Applied Intelligence (2023) 53:23336–23348
https://doi.org/10.1007/s10489-023-04742-x

Bunet: An effective and efficient segmentation method based on bilateral encoder-decoder structure for rapid detection of apple tree branches

Shanshan Zhang[1] · Hao Wan[1] · Zeming Fan[1] · Xilei Zeng[1] · Ke Zhang[1]

Accepted: 28 May 2023 / Published online: 8 July 2023

Abstract
Automatic apple harvesting robots have received much research attention in recent years to lower harvesting costs. A fundamental problem for harvesting robots is how to quickly and accurately detect branches to avoid collisions with limited hardware resources. In this paper, we propose a lightweight, high-accurate and real-time semantic segmentation network, Bilateral U-shape Network (BUNet), to segment apple tree branches. The BUNet consists mainly of a U-shaped detail branch and a U-shaped semantic branch, the former for capturing spatial details and the latter for supplementing semantic information. These two U-shape branches complement each other, keeping the high accuracy of the Encoder-decoder Backbone while maintaining the efficiency and effectiveness of the Two-pathway Backbone. In addition, a Simplified Attention Fusion Module (SAFM) is proposed to effectively fuse different levels of information from two branches for pixel-by-pixel prediction. Experimental results show (on our own constructed dataset) that BUNet achieves the highest Intersection over Union (IoU) and F1-score of 75.96% and 86.34%, respectively, with minimum parameters of 0.93M and 11.94G Floating-point of Operations (FLOPs) in branch segmentation. Meanwhile, BUNet achieves a speed of 110.32 Frames Per Second (FPS) with input image size of 1280×720 pixels. These results confirm that the proposed method can effectively detect the branches and can, therefore, be used to plan an obstacle avoidance path for harvesting robots.

Keywords Fruit harvesting robots · Semantic segmentation · Branch segmentation · U-shape structure · Two-pathway backbone

1 Introduction

Apples are one of the most important fruits in the world. In 2020, the world's total apple production was 86 million tons, ranking fourth in global fruit production. Currently, the dominant method of harvesting fresh apples is manual harvesting, which is time-consuming and labour-intensive. Research is being conducted on harvesting robots to replace human labor [1, 2]. One of the most fundamental and major research components of harvesting robots is computer vision technology. In the field of robotic harvesting, existing related work focuses on apple detection, but rarely on branch detection [3]. As reported in [4] and [5], thick branches of apple trees can prevent the robotic arm from approaching the target apples, leading to a lower harvest success rate and an increasing risk of damage to the robotic arm. Thus, accurate detection of apple tree branches is also the integral part that can provide environmental information to the harvesting robot for dynamic planning and obstacle avoidance strategies.

As a thriving technology in recent years, Convolutional Neural Network (CNN) (mainly object detection and semantic segmentation) has been widely used for image processing in agriculture [6, 7]. Due to the lack of pixel by pixel classification information, the accuracy of the branch detection algorithm based on object detection is relatively low, and

Shanshan Zhang and Hao Wan were both co-first authors.

✉ Zeming Fan
zmfan_npu@163.com

Shanshan Zhang
158961769@qq.com

Hao Wan
wanhaoh@126.com

Xilei Zeng
zengxilei@mail.nwpu.edu.cn

Ke Zhang
zhangkezk88@163.com

[1] School of Automation, Northwestern Polytechnical University, Xi'an 710100, Shaanxi Province, China

extensive subsequent processing is required. While semantic segmentation, which classifies each pixel in an image, can preserve the spatial information of objects more completely [8], it makes up for the shortcomings of object detection in branch detection.

Over decades, a series of competitive segmentation algorithms have emerged, such as Fully Convolutional Networks (FCN [9]), Deeplab series [6, 10, 11], Mask-RCNN [12]. Algorithms mentioned above have been gradually applied to branch segmentation [4, 13]. Lin et.al [14] use FCN to segment guava branches with a detection accuracy of 0.983, suggesting that semantic segmentation performs well for branch prediction. After that, the authors propose tiny Mask R-CNN in a subsequent paper [15], which greatly reduce the size of the model and shorten the inference time. However, while Deeplabv3 [11] has high segmentation accuracy when used to segment RGB images of litchi into background, fruit and branch, it is computationally expensive and therefore not suitable for real-time applications [16]. Notably, Kang and Chen [17] propose DaSNet based on residual network structure for real-time detection and semantic segmentation of apples and branches. Experimental results demonstrate that semantic segmentation is well suited for real-time branch segmentation.

It is noted that current harvesting robots typically use mobile devices with relatively limited hardware resources [18, 19]. Therefore, lightweight semantic segmentation models are essential to ensure their portability on mobile devices. However, the semantic segmentation algorithms applied in the agricultural domain mentioned above pay less attention to computational cost and inference speed. Hence, a more lightweight semantic segmentation model that can fulfill the real-time segmentation requirements of outdoor mobile scenarios with limited devices is required.

The key strategies for constructing lightweight models are to devise lightweight modules (usually based on depthwise separable convolution and dilation convolution) and to enhance the network structure. For example, ESPNet [20] decomposes the standard convolution into point-wise convolution and spatial pyramid of dilated convolution, which can effectively reduce the number of parameters and computation while maintaining a large perceptual field. Further, as an updated version, ESPNetV2 [21] introduces an EESP module based on group point-wise convolutions and depthwise dilated separable convolutions, which further reduces the number of parameters. PP-LiteSeg [22] does not use a lightweight convolutional module to extract features, but mitigates the redundancy of the decoder by reducing the number of channels in the decoding process. By changing the interaction between high and low resolution from series to parallel, HRNetV2 [23] is able to maintain high resolution representation and thus improve the accuracy of spatial representation and detail information. To meet the needs of different scenarios, the authors provide several models with different volumes, such as HRNetV2-W48 and HRNetV2-W18-small, the latter of which is very suitable as a backbone for lightweight semantic segmentation. The Transformer-based Topformer [24] uses some stacked lightweight MobileNetV2 blocks and Fast Down-Sampling strategy to build Token Pyramid, thus reducing the number of parameters. In addition to the common U-shaped encoder-decoder architecture, there is also the two-pathway architecture which is well suitable for semantic segmentation tasks, one pathway for extracting semantic information, the other shallow one provides rich spatial details as a supplement. One representative network is BiSeNet [25], which uses a bi-directional contextual information flow module to capture global contextual information, and achieves network lightweighting by reducing the overall number of channels and downsampling times of the network. Based on BiSeNet, STDC [26] removes the potentially redundant spatial path and adds a Short-Term Dense Concatenate module to extract the underlying features to speed up inference. Similarly, BiSeNetV2 [27] is also an updated version of BiSeNet, designed with a lightweight semantic branch based on depthwise separable convolution, which greatly reduces the number of parameters.

