the image, enhancing its perception of global information. The CBAM comprehensively considered both spatial and channel attention, capturing key information in the feature map. The CA attention mechanism encoded precise positional information in the neural network, aiding in modeling channel relationships and long-term dependencies.

The comparative experiments from Table 4 reveal that ECA demonstrates the most significant improvement in the MIoU when employed as the backbone network in the MobileNetV2-based model, showcasing its remarkable performance. Therefore, we opted for ECA as the attention mechanism for our model in this paper, aiming to further enhance the model's performance. Regarding the MIoU parameter rather than the MPA, the introduction of the ECA attention mechanism leads to a 0.52% increase in the MIoU for models switched to lightweight backbone networks. The MIoU serves as a metric for measuring the degree of overlap between predicted results and ground truth labels in semantic segmentation models, whereas the MPA predominantly focuses on the accuracy of predicted results. In the comparison provided in Table 4, we place greater emphasis on the impact on the MIoU because it not only considers the accuracy of predicted results but also consistency with ground truth labels. Therefore, in the comparison of these attention mechanisms, the MIoU holds more reference value.

Table 4. Comparative results of attention mechanisms: ECA attention mechanism achieves the best performance in MIoU metric.

Backbone	Attention	MIoU/%	MPA/%
MobileNetV2		90.55	94.66
MobileNetV2	ECA	91.07	94.71
MobileNetV2	SE	90.66	94.85
MobileNetV2	CBAM	90.89	94.88
MobileNetV2	CA	90.73	94.39

4.4.3. Ablation Experiment

This section evaluates modular variables in different neural networks and analyzes the factors influencing neural network performance. We used DeepLabv3+ as the baseline, using ResNet50 as the baseline's backbone. We conducted improvement experiments on the following different network structures: (1) Baseline + A, an ECA attention module embedded in the baseline method. (2) Baseline + B, with MobileNetV2 replacing the baseline method's backbone. We initially conducted single-variable improvement experiments. (3) Baseline + A + B: Both variables were added and experiments were performed to examine the combined effects on the model. Quantitative evaluations were performed on the test set for each experiment, and the results are presented in Table 5.

Table 5. Ablation experiment results.

Method	MIoU/%	MPA/%	GFLOPs	Params	FPS
Baseline	92.28	95.52	176.52	39.31	14.86
Baseline + A	92.86	96.13	176.53	40.87	14.85
Baseline + B	90.55	94.66	68.24	14.32	28.39
Baseline + B + A	91.07	94.71	68.25	15.10	27.69

From Table 5, it can be observed that our method exhibited a steady improvement in segmentation results compared to the baseline. The addition of the ECA module, whether integrated into the baseline or replacing the backbone with MobileNetV2, contributed to an increase in the model's recognition MIoU. After replacing the backbone network, there were significant reductions in computational and parameter counts accompanied by a substantial improvement in recognition speed. Regarding the MIoU parameters, the introduction of the ECA attention mechanism led to a 0.52% improvement in the model when transitioning to a lightweight backbone.

4.4.4. Image Segmentation Results

Using the same training environment and parameters, we conducted comparative experiments on different semantic segmentation networks. Figure 12 illustrates the prediction results of various networks. From top to bottom are the results of segmenting different types of images using FCN, HRNet18, U-Net, DeepLabv3+, and our proposed method, with the last row showing the annotated ground truth images. We chose to compare our method with FCN, HRNet18, U-Net, and DeepLabv3+, which are some of the most outstanding methods in the field of semantic segmentation, demonstrating excellent performance on various datasets. These methods represent different characteristics: FCN transforms traditional convolutional networks into fully convolutional structures for segmentation tasks; U-Net addresses the issue of information loss in semantic segmentation by introducing skip connections; DeepLabv3+ enhances segmentation performance through dilated convolutions and multi-scale feature fusion; and HRNet improves segmentation accuracy by preserving high-resolution features. By comparing these representative methods, we can gain a comprehensive understanding of the strengths and limitations of different segmentation approaches.

