An Improved Mask R-CNN for Instance Segmentation of Tree Crowns in Aerial Imagery

Ziyi Sun, Bing Xue, Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington
Wellington, New Zealand
{ziyi.sun, bing.xue, mengjie.zhang}@ecs.vuw.ac.nz

Jan Schindler
Manaaki Whenua-Landcare Research
Wellington, New Zealand
SchindlerJ@landcareresearch.co.nz

2023 38th International Conference on Image and Vision Computing New Zealand (IVCNZ) | 979-8-3503-7051-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/IVCNZ61134.2023.10343827

***Abstract*—Instance segmentation of individual tree crowns is an important real-world application that facilitates forest management, urban planning, biodiversity modelling, and pest control. Recently, Convolutional Neural Networks (CNNs) have achieved great success in instance segmentation. Several efforts have successfully applied CNNs to instance segmentation of tree crowns. As a representative instance segmentation method, Mask R-CNN shows good performance in the application of instance segmentation of individual tree crowns. However, Mask R-CNN with ResNet or ResNeXt is not enough to extract sufficient feature information for individual tree crowns. In this paper, an improved Mask R-CNN method is proposed by introducing an effective backbone structure. Furthermore, we propose a new mask branch to help segment tree crowns from complex backgrounds. Experimental results show that the new method can effectively identify and segment individual tree canopies for the Wellington city of Aotearoa New Zealand dataset.**

***Index Terms*—Instance Segmentation, Tree Crowns, Mask R-CNN**

I. Introduction

Instance segmentation is a pivotal problem in computer vision. It plays an important role in a broad range of application fields: autonomous driving [1], industrial defect detection [2], medical image analysis [3], and agriculture [4]. The main goal of instance segmentation is to detect and segment each object instance. Instance segmentation is a challenging task, because it not only performs pixel-level classification for all target objects, but also requires separating different objects belonging to the same class label (i.e. giving the instance ID for each object). The instance segmentation task of tree crowns in aerial imagery is an important practical application, which will help inform forest management, urban planning, biodiversity modelling, and pest control (possums, etc). This paper aims to achieve instance segmentation of tree crowns in aerial imagery, that is, to detect and segment individual tree crowns in the images. Mapping of individual tree crowns will be done at the city level, here in Wellington, Aotearoa New Zealand. An example of a tree crown image and its ground truth is shown in Fig. 1.

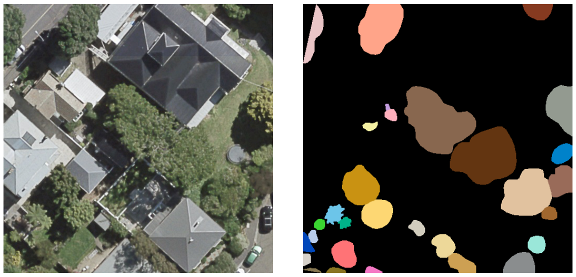

Fig. 1. An example of a tree crown image and its ground truth for Wellington City in Aotearoa New Zealand. Different colors indicate different canopy instances.

With the rise of deep neural networks (DNNs), convolutional neural network (CNN) models are springing up like mushrooms, which yield state-of-the-art results in instance segmentation. For example, the two-stage top-down approach [5]–[11] has achieved good performance on instance segmentation tasks, which first performs object detection by predicting bounding boxes, and then segments the instance within each bounding box [12]. The pioneering network was Mask R-CNN [5], and then, more methods [6], [7], [9], [10], [13] were proposed to further improve performance. Now there are a few methods for instance segmentation of individual tree crowns, many of which are based on Mask R-CNN [14]–[16]. Therefore, in this work, we address instance segmentation of tree crowns in aerial imagery based on Mask R-CNN.

However, achieving instance segmentation of individual tree crowns on aerial images is not an easy task due to the following reasons. First, as far as the instance segmentation task is concerned, it requires the detection of target objects, but also delineating the boundaries of individual objects. Second, the boundaries of tree crowns are often not clear, since the trees themselves have a complex canopy structure, such as various crown sizes, overlapping crowns, and mutual shading; the color, shape, and texture characteristics of tree crowns are very similar to the surrounding background, such as weeds and grass, which increases the difficulty of the problem for detection and segmentation. Finally, there are many small objects in the images, and some small-size tree crowns may only take up a few pixels in the image.