Our model is based on two key observations: first, the semantic segmentation task demands both rich detail and spatial information as well as contextual semantic information from large perceptual fields. Second, the two-pathway architecture, in contrast to the U-shaped encoder-decoder architecture, directly utilizes small-resolution feature maps and upsamples them at high magnification to segment the image, which results in the loss of low-level details and the introduction of error information. Therefore, we propose the Bilateral U-Shape Network (BUNet), which is composed of three parts: 1) U-shape Detail Branch, 2) U-shape Semantic Branch, and 3) Simplified Attention Fusion Module. The main contributions are summarized as follows:

I A lightweight, efficient and highly accurate bilateral encoder-decoder architecture is proposed for real-time segmentation of apple tree branches, which obtains rich low-level spatial details and high-level semantic information through two U-shaped branches, respectively.
II A novel fusion module, i.e., Simplified Attention Fusion Module is designed to fuse the features of two different levels, which can effectively improve the segmentation accuracy and hardly introduce additional parameters and computation.
III We collect a complex and comprehensive dataset of apple tree branches, including different light and weather conditions, and annotate the dataset in great detail.

The paper is organized as follows. Section 2 describes our novel architecture consists of UDB, USB and SAFM.

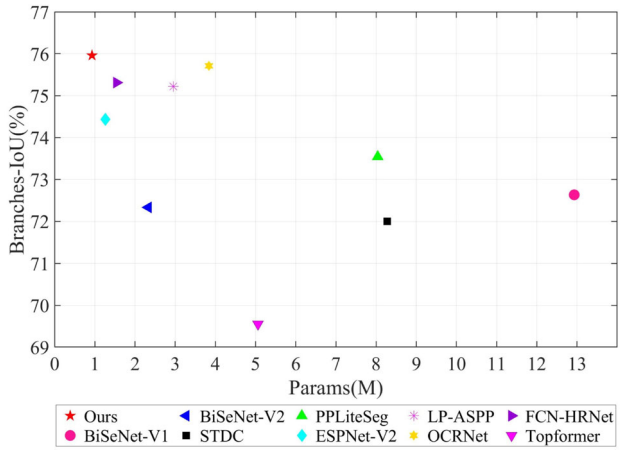

Fig. 1 Params-Accuracy performance comparison of different network. Experiments are performed on 9 different neural networks on our constructed branches dataset

Section 3 introduces our own constructed dataset. The experiments and discussion are in Section 4 and Section 5 concludes the work.

2 Methodology

2.1 The overall structure of bunet

Our BUNet architecture consists of the U-shape Detail Branch (UDB) (Section 2.2) and the U-shape Semantic Branch (USB) (Section 2.3), which are combined by a Simplified Attention Fusion Module (SAFM) (Section 2.4).

Unlike previous large networks [6, 11, 28] with hundreds of layers and numerous channels, our BUNet has relatively few layers and channels, which is very beneficial for real-time semantic segmentation tasks on mobile devices. The overall structure of the BUNet is shown in Fig. 2. Table 1 shows the detailed configuration. The UDB uses wide channels and shallow layers to extract diverse features and rich low-level spatial information. The USB, on the other hand, is a structure with low channel capacity and deeper layers, aiming to capture high-level semantics. Meanwhile, A lightweight Encoder-Decoder structure based on Gather-and-Expansion Layer [27] is constructed in USB to reduce the parameters while not losing too much feature information. As for the feature fusion module, SAFM is proposed to effectively fuse sufficient spatial information with accurate semantic information to obtain the final prediction map.

2.2 The U-shape detail branch (UDB)

This branch aims to capture low-level spatial and detail information for accurate position prediction, requiring large scale feature maps with multiple channels, which are capable of presenting richer detail information. Thus, the UDB is designed to have high channel capacity and shallow layers (only 10 layers), the former to represent various features and the latter to ensure efficiency. The exact structure of the UDB is shown in Table 1 and Fig. 2. We try four backbones in Experiment 4.2.1: HRNetV2-W18-small, MobilenetV3-small, STDC1 and UDB. The UDB has the best segmentation effect in our model.

The UDB is an encoder-decoder structure. In the encoder part, three shallow convolution blocks (DEB1, DEB2 and

Table 1 Details of the UDB and the USB. Each operation (*opr*) has a kernel size *k*, stride *s* and output channels *c*, repeated *r* times, generating different output *size*s

Stage	U-shape Detail Branch							U-shape Semantic Branch						
	block	*opr*	*k*	*c*	*s*	*r*	*size*	block	*opr*	*k*	*c*	*s*	*r*	*size*
								MSCD		3	8	4	1	256×128
Encoder	DEB1	Conv2d	3	32	2	1	512×256	SEB1	GE2	3	16	2	1	128×64
		Conv2d	3	32	1	1	512×256		GE1	3	16	1	1	128×64
	DEB2	Conv2d	3	64	2	1	256×128	SEB2	GE2	3	32	2	1	64×32
		Conv2d	3	64	1	2	256×128		GE1	3	32	1	1	64×32
	DEB3	Conv2d	3	128	2	1	128×64	SEB3	GE2	3	64	2	1	32×16
		Conv2d	3	128	1	1	128×64		GE1	3	64	1	3	32×16
Decoder	DDB1	Conv2d	3	128	1	1	128×64	SDB1	GE3	3	32		1	32×16
									2×Up		32		1	64×32
	DDB2	Conv2d	3	64	1	1	128×64	SDB2	GE3	3	16		1	64×32
		2×Up		64		1	256×128		2×Up		16		1	128×64
	DDB3	Conv2d	3	32	1	1	256×128	SDB3	GE3	3	8		1	128×64
		2×Up		32		1	512×256		2×Up		8		1	256×128

"Conv2d" donates convolutional layer, followed by a batch normalization and ReLu activation function."2×up" donates bilinear interpolation for 2 times upsampling. GE1, GE2 and GE3 contribute the specific components of Encoder and Decoder based on Expansion Layer described in Section 2.3.2

DEB3 in Fig. 2) are constructed, Each consisting of several standard convolution layers. The role of each convolution block is to double the number of channels and halves the size of the feature map. After being encoded by UDB, the generated feature map is 1/8 of the original image. Three sets of convolutional blocks, DDB1, DDB2 and DDB3, are used in the decoding process. However, compared to the encoder stage, the number of standard convolution layers of convolution blocks is reduced to 1, and stride=1. After DDB2 and DDB3, the input feature map is upsampled using bilinear interpolation. To avoid introducing too many parameters, we also use a point-wise fast summation approach, which directly fuses the corresponding feature maps of encoder and decoder, effectively capturing the context information and the features of receptive fields at different scales. Finally, the input image is restored to a high-resolution feature map of 1/2 of the original image (Fig. 2).

2.3 The U-shape semantic branch(USB)

The USB aims to obtain high-level semantic information based on a large receptive field. Therefore, it is designed as a deep-level structure for expanding the receptive field. Since spatial information is provided by the UDB, low channel capacity is reasonable for USB. To further reduce the number of parameters without losing too much feature information, we apply the Gather-and-Expansion Layer (GE) to the Encoder and Decoder section, i.e. GE1, GE2 and GE3 (see Section 2.3.2 and Fig. 5 for the detailed design). In the following part of the section, we demonstrate the core module of the branch as illustrated in Fig. 3.