Our model demonstrates superior segmentation results across the six object categories requiring segmentation, with masks that are smoother and more complete. It is observable that in the recognition of lawns, boundaries, and humans, the performances of these algorithms appear quite similar. However, in the recognition of tree trunks, although the FCN and U-Net networks can correctly identify tree trunks, their ability to recognize lawns is significantly lacking, which is undesirable. In the recognition of obstacles, the algorithm proposed in this paper produced results closest to the ground truth images, with other algorithms missing some details. In recognizing shrubs, especially in the hard-to-recognize area on the lower right side of the shrubs, our algorithm's recognition effect is visibly better than that of the other algorithms.

Overall, as shown in Table 6, our method achieves segmentation results that are nearly identical to those of the computationally expensive DeepLabV3+ architecture, with a reduction in both the number of parameters and computational cost in GFLOPs. This lays a solid foundation for our model's future deployment in embedded systems, making it suitable for operation on machines with limited resources.

As observed in Table 7, the network model proposed in this study was compared with other types of networks. Clearly, our network model far surpasses the other network models in terms of FPS, approaching 30. The closer the FPS value is to 30, the better the compatibility of the model with the ORB-SLAM3 algorithm. In terms of the MIoU and MPA, which are indicators of model recognition accuracy, our proposed network forms the best, exceeding the recognition effects of other networks.

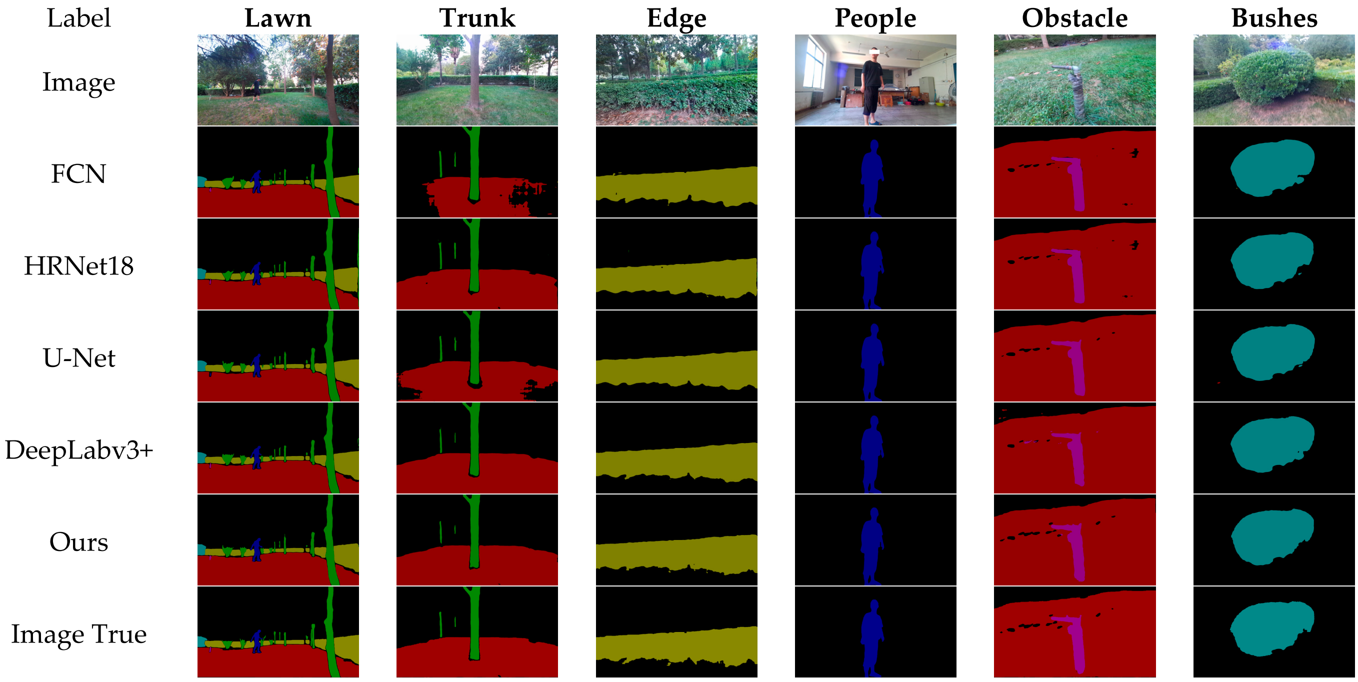


Figure 12. Experimental results and comparison with other methods.