979-8-3503-7051-5/23/$31.00 ©2023 IEEE

A. Goals

The goal of this paper is to propose an effective method to achieve instance segmentation of individual tree crowns in aerial images. This method is expected to extract informative features of tree crowns from the training data, so as to correctly detect and segment individual canopies. To be specific, the goal can be summarized as follows:

- Investigate an effective backbone structure in the Mask R-CNN architecture to extract rich canopy features in the images.
- Design an effective and efficient mask branch to help segment each tree crown from the complex background.
- Select an appropriate hyperparameter setting for the proposed method.
- Compare with state-of-the-art methods and deeply analyze the experimental results.

II. Background

A. Mask R-CNN for Instance Segmentation

He et al. [5] proposed a general method for instance segmentation named Mask R-CNN. Mask R-CNN extended Faster R-CNN [17] by adding a mask prediction branch, in which Faster R-CNN calculated the bounding box coordinates and the corresponding class label to achieve object detection. The Mask R-CNN framework yielded promising results in instance segmentation tasks. Based on Mask R-CNN, a multi-stage framework was proposed for object detection and instance segmentation, named Cascade R-CNN [6]. This method trained stages sequentially by using the output of one to train the next, which can progressively refine the predictions. Chen et al. [7] further improved Mask R-CNN and proposed Hybrid Task Cascade (HTC), which effectively integrates cascade into instance segmentation by interweaving detection and segmentation tasks together for joint multi-stage processing. Qiao et al. [9] explored the idea of looking and thinking twice in the backbone design of the Mask R-CNN framework and proposed an approach called DetectoRS. DetectoRS implemented this idea by designing Recursive Feature Pyramid at the macro level and Switchable Atrous Convolution at the micro level. These methods use ResNet or ResNeXt as the backbone architecture, however, ResNet has been far surpassed by some state-of-the-art backbone structures on many computer vision tasks [18]–[20].

B. Instance Segmentation of Individual Tree Crowns

For instance segmentation of tree crowns, many practical tasks were addressed by applying well-performed architectures. Based on Mask R-CNN, [21] realized instance segmentation of standing dead trees in dense forest from aerial imagery. Since the training dataset (195 images) is limited, transfer learning and the image augmentation technique were employed to leverage the limitation of training datasets. Also, [22] applied Mask R-CNN to perform tree crown detection and delineation by using satellite images from tropical forests. Li et al. [15] proposed ACE R-CNN based on Mask R-CNN by adding an attention complementary module in the backbone and an edge loss. Using unmanned aerial vehicle (UAV) RGB images and LiDAR data, ACE R-CNN achieved individual tree species identification. Based on Google Earth images, [23] applied Mask R-CNN to detect tree crown cover in New York's Central Park. Sun et al. [24] counted trees in a subtropical mega city by achieving instance segmentation of tree crowns in aerial images. This work was based on Cascade R-CNN and improved Cascade R-CNN by adding three types of attention modules. MO et al. [25] adopted YOLACT to perform instance segmentation of litchi tree canopies from images acquired by UAVs. For high-resolution digital orthophoto maps, a partition-based method was designed in this work. The final prediction result was obtained by integrating the inference results of image patches into a unified result. However, the visual results showed that the segmentation effect of this method was worth improving. The above methods can effectively achieve the instance segmentation task of tree crowns, but they have not explored more effective backbone structures. The existing methods motivated us to design an effective architecture to solve the instance segmentation task of individual tree crowns for Wellington City in Aotearoa New Zealand.

III. Method

As a typical work of the two-stage top-down approach, Mask R-CNN shows great potential in instance segmentation and also demonstrates its effectiveness in the application of tree crowns [15], [21], [23]. In this work, Mask R-CNN is used to achieve instance segmentation of individual tree crowns, which is an end-to-end training model. The architecture of the proposed method is shown in Fig. 2. Specifically, the network architecture mainly consists of three components, backbone, Region Proposal Network (RPN), and network head.

Backbone: The backbone is a multi-layer neural network used to extract feature maps from input images. ResNet [26] is one of the commonly used backbone network structures. Feature Pyramid Network (FPN) [27] is often employed in the backbone to extract multi-scale Region of Interest (RoI) features from different levels of the feature pyramid. The backbone with FPN can effectively improve recognition accuracy without increasing the computational time, which is also adopted in this work. By adding the shortcut connection, ResNet alleviates the problem of gradient vanishing. However, ResNet architectures might suffer from gradient exploding as the depth becomes large [28], and lose expressivity as the depth goes to infinity [29], [30]. In addition, ResNet's performance has its limitations even as the network deepens further.