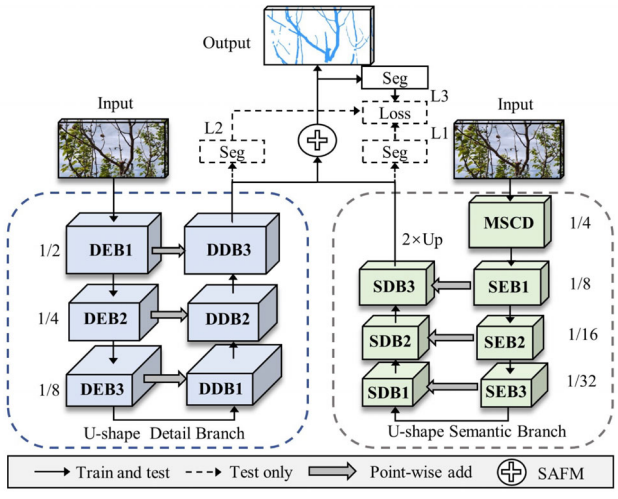

Fig. 2 Overview of the Bilateral U-shape Network. "DEBx" and "DDBx" donate different blocks in the Encoder and Decoder stage of the UDB, respectively. Similarly, "SEBx" and "SDBx" represent different blocks of the USB. "MSCD" is Max-pooling and Separable Convolution based Down-sampling module described in Section 2.3.1. "Seg" donates the segmentation head consisting of a 3 × 3 convolutional layer followed by a 1 × 1 convolutional layer. L1, L2 and L3 represent different losses of multi-loss strategy. "2×up" donates bilinear interpolation for 2 times upsampling

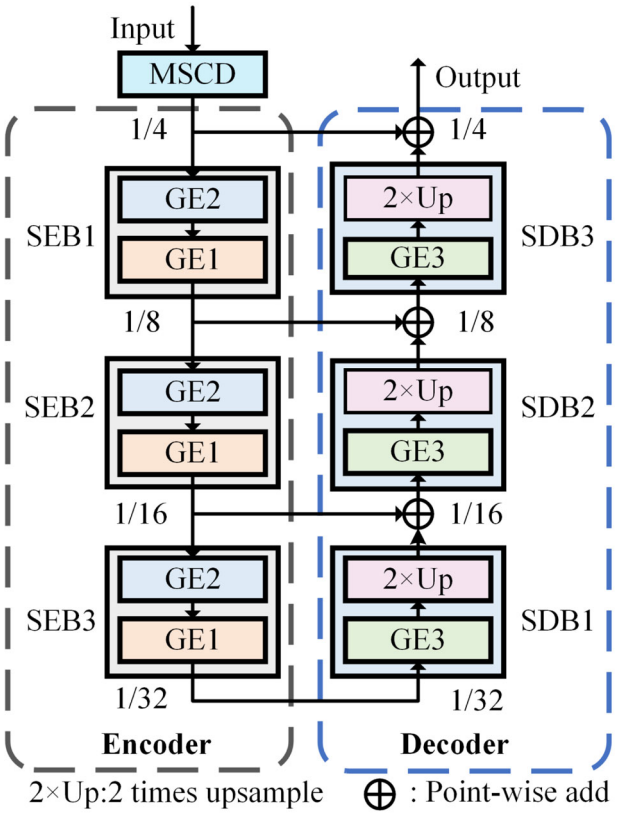

Fig. 3 Detailed design of the USB. "MSCD" is Max-pooling and Separable Convolution based Down-sampling module described in Section 2.3.1. "SEBx" and "SDBx" denote the basic blocks of the encoding and decoding stages, respectively. The structures of "GE1", "GE2" and "GE3" are showed in Fig. 5(a), (b) and (c), respectively

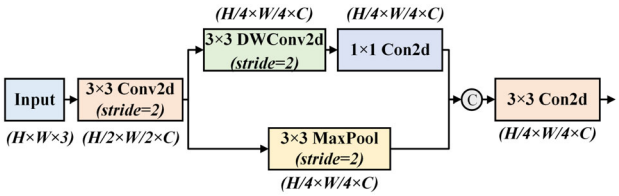

Fig. 4 Detailed design of MSCD."Conv2d" donates convolutional layer, followed by a batch normalization and ReLu activation function. "Conv" is a simple convolutional layer. "DWConv2d" means a depth-wise convolution followed by a batch normalization and ReLu activation function. "3 × 3" and "1 × 1" donate kernel size. "$H \times W \times C$" donates the height, width and channel of the tensor. "C" donates concatenate

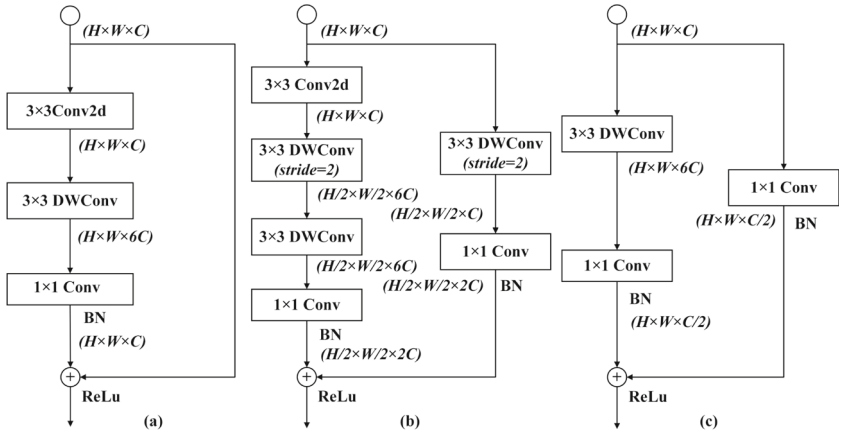

Fig. 5 Detailed design of Encoder and Decoder based on Expansion Layer. (a), (b) and (c) demonstrate the detail structure of GE1, GE2 and GE3, respectively. "Conv2d" donates convolutional layer, followed by a batch normalization and ReLu activation function. "Conv" is a simple convolutional layer. "DWConv" donates a depth-wise convolution followed by a batch normalization. "MaxPool" donates max-pooling. "3 × 3" and "1 × 1" donate kernel size. "$H \times W \times C$" donates the height, width and channel of tensor. "+" is point-wise add

2.3.1 Max-pooling and separable convolution based down-sampling(MSCD)

We design MSCD before the encoder, taking a fast down-sampling strategy to promote the level of the feature representation and quickly enlarge the receptive field, without introducing too many additional parameters. The specific structure is shown in Fig. 4. Instead of traditional convolution, we use depth-wise separable convolution instead to reduce the number of parameters, and maximum pooling to extract edge and texture information. Both strategies are used to downsample the input image simultaneously to increase the diversity of features.

2.3.2 Encoder-decoder based on GE modules

For the encoder part, we use the Gather-and-Expansion Layer(GE) modules (BiSENetV2 [27]), which consists of GE1 and GE2, and we design a brand-new structure namely GE3 based on it, as shown in the Fig. 5(c). The functions of GE1, GE2 and GE3 are: (1) to maintain the number of channels and the size of the feature map, (2) to double the number of channels and halve the size of the feature map, and (3) to halve the channels and maintain the size of the feature map. The core conception is using depth-wise separable convolution to reduce parameters, in the meantime, expanding the number of channels by a factor of n (set to 6 here) to prevent losing too much feature information. (When the kernel size is 3 × 3, the computational cost is 8 to 9 times smaller than that of standard convolutions at only a small reduction in accuracy [29]).