Table 6. Our architecture achieves similar MIoU and MPA values as the computationally expensive DeepLabv3+.

Method	MIoU/%	MPA/%	GFLOPs	Params	FPS
DeepLabv3+	92.28	95.52	176.52	39.31	14.86
Ours	91.07	94.71	68.25	15.10	27.69

Table 7. Comparison with other networks: our network significantly outperforms others in terms of MIoU and MPA.

Method	MIoU/%	MPA/%	GFLOPs	Params	FPS
FCN	89.71	94.66	197.86	44.94	16.25
HRNet18	90.04	94.28	18.60	9.19	20.94
U-Net	90.57	94.69	203.04	27.64	21.21
Ours	91.07	94.71	68.25	15.10	27.69

Although our architecture does not appear to be lighter than HRNet18 in terms of GFLOPs and Params, as shown in Table 7, compared to HRNet18, our method only demonstrates decreases of 1.03% and 0.43% in the MIoU and MPA, respectively. Relatively speaking, although HRNet18 is lighter, it has a lower degree of recognition accuracy and cannot achieve satisfactory results. Moreover, even though it has a lower number of parameters, its FPS improvement is not significant, showing no advantage over our proposed architecture. Considering all indicators, our proposed architecture is the optimal choice, and in terms of integrating a semantic segmentation module into the ORB-SLAM3 environment, our method is the most suitable.

4.5. Lawn Map Construction Experiment

This section validates the method proposed in this paper for constructing a 3D semantic map of a lawn scene through experimentation. The experiment utilized an Intel Realsense D455 depth camera with a resolution of 640 × 480. The program ran on a laptop running Ubuntu 20.04. The depth camera was handheld and walked around the lawn that needed mapping. During the walking process, the camera was continuously swayed from side to side to cover the entire lawn that needed to be captured. Figure 13 illustrates the real-time recognition of the lawn during the walking process.