Many effective backbone structures have been proposed to enhance feature extraction. In this paper, we use ConvNeXt [20] as the backbone network, which shows superiority even compared to the dominant backbone, i.e. Transformers [18], [31]. A ConvNeXt block is shown in Fig. 3. ConvNeXt is an enhanced version based on ResNet, which is improved from five aspects: macro design, ResNeXt-ify, inverted bottleneck, large kernel size, and various layer-wise micro designs [20].

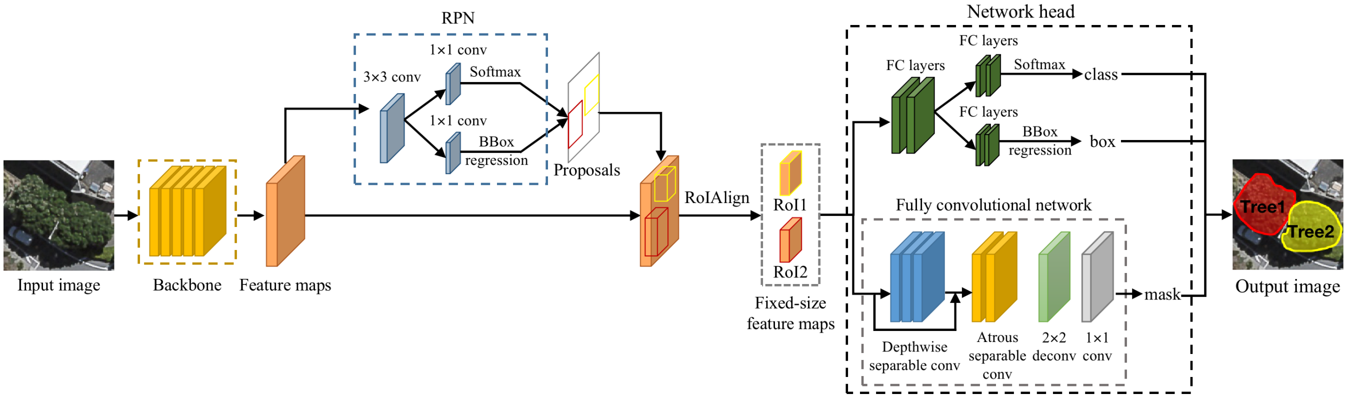

Fig. 2. The network architecture of the proposed method. BBox represents the bounding box, and FC layers are fully-connected layers.

In the macro design, the number of blocks in each stage is changed from (3, 4, 6, 3) in ResNet-50 to (3, 3, 9, 3), which is inspired by the design of Swin Transformers [18]. The ResNet-stem cell (containing a 7×7 convolution layer with stride 2 and a max pooling) is adjusted to a patchify layer implemented using a 4×4, stride 4 convolutional layer. The grouped convolution of ResNeXt [32] is adopted in ConvNeXt, which is implemented by using depthwise convolution. The inverted bottleneck is created by setting the hidden dimension of the block to four times wider (384) than the input dimension (96). ConvNeXt employs a large kernel size (7×7), before which the position of the depthwise convolution layer is moved up. In the micro design, ConvNeXt replaces ReLU with GELU, BatchNorm with Layer Normalization (LN), and uses fewer activation functions and normalization layers.

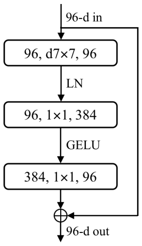

Fig. 3. A ConvNeXt block. A layer is shown as input channels, filter size, and output channels. 96-d indicates that the channel dimension of the feature is 96. In d7×7, d represents depthwise convolution. LN is Layer Normalization.

Region Proposal Network: RPN is a full convolutional network (FCN) structure, which plays the role of object detection by generating candidate bounding boxes (proposals). After feature extraction, RPN can predict the bounding boxes of individual objects, which are generated from the output features of the backbone network. The area surrounded by the bounding box is referred to as the Region of Interest (RoI). Then, for each RoI of different sizes, the RoI Align layer is used to resample the feature map and output the feature map of fixed size. Generally, RPN can only generate a rough detection box, which needs to be processed accurately by the subsequent network head.