2.4 Simplified attention fusion module (SAFM)

To further improve segmentation accuracy and enhance the ability of network to focus on critical regions without significantly increasing parameters, we propose the SAFM as a fusion module. The Convolutional Block Attention Module (CBAM [30]) is one of the widely used methods to improve accuracy in semantic segmentation tasks. However, the introduction of CBAM significantly reduces the inference speed and increases some parameters. Based on this, we design 1 × 1 convolution layers and Sigmoid activation function as a simplification of the attention module in CBAM.

The detailed structure is shown in Fig. 6. We first concatenate the two branches and then further extract features with a 3 × 3 convolution layer. After that, a 1 × 1 convolutional layer is used to adjust the number of channels. Next, the global maximum pooling and the global average pooling are utilized to augment the feature representation of the focal region and global information. Then, two 1 × 1 convolution layers, followed by the Sigmoid activation functions to increase nonlinearity, are used as a simplified fully connected layer to obtain the channel weights. Also, to reduce the loss of information, the channel weighted feature maps obtained by multiplying with the channel weights are added with the adjusted feature maps, generating the channel augmented feature maps as the final output.

3 Experimental materials

A total of 1,035 images are collected from an apple orchard in northern Shaanxi Province, China, with a resolution of

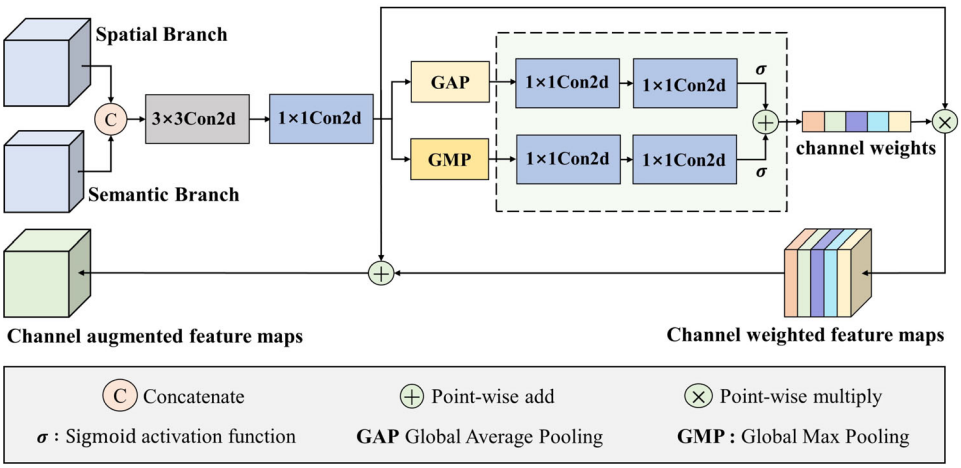

Fig. 6 Detailed design of the Simplified Attention Fusion Module. "Conv2d" donates convolutional layer, followed by one batch normalization and ReLu activation function."3 × 3" and "1 × 1" donate the kernel size of convolutions

1,280x720, and saved as JPEG. Images are captured from April to June 2022, from 9:00 a.m. to 6:00 p.m. Over 200 apple trees are photographed at random angles to better suit the actual harvesting scene, with no more than 5 images per tree. Keep the shooting distance of 0.5m-1.5m, which is the typical range of harvesting robots. The apple trees are between 2 and 2.5m high, the row spacing is of about 4m (as shown in Fig. 7).

In the actual orchard scene, harvesting robots must face complex scenarios of varying illumination and changing harvesting angles. Hence, the camera's viewing directions are random, for example, are set with parallel, antiparallel and perpendicular to the direction of sunlight to simulate forward lighting, backward lighting and side lighting, respectively (the top raw of Fig. 8). The camera angle is different to simulate various harvesting angles, including horizontal harvesting, upward harvesting, sideways harvesting, etc. (the bottom of Fig. 8).

828 images in the dataset are used as the training set and the remaining 207 as the test set. Using Photoshop's magnetic lasso tool, we meticulously label all the visible branches. Fig. 9(a) shows the original image sample; Fig. 9(b) shows the annotated images with Photoshop; and Fig. 9(c) displays the corresponding final pseudo-colour annotation. Once the labelling is complete, we only perform data transformations and data enhancements on the training set before the training, specifically including RandomPaddingCrop, RandomDistort, RandomHorizontalFlip and Normalize.

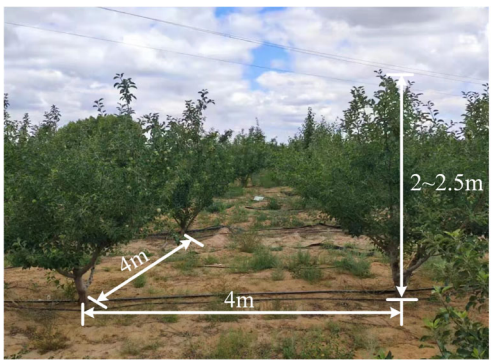

Fig. 7 Actual orchard scene

4 Results and discussion

4.1 Experiment detail

4.1.1 Training setting

We train all models for 40000 iterations on one RTX3090 GPU card with a batch size of 6 and a crop size of 1024×512 for the input. We use the Adam algorithm with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to optimize the models. The initial learning rate is set to 0.05 with a 'poly' learning rate strategy, in which the initial rate is multiplied by $(1 - iter/iters_{max})^{power}$. We adopt multi-loss training strategy as $L1$, $L2$, $L3$ in Fig. 2 to improve the segmentation accuracy. The cross-entropy loss is used as the loss function and can be calculated by Eq. 1 for the dichotomous task in this paper. The deep learning framework Paddle (PaddleSeg-2.5.0 [31]) is employed for all experiments.

$$L = -\left[y log \check{y} + (1 - y) log\left(1 - \check{y}\right)\right] \tag{1}$$

Where L represents the loss function, y and $\check{y}$ represents real labels and predicted labels, respectively.

4.1.2 Evaluation criterion

The following semantic segmentation indicators are used for evaluation: intersection over union (IoU), F1-score, number of parameters (Total Params), Floating-point of Operations (FLOPs) and Frames Per Second (FPS). We use the method

Fig. 8 Examples of images of our own constructed branches datasets. (a), (b) and (c) are examples of forward lighting, backward lighting and side lighting, respectively. (d), (e) and (f) stimulate the horizontal harvesting, upward harvesting and sideways harvesting, respectively

provided by the official Paddle to calculate the Total params, FPS and FLOPs of the evaluated models. The IoU and F1-score can be calculated by Eq. 2 and Eq. 3-5. Higher values of IoU and F1-score indicate better segmentation performance.

$$IoU = TP/(FN + FP + TP) \times 100\% \tag{2}$$

$$recall = TP/(TP + FN) \times 100\% \tag{3}$$

$$precision = TP/(TP + FP) \times 100\% \tag{4}$$

$$F1 = \frac{2 \times pression \times recall}{(pression + recall)} \times 100\% \tag{5}$$

Where TP, FP, FN represent true positive, false positive, and false negative, respectively.