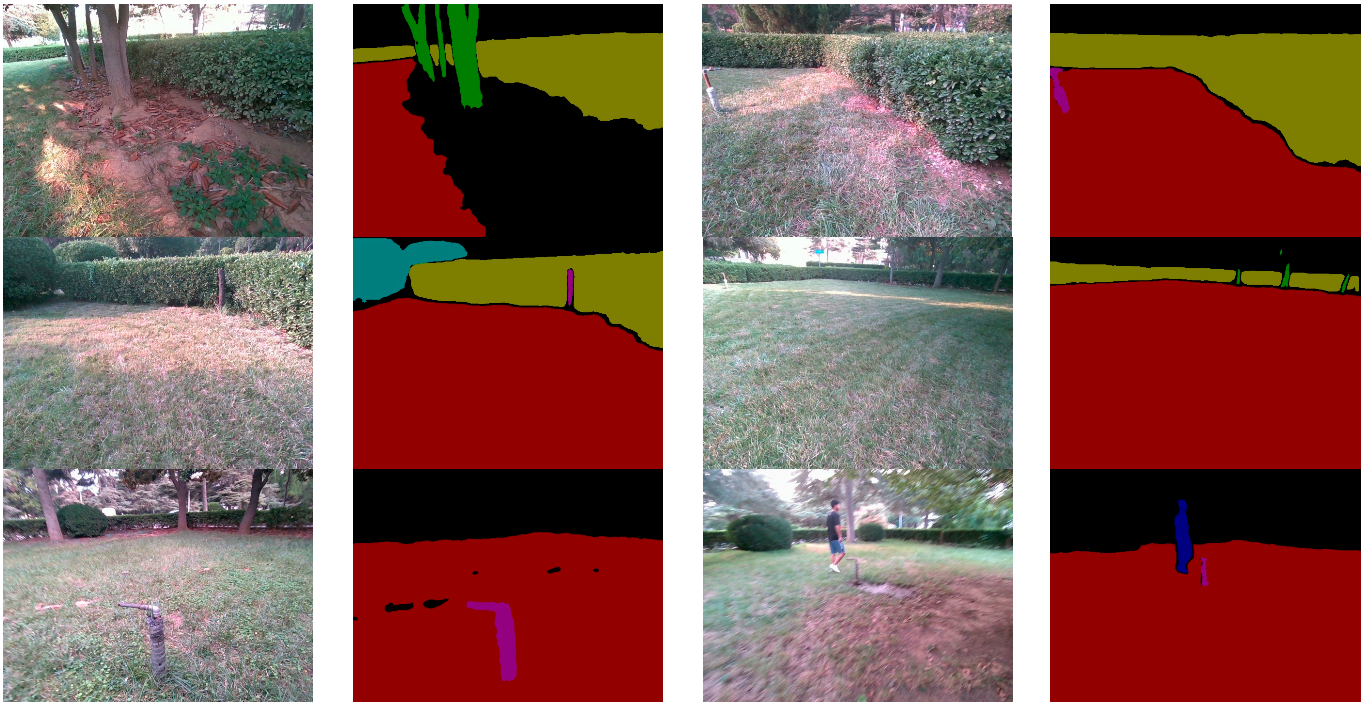


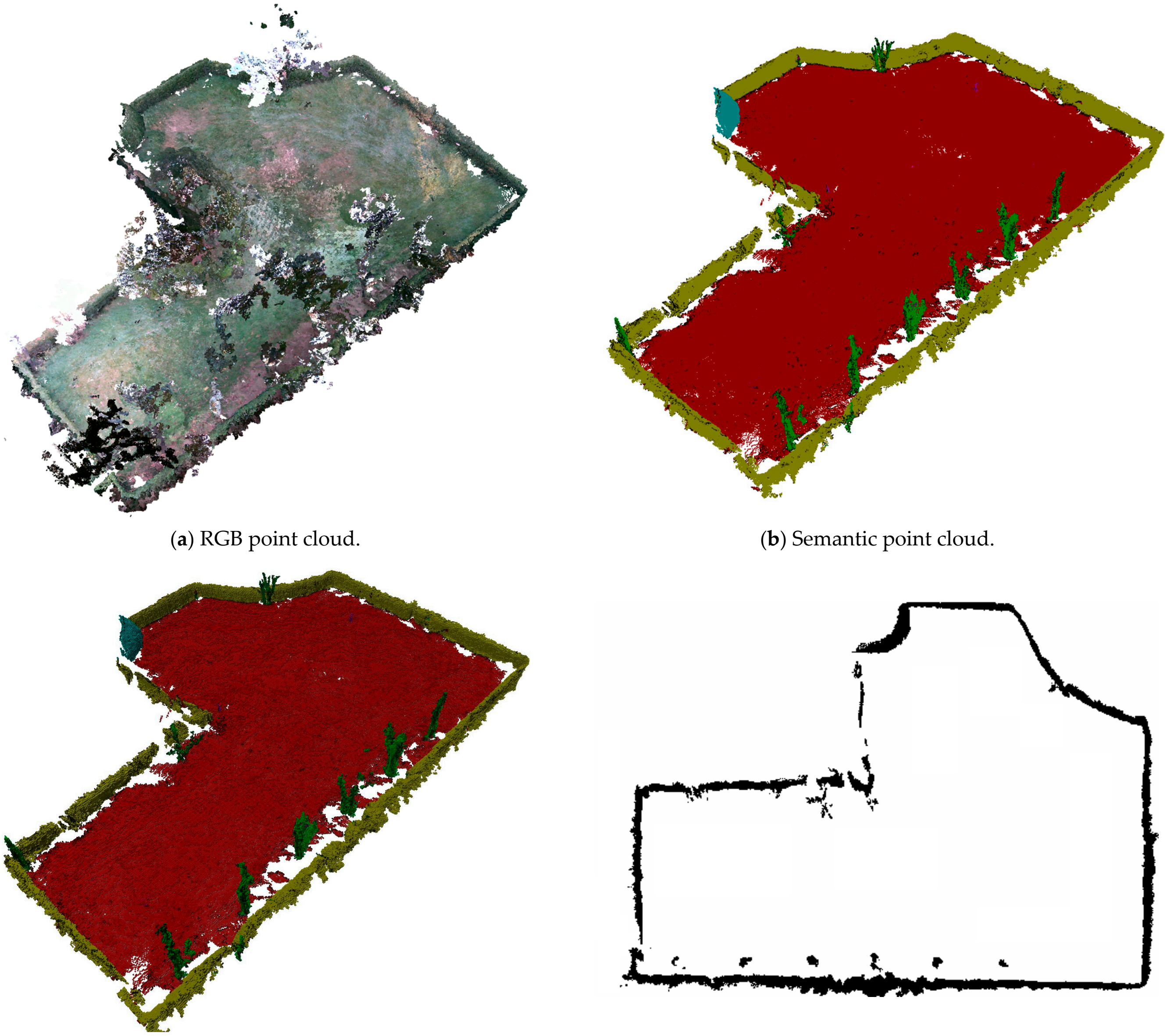
Figure 13. Semantic segmentation of RGB images using DeepLabv3+ in ORB-SLAM3 build maps.