Network head: The network head is mainly used for the classification and segmentation of target instances, including a classification branch and a mask branch. The box branch is used for bounding box classification and regression, where classification refers to predicting the category of objects in a bounding box, and regression is to obtain a more refined position of a bounding box. The mask branch is used to predict a segmentation mask at the pixel level, including all pixels belonging to the object within the bounding box. Mask R-CNN uses several common standard convolution layers to build the mask branch, which is simple but not lightweight. In this work, we propose a new and effective mask branch by adopting depth-wise separable convolution [33] and atrous separable convolution [34]. Depth-wise separable convolution greatly reduces computational complexity by decomposing the standard convolution into a depth-wise convolution and a point-wise convolution (i.e. 1×1 convolution). The atrous separable convolution is proven to significantly reduce the computational complexity while maintaining the performance [34]. The new branch consists of three depth-wise separable convolution layers, with a skip connection realized by 1×1 convolution, and two atrous separable convolution layers, followed by a 2×2 deconvolution layer and a 1×1 convolution layer. The structure consisting of continuous depth-separable convolutions with a skip connection followed by atrous separable convolutions is inspired by the Exit Flow of the improved Xception [34]. This structure can also achieve good results in instance segmentation tasks, which is proved by experiments.

IV. Design of the Experiments

A. Datasets

Manaaki Whenua - Landcare Research provided the manually labeled dataset of individual tree crowns in aerial images for Wellington City in Aotearoa New Zealand. The labels were hand-drawn and comprised about 12,000 canopy objects in 670 RGB images (512 × 512 pixel dimension). The dataset was divided by experts into a training set and a test set containing approximately 11,000 objects in 603 images and 1,100 objects in 67 images, respectively. The images in the dataset are urban scenes with trees, roads, cars, houses, etc. These aerial images are taken from the top down, unlike the front view in benchmark datasets, such as ImageNet [35] and MS COCO [36]. During training, we only use a simple data augmentation method, which is random flips with a probability of 0.5.

B. Parameter Settings

Following MS COCO [36], Average Precision (AP) is used to evaluate the performance of these Mask R-CNN models. We also use the averaged AP value [36] over 10 multiple IoU thresholds of 0.50:0.05:0.95 to measure the overall performance. The AP mentioned below refers to the average of the 10 thresholds. AP_{50} and AP_{75} are also adopted, which are computed at a single IoU of .50 and .75. Since the tree image dataset contains canopies of different sizes, we follow MS COCO and use AP_{S}, AP_{M}, and AP_{L} to evaluate Small, Medium, and Large objects.

Our experiments are based on the open-source detection toolbox MMDetection [37]. The experiments are carried out on one Quadro RTX 6000 GPU card and are conducted with Torch. The parameter settings are shown in Table I. The batch size is set to 4 and the number of training epochs is set to 120, which is enough to make sure the weights converge and achieve good performance. We follow ConvNeXt [20] that adopts AdamW optimizer with a cosine decay learning rate scheduler, the weight decay of 0.05, the momentum of (0.9, 0.999), and the linear warmup schedule. All the backbone models used in experiments adopt the publicly pre-trained weights on ImageNet-1K.

TABLE I
THE PARAMETER SETTING USED IN TRAINING.

Parameter	Value
Batch size	4
Epochs	120
Optimizer	AdamW
Base learning rate	0.001
Weight decay	0.05
Optimizer momentum	(0.9, 0.999)
Learning rate schedule	Cosine decay
Warmup schedule	Linear
Min learning rate	0.00001

C. Comparison Methods

To show the effectiveness of the proposed method, 6 effective methods are used for comparisons on the tree image dataset. These methods include two-stage methods and one-stage methods. Four well-known two-stage top-down methods are adopted, which are Mask R-CNN [5], Cascade R-CNN [6], HTC [7], and DetectoRS [9]. To demonstrate efficiency, the proposed method is also compared to one-stage methods, which are generally faster. The one-stage approach includes YOLACT [38] and SOLOv2 [39]. Following Mask R-CNN [5], Cascade R-CNN [6], HTC [7], and DetectoRS [9], two backbone structures, ResNet-101 [1] and ResNeXt-101 (64×4d variant [2]), are both included in comparative experiments. The number of epochs for these models is set to 120, which is the same as our method, and the rest of the parameter settings refer to their original papers.