4.2 Ablation experiments

4.2.1 Different backbones in Detail Branch

To evaluate the effectiveness of UDB as a detail branch of BUNet, we design the control experiment using three lightweight backbone networks: HRNetV2-W18-small [23], MobilenetV3-small, and STDC1 [26]. We select the last feature map of HRNetV2-W18-small, the first feature map of MobilenetV3-small, and the second feature map of STDC1 as the output of the detail branch, and ensure that they have the same size. Then use a convolutional layer to adjust the number of channels of these feature maps to 32, and fuse them with USB to generate the final output. All four experiments are conducted under the same settings. The experimental results are shown in Table 2. From these results, it can be seen that the BUNet has the least number of parameters when using UDB as the detail branch, which is 34.41% less than MobileNetV3-small as the detail branch. Additionally, the segmentation effect is the best (Branches-IoU improves by 2.81%, 1.65% and 2.62% respectively). These results prove that UDB has a positive effect on BUNet.

4.2.2 Different feature fusion modules

In order to verify the effectiveness of our proposed Simplified Attention Fusion Module (SAFM), we compare four different fusion methods for the two-way branch features: direct concatenation, direct addition, CBAM-based fusion, and SAFM-based fusion. Table 3 shows that SAFM obviously improves the Branches-IoU (1.42%, 2.67% and 1.84% improvements, respectively) with almost no increase in the number of parameters than direct concatenating or addition, a slight increase in computation and inference time. This indicates that SAFM is actually beneficial in fusing different levels of features, focusing on the key regions of the feature maps and achieving better segmentation results.

4.2.3 Channel capacity of the USB

To further reduce the amount of parameters and improve the efficiency of the model, we assume that the ratio of the number of channels of the USB and the UDB is λ ($\lambda < 1$). We compare the performance of the model for several values of λ. As shown in Table 4, different values of λ bring in different extents of improvement on the segmentation accu-

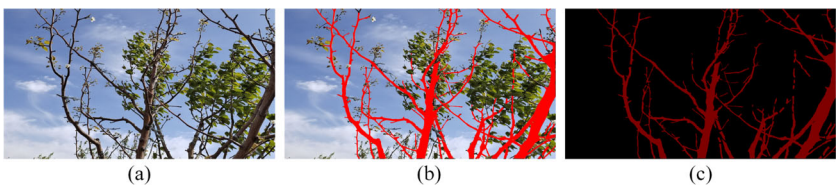

Fig. 9 Illustrations of the labelling process

Table 2 Ablation study results of lightweight backbones in UDB

Backbone	Publish	Branches-IoU(%)	Parmas(M)
MobileNetV3_small	ICCV2019	73.15	1.25
HRNet-W18-small	arXiv2019	74.31	1.92
STDC1	CVPR2021	73.34	5.72
UDB		**75.96**	**0.93**

Table 3 Ablation study results of fusion modules

	Branches-IoU(%)	Total params(M)	FLOPs(G)	FPS ($frames/s$)
Concatenate	74.54	0.92	11.79	116.96
Add	73.29	**0.90**	**9.16**	**117.35**
CBAM	74.12	1.06	11.99	90.24
SAFM	**75.96**	0.93	11.94	110.32

Table 4 Ablation study results of the ratio λ of channels of the UDB and the USB

	Branches-IoU(%)	FLOPs(G)	Total params(M)
Detail-only	73.03	**11.08**	**0.59**
$\lambda = 1/2$	75.26	13.32	1.87
$\lambda = 1/4$	**75.96**	11.94	0.93
$\lambda = 1/8$	75.17	11.49	0.68

racy compared to retaining only the UDB. Branches-IoU is highest at $\lambda = 1/4$ with 75.96% (2.93% improvement), and even at $\lambda = 1/8$ with only 4 channels in the first layer of the USB, it brings in 2.14% improvement, which is inspiring. We take $\lambda = 1/4$ as the default parameter, which has the highest Branches-IoU. Moreover, the computational complexity (0.86G more) and the number of parameters (0.34M more) are acceptable compared to using the UDB alone.

4.2.4 Visual results of extracted features by UDB, USB and SAFM

In Fig. 10, we visualize the features of UDB, USB and their combined results (fused by SAFM). The UDB provides a wealth of detailed information, which is mainly low-level information, including branch outlines and boundaries, the shapes of leaves, and spatial location relationships (1, 3, 5 in

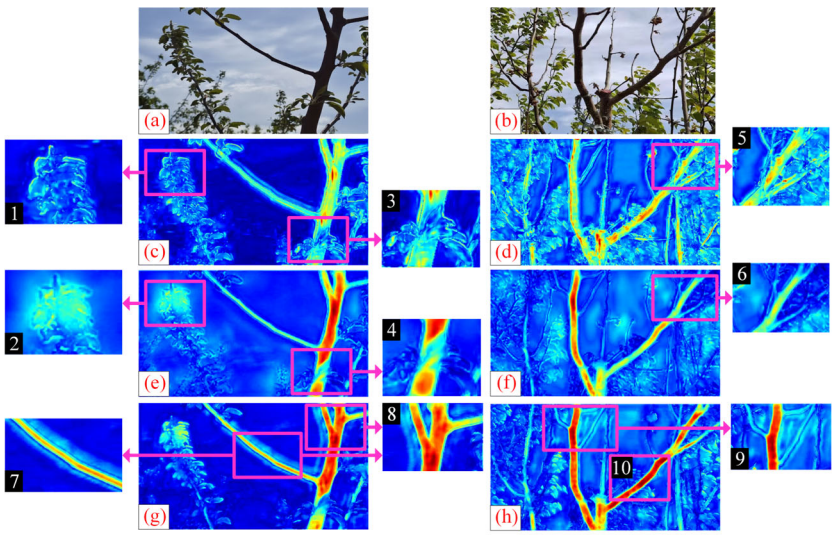

Fig. 10 Visual comparison of the feature maps of UDB, USB and fusion output. (a), (b) Original inputs; (c), (d) The visual feature maps of UDB; (e), (f) The visual feature maps of USB; (g), (h) The visual feature maps of output fused by SAFM. 1, 3 and 5 show detailed information extracted by UDB; 2, 4 and 6 show semantic context obtained by USB, without abundant features; 7~10 show complete branch acquired by feature maps fused by SAFM

Table 5 Comparisons of the achieved accuracy between BUNet and other state-of-the-art models on our apple tree branches dataset

Model	Backbone	Publish	Branches Precision(%)	Branches Recall(%)	Branches F1-score(%)	Background -IoU(%)	Branches -IoU (%)	Mean -IoU(%)
BiSENet-V1	ResNet18	ECCV2018	86.59	81.82	84.14	98.08	72.63	85.36
BiSENet-V2		IJCV2021	**87.97**	80.08	83.84	98.13	72.33	85.23
STDC	STDC1	CVPR2021	86.09	81.48	83.72	98.07	72.00	85.08
PPLite-Seg	STDC1	arXiv2022	87.55	82.13	84.57	98.20	73.54	85.88
ESPNet-V2		CVPR2019	84.65	**86.05**	85.34	98.19	74.43	86.31
LR-ASPP	MobileNet V3-small	ICCV2019	87.78	85.43	85.86	98.27	75.22	86.75
OCRNet	HRNet-W18-small	arXiv2019	86.93	85.92	86.17	**98.29**	75.71	87.00
FCN-HRNet	HRNet-W18-small	arXiv2019	85.93	85.91	85.92	98.24	75.31	86.78
Topformer	Topformer -base	CVPR2022	86.00	78.42	82.04	97.87	69.55	83.71
BUNet	ours		87.44	85.26	**86.34**	98.28	**75.96**	**87.12**

Fig. 10(c), (d)). The USB, on the other hand, extracts a large amount of semantic context (high-level information), focusing more on the overall shape of the branches and omitting much of the redundant information; it also enhances the distinction between branches and background in the figure (2, 4, 6 in Fig. 10(e), (f)). The output feature maps of these two branches fused by SAFM further focus on the focal region, i.e., the branches (7, 10 in Fig. 10(g), (h)). Besides, clearer branches with precise boundaries are also extracted (8, 9 in Fig. 10(g), (h)). It can be concluded that SAFM effectively combines different levels of features and improves segmentation performance.