It can be seen in the process of building a 3D semantic map that the architecture proposed in this paper demonstrates relatively good recognition of the working lawn. In general, it can accurately identify all six categories that need to be identified. According to the method mentioned in Section 3.2, the results of the semantic segmentation are added to the ORB-SLAM3 system. Through this system, a 3D semantic point cloud map can be obtained. As shown in Table 8, compared with the original RGB point cloud map, the semantic point cloud map occupies much less space. Since the semantic map constructed in this paper is for the establishment of a working space service for gardening pruning robots, only a semantic map based on the working height of the robot is needed. The crowns at the tops of trees have no effect on the lawn mowing task of the robot, so they are all removed. This makes the saved point cloud map more abundant, and more effective information is saved in the same point cloud map.

Table 8. Point cloud map size comparison results.

Point Cloud File Type	Size of Space/MB
RGB Point Cloud	68.5
Semantic Point Cloud	42.9

As relying solely on a point cloud map is insufficient for path navigation, the 3D semantic point cloud map needs to be converted into an octree map and an occupancy grid map suitable for future three-dimensional and two-dimensional path navigation. Figure 14 illustrates the colored point cloud map, semantic point cloud map, octree map, and occupancy grid map obtained using the MM-ED method proposed in this paper. As the presence of moving individuals in the environment can significantly impact the 3D semantic point cloud map, the point cloud maps established in this experiment exclusively feature scenes without any people.



(**a**) RGB point cloud.

(**b**) Semantic point cloud.

(**c**) Octree map.

(**d**) Occupancy grid map.

Figure 14. Generating a wide variety of maps.

5. Conclusions and Future Work

This paper addresses the construction of a 3D semantic map for a lawn-mower robot operating in an unstructured lawn environment. The proposed multi-object semantic segmentation detection network MM-ED is integrated with the ORB-SLAM3 system to generate a 3D semantic map, facilitating a robust three-dimensional reconstruction of the working area and providing rich environmental information for future robotic weeding operations. To validate the effectiveness of the MM-ED network proposed in this paper, a lawn dataset was created, and comparative experiments with other models were conducted. The main conclusions of this study are as follows: (1) The MM-ED network proposed in this paper utilizes the MobileNetV2 backbone and incorporates the ECA attention mechanism to enhance feature recognition in images. The experimental results demonstrate that compared to other semantic segmentation models with similar MIoU accuracy values, the MM-ED network reduces the model's parameter count and improves its recognition speed. (2) The experimental results indicate that the proposed method performs well in recognizing major objects on a lawn, achieving the satisfactory identification of various items. (3) The method proposed in this paper is easy to deploy, allowing for convenient integration with the ORB-SLAM3 system. In the future, it can be easily migrated to embedded devices, and it effectively generates 3D semantic maps, octree maps, and occupancy grid maps for unstructured lawn environments.