V. RESULTS

A. Overall Results

In this section, we show comparison results with other methods on the tree image dataset in Table II, which also includes the generated model size (memory) of these methods and the training time. The proposed method with ConvNeXt-Base achieves a mask AP of 38.1%, which is much better than other instance segmentation methods. Furthermore, our method shows superiority over objects of all sizes. Compared with the other two-stage top-down methods, our method is superior in terms of all AP metrics and competitive in model size and training time. Although the models generated by one-stage methods have smaller memory sizes and relatively shorter training times, our method significantly outperforms them by more than 16% on AP.

TABLE II
COMPARISON RESULTS WITH OTHER METHODS. MEMORY REPRESENTS THE GENERATED MODEL SIZE. TIME REFERS TO THE TRAINING TIME REQUIRED TO GET A MODEL.

	Backbone	AP	AP_{50}	AP_{75}	AP_{S}	AP_{M}	AP_{L}	Memory	Time
Mask	ResNet-101	25.8	51.4	23.3	18.0	42.3	57.1	480M	1.8h
R-CNN	ResNeXt-101	29.5	56.1	28.6	22.6	44.6	61.3	776M	2.1h
Cascade	ResNet-101	30.7	57.6	29.1	23.5	46.6	60.2	732M	3.1h
R-CNN	ResNeXt-101	30.7	57.3	28.8	23.5	46.6	62.0	1.00G	4.6h
HTC	ResNet-101	31.1	57.8	30.7	23.4	47.9	60.9	733M	4.1h
	ResNeXt-101	31.6	59.0	30.9	24.2	48.2	60.7	1.01G	5.6h
YOLACT	ResNet-101	21.6	48.8	15.6	14.9	35.8	55.7	411M	1.5h
SOLOv2	ResNet-101	20.6	46.5	16.6	14.9	34.8	59.1	497M	2.0h
DetectoRS	ResNet-101	30.5	56.7	29.7	23.3	46.1	67.2	1.47G	7.0h
	ResNeXt-101	31.5	59.1	30.7	24.8	46.1	60.5	2.23G	10.1h
Ours	ConvNeXt-B	**38.1**	**61.7**	**40.2**	**32.7**	**51.5**	**70.7**	1.18G	2.7h

B. Comparisons on Different Backbones

To demonstrate the effectiveness of the adopted backbone, we compare different backbones, i.e. ResNet-101, ResNeXt-101, and ConvNeXt-Base, under the Mask R-CNN framework. To be fair, all these models have the same parameter settings. Fig. 4 describes some qualitative results, which show that these models achieve good performance even if the tree images are viewed from the top down, which is different from the front view of the data (ImageNet) used in the pre-trained weights. Fig. 4 highlights some areas with yellow and blue boxes. The yellow box in the first image contains a single canopy. ResNet and ResNeXt detect two canopies in the region, while ConvNeXt correctly identifies the canopy. In the second image, the regions containing one and two tree crowns are highlighted

[1] 101 means that the number of convolutional layers in the network is 101.

[2] 64×4d indicates cardinality = 64 and bottleneck width = 4d.

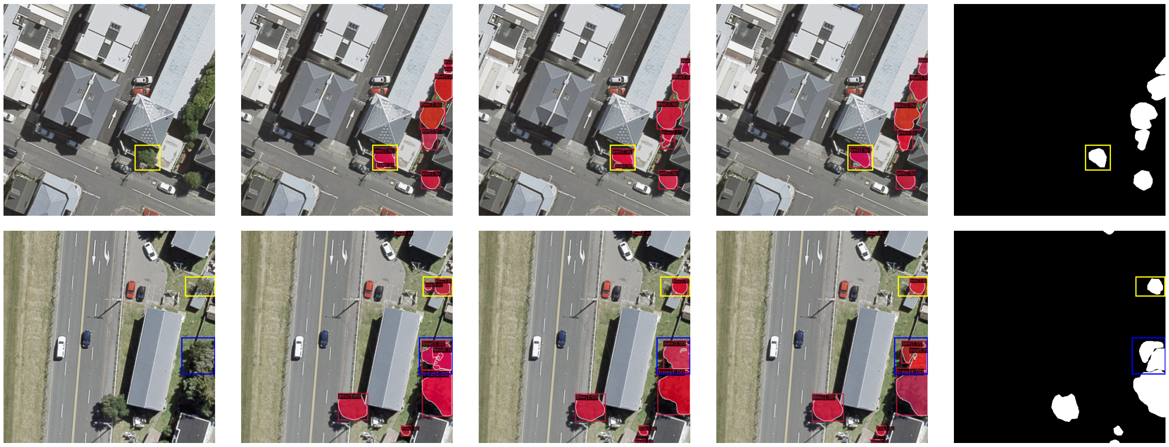
(a) Input (b) ResNet (c) ResNeXt (d) ConvNeXt (e) Ground truth

Fig. 4. Qualitative results of Mask R-CNN models with different backbones.

with yellow and blue boxes, respectively. However, ResNet detects two canopies in the area surrounded by the yellow box. For the area enclosed by the blue box, ResNeXt only recognizes one canopy. In comparison, ConvNeXt correctly identifies canopies and achieves clearer boundary delineation.