4.3 Comparison to state-of-the-art methods

We compare the performance of nine other state-of-the-art models on our collected dataset in terms of Branches-IoU, Branches F1-score, FPS, Total params, FLOPs under equivalent experimental conditions (including BiSENetV1, BiSENetV2 [27], STDC [26], PP-LiteSeg-STDC1 [22], ESPNetV2 [21], LR-ASPP [32], OCRNet [33], FCN-HRNet [23], Topformer [24]). The results are shown in Fig. 1 and Tables 5 and 6.

4.3.1 Accuracy and segmentation results comparison

As Table 5 indicates, BiSENetV2 achieves the highest Branches Precision at 87.97%, which means that it predicts most of the samples that are branches correctly. However, our proposed model, BUNet, is only slightly lower at 87.44%. On the other hand, Branches Recall measures the model's recognition rate of real branches in the images. ESPNetV2 performs the best in this metric, with BUNet following closely at 86.93%. Moreover, F1-score combines both Precision and Recall to give a more comprehensive assessment of the model's performance. Interestingly, BUNet obtains

Table 6 The comparison of params, FLOPs and inference time between our BUNet and other state-of-the-art models on our apple branch dataset (image size of 1280×720)

Model	Backbone	Publish	Parmas(M)	FLOPs(G)	Inference time (s)
BiSENet-V1	ResNet18	ECCV2018	12.93	113.29	19.71
BiSENet-V2		IJCV2021	2.33	16.12	16.9
STDC	STDC1	CVPR2021	8.28	24.8	10.61
PPLite-Seg	STDC1	arXiv2022	8.04	19.37	16.79
ESPNet-V2		CVPR2019	1.26	5.55	**7.16**
LR-ASPP	MobileNetV3_large	ICCV2019	2.95	12.5	14.78
OCRNet	HRNet-W18-small	arXiv2019	3.84	72.7	30.37
FCN-HRNet	HRNet-W18-small	arXiv2019	1.54	8.31	14.82
Topformer	Topformer-base	CVPR2022	5.06	**3.28**	11.35
BUNet	ours		**0.93**	11.94	9.06

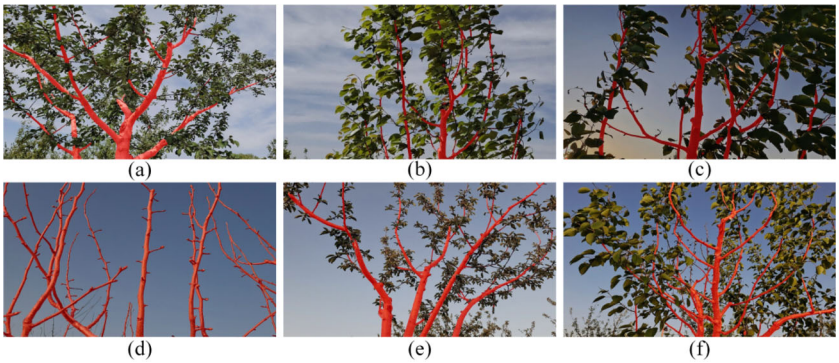

Fig. 11 The segmentation results of apple tree branches of BUNet. (a) The segmentation result under forward lighting. (b) The segmentation result under side lighting. (c) The segmentation result under backward lighting. (d), (e) and (f) demonstrate the results of segmentation with varying degrees of leaf obscuration, respectively

the highest F1-score at 86.34%, which means that it has the best overall performance among all models. Finally, Mean-IoU calculates the average intersection over union between the predicted and ground truth masks for both branches and background. Almost all models achieve a Background-IoU of 98% or more, so we mainly compare Branches-IoU. Remarkably, BUNet ranks first with 75.96% Branches-IoU and 87.12% Mean-IoU, achieving the best segmentation results among all models. It is worth mentioning that although BUNet is inspired by BiSENetV2, its Branches-IoU is 3.36% higher than BiSENetV2.

As shown in Fig. 11, our BUNet segments apple tree branches under different light and occlusion conditions effectively, with almost all visible branches well separated. This demonstrates that the proposed method can detect apple tree branches in natural images with high accuracy and speed by combining Table 6.

Figures 12 and 13 show the branch segmentation results of different models when the background is soil that has a similar colour to the branches. It is clear that BUNet has the least false segmentation, segmenting almost all visible branches completely and accurately. The other nine networks are unable to precisely distinguish between branches and soil with similar colour characteristics. For example, in Fig. 12(j), Fig. 13(c), Fig. 13(e), Fig. 13(h) and Fig. 13(i), there are many areas of mismatches. On the contrary, the segmentation results of BUNet are more accurate and complete (as shown in Fig. 12(l) and Fig. 13(l)), demonstrating the superiority and robustness of BUNet. Overall, for branches with less shading, almost all networks can be segmented clearly, but the segmentation contour of BUNet is smoother and more complete. In terms of detail, BUNet segments the heavily occluded, small branches completely and accurately (blue boxes in the Fig. 14), which performs best in segmentation effect.

4.3.2 Volume comparisons

For mobile devices with limited hardware resources, the lower the number of parameters and computation of the model, the easier it is to deploy to mobile devices. We have made huge progress on the parameters of the model. As shown in Table 5, the total params of BUNet is only 0.93M, which is smallest among all the models, 35.5% less than ESP-NetV2(ranks second). In addition, the FLOPs of BUNet is 11.94G, which is acceptable for mobile devices. As for the inference time, the BUNet ranks second at 9.06ms, which means that 110.38 images (1280x720) can be segmented in one second, meeting the real-time requirements of the picking operation.