Although the 3D semantic map construction method for unstructured lawn scenes proposed in this paper currently demonstrates effective mapping of the working area, there are certain limitations. The current MM-ED network can recognize humans, but in the context of lawn-mower-robot operation, humans should be treated as dynamic targets and removed to prevent interference with map creation. Additionally, this study only established a semantic point cloud map and obtained octree and occupancy grid maps. Future research could focus on designing working paths for lawn-mower robots based on the acquired maps. Therefore, in future work, we will continue to address these issues and explore innovative solutions.

Author Contributions: Resources, formal analysis, conceptualization, methodology, and funding acquisition: X.X.; validation and writing—original draft preparation: Z.Y.; investigation and methodology: Z.Z.; data curation: Y.Q.; software: H.J.; visualization: M.X.; software: C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Longmen Laboratory "Trendy Industry Projects" (No. LMFKCY2023002), the Luoyang Municipal Public Welfare Special Project (No. 2302031A), and the Science and Technology Tackling Key Problems Project of Henan Province (No. 232102110294).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

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

References

1. Ochman, M.; Skoczeń, M.; Krata, D.; Panek, M.; Spyra, K.; Pawłowski, A. RGB-D odometry for autonomous lawn mowing. In Proceedings of the 20th International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 21–23 June 2021.
2. Wu, M.H.; Yu, J.C.; Lin, Y.C. Study of Autonomous Robotic Lawn Mower Using Multi-Sensor Fusion Based Simultaneous Localization and Mapping. In Proceedings of the 2022 International Conference on Advanced Robotics and Intelligent Systems (ARIS), Taipei, Taiwan, 24–27 August 2022.
3. 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]
4. Baheti, B.; Innani, S.; Gajre, S.; Talbar, S. Semantic scene segmentation in unstructured environment with modified DeepLabV3+. *Pattern Recognit. Lett.* **2020**, *138*, 223–229. [CrossRef]
5. Chen, Y.; Zhang, B.; Zhou, J.; Wang, K. Real-time 3D unstructured environment reconstruction utilizing VR and Kinect-based immersive teleoperation for agricultural field robots. *Comput. Electron. Agric.* **2020**, *175*, 105579. [CrossRef]
6. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
7. Li, X.; Chen, J.; Ye, Y.; Wang, S.; Wang, X. Fast Semantic Segmentation Model PULNet and Lawn Boundary Detection Method. In Proceedings of the 2020 International Symposium on Automation, Information and Computing (ISAIC), Beijing, China, 2–4 December 2020.
8. Sportelli, M.; Martelloni, L.; Orlandi, A.; Pirchio, M.; Fontanelli, M.; Frasconi, C.; Raffaelli, M.; Peruzzi, A.; Consorti, S.B.; Vernieri, P. Autonomous mower management systems efficiency improvement: Analysis of greenspace features and planning suggestions. *Agriculture* **2019**, *9*, 115. [CrossRef]
9. Mo, Y.; Wu, Y.; Yang, X.; Liu, F.; Liao, Y. Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing* **2022**, *493*, 626–646. [CrossRef]
10. Yan, C.; Chen, Z.; Li, Z.; Liu, R.; Li, Y.; Xiao, H.; Lu, P.; Xie, B. Tea sprout picking point identification based on improved deepLabV3+. *Agriculture* **2022**, *12*, 1594. [CrossRef]
11. Shi, L.; Wang, G.; Mo, L.; Yi, X.; Wu, X.; Wu, P. Automatic segmentation of standing trees from forest images based on deep Learning. *Sensors* **2022**, *22*, 6663. [CrossRef]
12. Feng, G.; Wang, H.; Chen, M.; Liu, Z. Accurate Segmentation of Tilapia Fish Body Parts Based on Deeplabv3+ for Advancing Phenotyping Applications. *Appl. Sci.* **2023**, *13*, 9635. [CrossRef]