TABLE III
QUANTITATIVE RESULTS OF MASK R-CNN MODELS WITH DIFFERENT BACKBONES.

Backbone	AP	AP_{50}	AP_{75}	AP_{S}	AP_{M}	AP_{L}	Memory
ResNet-101	32.6	56.7	33.2	27.8	45.4	58.1	720M
ResNeXt-101	34.2	58.1	36.4	29.6	45.5	59.6	1.14G
ConvNeXt-B	**37.6**	**60.3**	**40.0**	**32.7**	**50.4**	**69.2**	1.20G

The quantitative results of the Mask R-CNN models with different backbone structures are indicated in Table III. The results show that ConvNeXt-Base is more effective than ResNet-101 and ResNeXt-101, it significantly outperforms the other two models on all indicators.

TABLE IV
QUANTITATIVE RESULTS OF OUR METHOD AND MASK R-CNN.

	Backbone	AP	AP_{50}	AP_{75}	AP_{S}	AP_{M}	AP_{L}	Memory
Mask R-CNN	ConvNeXt-B	37.6	60.3	40.0	**32.7**	50.4	69.2	1.20G
Ours	ConvNeXt-B	**38.1**	**61.7**	**40.2**	**32.7**	**51.5**	**70.7**	1.18G

C. Effectiveness of the Proposed Mask Branch

In order to demonstrate the effect of the proposed mask branch, we compare the proposed method with Mask R-CNN with the same backbone structure, where the parameter settings of Mask R-CNN are the same as our method for fairness. Qualitative results are shown in Fig. 5. In the first image, Mask R-CNN fails to detect another tree crown located in the yellow box. Also, in the second image, the small-sized canopies in the yellow boxes are not detected by Mask R-CNN. In contrast, our method successfully identifies most of the tree canopies in the image. Table IV reports quantitative results, which show that our method outperforms Mask R-CNN, and the values of our method are higher than or equal to those of Mask R-CNN on all metrics. These results prove the effectiveness of the proposed mask branch, which also uses less memory.

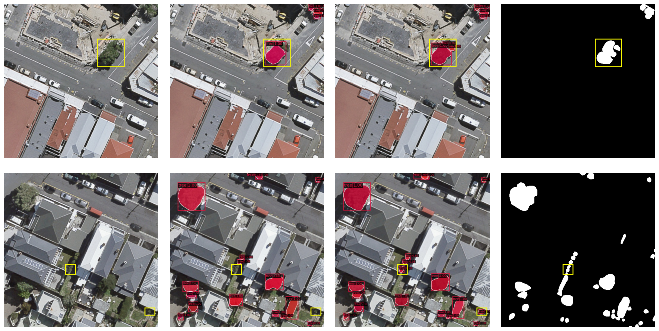
(a) (b) (c) (d)

Fig. 5. Qualitative results of our method and Mask R-CNN. (a) Input, (b) Mask R-CNN, (c) Ours, (d) Ground truth.

VI. CONCLUSIONS

In this paper, we aimed to address the instance segmentation task of individual tree crowns in aerial images. This goal has been successfully achieved by proposing an improved Mask R-CNN method. An effective backbone structure and a new mask branch were developed in this method. With these designs, the proposed method obtained significantly better performance on the tree image dataset than other methods, and it can accurately identify and delineate different sizes of tree crowns in the images. The results demonstrated that the ConvNeXt backbone can extract rich image features to facilitate the detection and segmentation of canopies. Meanwhile, the results showed that the proposed mask branch achieved effective and efficient pixel-level mask prediction.

For future work, to reduce the computational cost, it is worth investigating how to design a lightweight architecture, especially the backbone part, which has a great impact on the performance of the model.

Acknowledgment

This work was supported by the New Zealand Ministry of Business, Innovation and Employment under contract C09X1923 (Catalyst: Strategic Fund).