Overall, comparisons between these models show that our model achieves a perfect balance among accuracy, efficiency

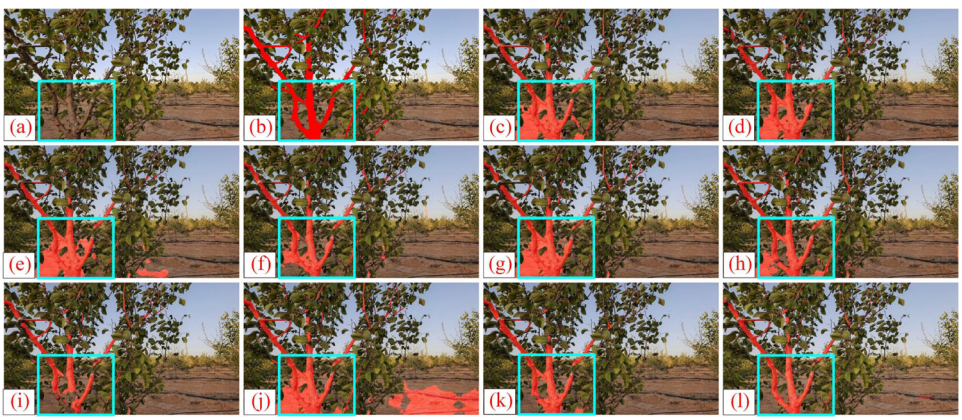

Fig. 12 The effect of segmentation of different networks in the first scenario. (a) Original image. (b) Annotated images with Photoshop (Ground Truth). (c) BiSENetV1. (d) BiSENetV2. (e) STDC. (f) PPLiteSeg-STDC1. (g) ESPNetV2. (h) LR-ASPP. (i) OCRNet. (j) FCN-HRNet. (k) Topformer. (l) BUNet(ours)

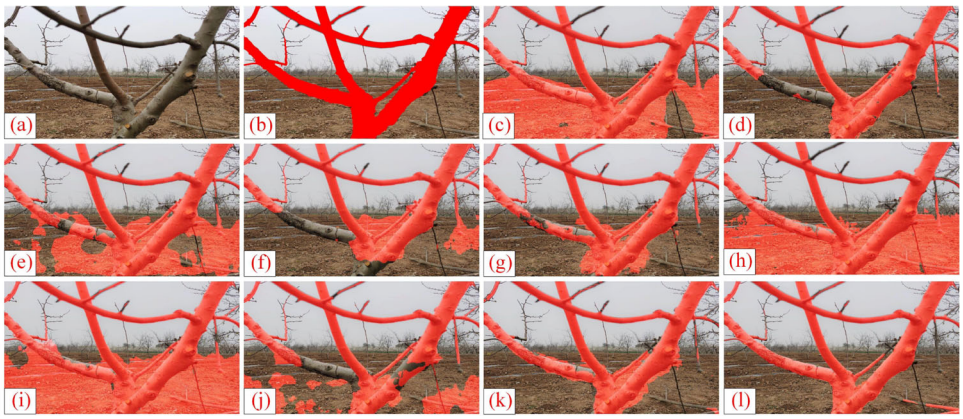

Fig. 13 The effect of segmentation of different networks in the second scenario. (a) Original image. (b) Annotated images with Photoshop (Ground Truth). (c) BiSENetV1. (d) BiSENetV2. (e) STDC. (f) PPLiteSeg-STDC1. (g) ESPNetV2. (h) LR-ASPP. (i) OCRNet. (j) FCN-HRNet. (k) Topformer. (l) BUNet(ours)

and portability, so it is more suitable to be applied for realising resource-constrained harvesting robots for apple tree branches segmentation tasks.

4.4 Discussion

Accurate and rapid detection of apple tree branches is of great importance for the practical application and development of automatic apple harvesting robots. The harvesting robots are designed to address labour shortages, reduce the risk of human injury, and improve productivity and profitability of the fruit industry. Before the automatic harvesting robot performs its harvesting operation, except the given target point, i.e. the position of the apple, information on the location of obstacles is also necessary to carry out the dynamic planning and obstacle avoidance of the robotic arm before it can finally approach the target point for harvesting action. Therefore, the detection of apple tree branches can increase the success rate of automatic harvesting, while reducing the probability of damaging the robotic arm or the apple tree. Most of the current advanced detection algorithms are based on deep convolutional neural networks, which can be broadly classified into object detection and semantic segmentation. For specific application scenarios, we need accurate, pixel-by-pixel branch detection for obstacle avoidance, so a semantic segmentation algorithm that provides pixel-by-pixel prediction is suitable for branch segmentation. But most of today's mobile harvesting robots use mobile hardware devices with far less memory and calculation power than large servers in the laboratory. To improve the portability of the network on mobile devices with limited hardware resources, we design the semantic segmentation network BUNet with only 0.93M number of parameters.

However, since apple trees grow in an open and unstructured environment, factors such as complex natural environment, uncertain illumination, and the occlusion of massive leaves, make the segmentation of apple tree branches very difficult. Therefore, we collect a comprehensive and complex dataset of apple branches, taking into account different light and occlusion conditions to improve the robustness of the model. We also use different shooting angles to simulate different harvesting angles. However, the dataset is not obtained during the apple harvest season in this study because of the long collection period and the short harvest period of ripe

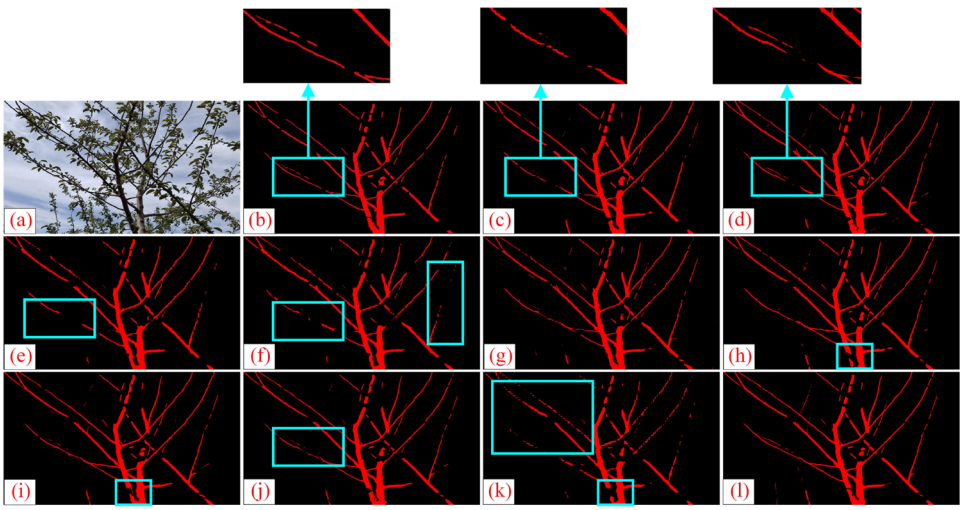

Fig. 14 Segmentation results of image with obscuration by leaves. (a) Original image. (b) Ground Truth. (c) BiSENetV1. (d) BiSENetV2. (e) STDC. (f) PPLiteSeg-STDC1. (g) ESPNetV2. (h) LR-ASPP. (i) OCRNet. (j) FCN-HRNet. (k) Topformer. (l) BUNet(ours)

apples, which may lead to insufficient segmentation accuracy during the actual harvesting process. In future work, we will expand the dataset and increase the data during the harvest season. Our proposed algorithm can quickly and accurately segment apple tree branches, and in addition to its application in the apple harvesting process, we consider that it can also be applied to automatic pruning robots, which is to enhance the ventilation and light penetration of fruit trees and increase the yield and quality of apples.