13. Cai, C.; Tan, J.; Zhang, P.; Ye, Y.; Zhang, J. Determining Strawberries' Varying Maturity Levels by Utilizing Image Segmentation Methods of Improved DeepLabV3+. *Agronomy* **2022**, *12*, 1875. [CrossRef]
14. Zheng, T.; Li, B.; Yao, J. LHRNet: Lateral hierarchically refining network for salient object detection. *J. Intell. Fuzzy Syst.* **2019**, *37*, 2503–2514. [CrossRef]
15. Chen, C.; Shen, P. Research on Crack Width Measurement Based on Binocular Vision and Improved DeeplabV3+. *Appl. Sci.* **2023**, *13*, 2752. [CrossRef]
16. Cui, X.; Lu, C.; Wang, J. 3D semantic map construction using improved ORB-SLAM2 for mobile robot in edge computing environment. *IEEE Access* **2020**, *8*, 67179–67191. [CrossRef]
17. Chen, Y.; Xiong, Y.; Zhang, B.; Zhou, J.; Zhang, Q. 3D point cloud semantic segmentation toward large-scale unstructured agricultural scene classification. *Comput. Electron. Agric.* **2021**, *190*, 106445. [CrossRef]
18. Yajima, Y.; Kahoush, M.; Kim, S.; Chen, J.; Park, J.; Kangisser, S.; Irizarry, J.; Cho, Y.K. AI-Driven 3D Point Cloud-Based Highway Infrastructure Monitoring System Using UAV. *Comput. Civ. Eng.* **2021**, *2021*, 894–901.
19. Koch, T.; Körner, M.; Fraundorfer, F. Automatic and semantically-aware 3D UAV flight planning for image-based 3D reconstruction. *Remote Sens.* **2019**, *11*, 1550. [CrossRef]
20. Zhang, C.; Liu, Z.; Liu, G.; Huang, D. Large-scale 3d semantic mapping using monocular vision. In Proceedings of the 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), Xiamen, China, 5–7 July 2019.
21. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the 15th European Conference Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
22. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
23. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
24. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *721*, 8026–8037.
25. Kazerouni, I.A.; Fitzgerald, L.; Dooly, G.; Toal, D. A survey of state-of-the-art on visual SLAM. *Expert Syst. Appl.* **2022**, *205*, 117734. [CrossRef]
26. Huang, L. Review on LiDAR-based SLAM techniques. In Proceedings of the 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML), Stanford, CA, USA, 14 November 2021.
27. Shu, F.; Lesur, P.; Xie, Y.; Pagani, A.; Stricker, D. SLAM in the field: An evaluation of monocular mapping and localization on challenging dynamic agricultural environment. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2021.
28. Liu, L.; Liu, Y.; Lv, Y.; Li, X. A Novel Approach for Simultaneous Localization and Dense Mapping Based on Binocular Vision in Forest Ecological Environment. *Forests* **2024**, *15*, 147. [CrossRef]
29. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardos, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
30. Hou, J.; Goebel, M.; Hübner, P.; Iwaszczuk, D. Octree-Based Approach for Real-Time 3D Indoor Mapping Using RGB-D Video Data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *48*, 183–190. [CrossRef]
31. Xu, L.; Feng, C.; Kamat, V.R.; Menassa, C.C. An occupancy grid mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments. *Autom. Constr.* **2019**, *104*, 230–245. [CrossRef]
32. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the 18th International Conference Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015.
33. Sun, K.; Xiao, B.; Liu, D.; Wang, J. Deep high-resolution representation learning for human pose estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
34. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
35. YatengLG, Alias-z, Horffmanwang. ISAT with Segment Anything: Image Segmentation Annotation Tool with Segment Anything [EB/OL]. 2023. Open Source Software. Available online: https://github.com/yatengLG/ISAT_with_segment_anything (accessed on 29 January 2024).
36. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.-Y.; et al. Segment Anything. *arXiv* **2023**, arXiv:2304.02643.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
38. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.

39. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the 15th European Conference Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
40. Zhang, C.; Lin, G.; Liu, F.; Yao, R.; Shen, C. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.

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.