References

[1] Z. Zhang, S. Fidler, and R. Urtasun, "Instance-level segmentation for autonomous driving with deep densely connected mrfs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 669–677.

[2] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Applied Sciences*, vol. 8, no. 9, p. 1575, 2018.

[3] Y. Xu, Y. Li, Y. Wang, M. Liu, Y. Fan, M. Lai, I. Eric, and C. Chang, "Gland instance segmentation using deep multichannel neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 12, pp. 2901–2912, 2017.

[4] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," *Pattern recognition letters*, vol. 81, pp. 80–89, 2016.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[6] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[7] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang *et al.*, "Hybrid task cascade for instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4974–4983.

[8] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6409–6418.

[9] S. Qiao, L.-C. Chen, and A. Yuille, "Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 10 213–10 224.

[10] H. Chen, K. Sun, Z. Tian, C. Shen, Y. Huang, and Y. Yan, "Blendmask: Top-down meets bottom-up for instance segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8573–8581.

[11] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "Global context networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[12] W. Gu, S. Bai, and L. Kong, "A review on 2d instance segmentation based on deep neural networks," *Image and Vision Computing*, p. 104401, 2022.

[13] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[14] Y. Sun, Z. Li, H. He, L. Guo, X. Zhang, and Q. Xin, "Counting trees in a subtropical mega city using the instance segmentation method," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, p. 102662, 2022.

[15] Y. Li, G. Chai, Y. Wang, L. Lei, and X. Zhang, "Ace r-cnn: An attention complementary and edge detection-based instance segmentation algorithm for individual tree species identification using uav rgb images and lidar data," *Remote Sensing*, vol. 14, no. 13, p. 3035, 2022.

[16] A. Safonova, E. Guirado, Y. Maglinets, D. Alcaraz-Segura, and S. Tabik, "Olive tree biovolume from uav multi-resolution image segmentation with mask r-cnn," *Sensors*, vol. 21, no. 5, p. 1617, 2021.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 201, 2015.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[19] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 009–12 019.

[20] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.

[21] A. Sani-Mohammed, W. Yao, and M. Heurich, "Instance segmentation of standing dead trees in dense forest from aerial imagery using deep learning," *ISPRS Open Journal of Photogrammetry and Remote Sensing*, vol. 6, p. 100024, 2022.

[22] J. R. G. Braga, V. Peripato, R. Dalagnol, M. P. Ferreira, Y. Tarabalka, L. E. OC Aragão, H. F. de Campos Velho, E. H. Shiguemori, and F. H. Wagner, "Tree crown delineation algorithm based on a convolutional neural network," *Remote Sensing*, vol. 12, no. 8, p. 1288, 2020.

[23] M. Yang, Y. Mou, S. Liu, Y. Meng, Z. Liu, P. Li, W. Xiang, X. Zhou, and C. Peng, "Detecting and mapping tree crowns based on convolutional neural network and google earth images," *International Journal of Applied Earth Observation and Geoinformation*, vol. 108, p. 102764, 2022.

[24] Y. Sun, Z. Li, H. He, L. Guo, X. Zhang, and Q. Xin, "Counting trees in a subtropical mega city using the instance segmentation method," *International Journal of Applied Earth Observation and Geoinformation*, vol. 106, p. 102662, 2022.

[25] J. Mo, Y. Lan, D. Yang, F. Wen, H. Qiu, X. Chen, and X. Deng, "Deep learning-based instance segmentation method of litchi canopy from uav-acquired images," *Remote Sensing*, vol. 13, no. 19, p. 3919, 2021.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[28] S. Hayou, E. Clerico, B. He, G. Deligiannidis, A. Doucet, and J. Rousseau, "Stable resnet," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1324–1332.

[29] G. Yang and S. Schoenholz, "Mean field residual networks: On the edge of chaos," *Advances in neural information processing systems*, vol. 30, 2017.

[30] S. Hayou, A. Doucet, and J. Rousseau, "On the impact of the activation function on deep neural networks training," in *International conference on machine learning*. PMLR, 2019, pp. 2672–2680.

[31] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[32] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[33] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[34] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[37] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[38] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "Yolact: Real-time instance segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9157–9166.

[39] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "Solov2: Dynamic and fast instance segmentation," *Advances in Neural information processing systems*, vol. 33, pp. 17 721–17 732, 2020.