5 Conclusion

We propose a lightweight, highly accurate and real-time semantic segmentation network, Bilateral U-shape Network (BUNet), for apple tree branches segmentation, which adopts two independent encoder-decoder structures to obtain spatial details and semantic information respectively. In addition, we design the Simplified Attention Fusion Module as a fusion module for the two branches to improve segmentation accuracy. Compared to the other eight models, BUNet achieves the highest Branches-IoU of 75.96% with a total parameter of 0.93M at 110.32 FPS. The experimental results demonstrate the effectiveness and efficiency of our method in real orchard scene, which provide accurate obstacle avoidance information for mechanical arm, reducing the risk of damage caused by colliding with the branches. In addition, the model size and computational complexity of BUNet is tiny enough to be easily deployed to the mobile harvesting robot. Future work will focus on the handling of leaves and fruits occlusion of branches using Generative adversarial network and 3D reconstruction of actual apple trees.

Funding No funding was received to assist with the preparation of this manuscript

Availability of data and materials The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request

Declarations

Competing interests The authors have no relevant financial or non-financial interests to disclose

References

1. Bac CW, Van Henten EJ, Hemming J, Edan Y (2014) Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. J Field Robotics 31(6):888–911. https://doi.org/10.1002/rob.21525
2. Kapach K, Barnea E, Mairon R, Edan Y, Ben-Shahar O (2012) Computer vision for fruit harvesting robots-state of the art and challenges ahead. Int J Comput Vision Robotics 3(1/2):4–34. https://doi.org/10.1504/IJCVR.2012.046419
3. Zhang Z, Igathinathane C, Li J, Cen H, Lu Y, Flores P (2020) Technology progress in mechanical harvest of fresh market apples. Comput Electr Agriculture 175:105606. https://doi.org/10.1016/j.compag.2020.105606
4. Tang Y, Chen M, Wang C, Luo L, Li J, Lian G, Zou X (2020) Recognition and localization methods for vision-based fruit picking robots: A review. Frontiers Plant Sci 11:510. https://doi.org/10.3389/fpls.2020.00510
5. Fu L, Gao F, Wu J, Li R, Karkee M, Zhang Q (2020) Application of consumer rgb-d cameras for fruit detection and localization in field: A critical review. Comput Electr Agriculture 177:105687. https://doi.org/10.1016/j.compag.2020.105687
6. Chen L-C, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV), pp 801–818. https://doi.org/10.1007/978-3-030-01234-2_49
7. Kamilaris A, X F (2018) Deep learning in agriculture: A survey. Comput Electr Agriculture 147:70–90. https://doi.org/10.1016/j.compag.2018.02.016
8. Mo Y, Wu Y, Yang X, Liu F, Liao Y (2022) Review the state-of-the-art technologies of semantic segmentation based on deep learning. Neurocomputing 493:626–646. https://doi.org/10.1016/j.neucom.2022.01.005
9. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440. https://doi.org/10.1109/TPAMI.2016.2572683
10. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Analysis Machine Int 40(4):834–848. https://doi.org/10.1109/TPAMI.2017.2699184
11. Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. https://doi.org/10.48550/arXiv.1706.05587 arXiv preprint arXiv:1706.05587
12. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In Proceedings of the IEEE International conference on computer vision, pp 2961–2969. https://doi.org/10.1109/ICCV.2017.322
13. Guo Y, Liu Y, Georgiou T, Lew MS (2018) A review of semantic segmentation using deep neural networks. Int J Multimedia Inf Retrieval 7(2):87–93. https://doi.org/10.1007/s13735-017-0141-z
14. Lin G, Tang Y, Zou X, Xiong J, Li J (2019) Guava detection and pose estimation using a low-cost rgb-d sensor in the field. Sensors 19(2):428. https://doi.org/10.3390/s19020428
15. Lin G, Tang Y, Zou X, Wang C (2021) Three-dimensional reconstruction of guava fruits and branches using instance segmentation and geometry analysis. Comput Electr Agriculture 184:106107. https://doi.org/10.1016/j.compag.2021.106107
16. Li J, Tang Y, Zou X, Lin G, Wang H (2020) Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots. IEEE Access 8:117746–117758. https://doi.org/10.1109/ACCESS.2020.3005386
17. Kang H, Chen C (2019) Fruit detection and segmentation for apple harvesting using visual sensor in orchards. Sensors 19(20):4599. https://doi.org/10.3390/s19204599
18. Bechar A, Vigneault C (2016) Agricultural robots for field operations: Concepts and components. Biosyst Eng 149:94–111. https://doi.org/10.1016/j.biosystemseng.2016.06.014
19. Bechar A, Vigneault C (2017) Agricultural robots for field operations. part 2: Operations and systems. Biosyst Eng 153:110–128. https://doi.org/10.1016/j.biosystemseng.2016.11.004
20. Mehta S, Rastegari M, Caspi A, Shapiro L, Hajishirzi H (2018) Espnet: Efficient spatial pyramid of dilated convolutions for semantic

segmentation. In Proceedings of the European conference on computer vision (ECCV). https://doi.org/10.1007/978-3-030-01249-6_34
21. Mehta S, Rastegari M, Shapiro L, Hajishirzi H (2019) Espnetv2: A light-weight, power efficient, and general purpose convolutional neural network. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
22. Peng J, Liu Y, Tang S, Hao Y, Chu L, Chen G, Wu Z, Chen Z, Yu Z, Du Y, et al. (2022) Pp-liteseg: A superior real-time semantic segmentation model. https://doi.org/10.48550/arXiv.2204.02681 arXiv preprint arXiv:2204.02681
23. Sun K, Zhao Y, Jiang B, Cheng T, Xiao B, Liu D, Mu Y, Wang X, Liu W, Wang J (2019) High-resolution representations for labeling pixels and regions. https://doi.org/10.48550/arXiv.1904.04514 arXiv preprint arXiv:1904.04514
24. Zhang W, Huang Z, Luo G, Chen T, Wang X, Liu W, Yu G, Shen C (2022) Topformer: Token pyramid transformer for mobile semantic segmentation. In Proceedings of the IEEE/CVF ccnference on computer vision and pattern recognition (CVPR), pp 12083–12093. https://doi.org/10.1109/CVPR52688.2022.01177
25. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N (2018) Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European conference on computer vision (ECCV), pp 325–341. https://doi.org/10.1007/978-3-030-01261_20
26. Fan M, Lai S, Huang J, Wei X, Chai Z, Luo J, Wei X (2021) Rethinking bisenet for real-time semantic segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9716–9725. https://doi.org/10.1109/CVPR46437.2021.00959
27. Yu C, Gao C, Wang J, Yu G, Shen C, Sang N (2021) Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. Int J Comput Vision 129(11):3051–3068. https://doi.org/10.1007/s11263-021-01515-2
28. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Analysis Machine Intell 39(12):2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615
29. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/arXiv.1704.04861 arXiv preprint arXiv:1704.04861
30. Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), pp 3–19. https://doi.org/10.1007/978-3-030-01234-2_1
31. Contributors P (2019) PaddleSeg, End-to-end image segmentation kit based on PaddlePaddle. https://github.com/PaddlePaddle/PaddleSeg
32. Howard A, Sandler M, Chu G, Chen L-C, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V, Le QV, Adam H (2019) Searching for mobilenetv3. In Proceedings of the IEEE/CVF international conference on computer vision (ICCV). https://doi.org/10.1109/ICCV.2019.00140
33. Yuan Y, Chen X, Chen X, Wang J (2019) Segmentation transformer: Object-contextual representations for semantic segmentation. https://doi.org/10.48550/arXiv.1909.11065 arXiv preprint arXiv:1909.11065