CEDAnet: Individual Tree Segmentation in Dense Orchard via Context Enhancement and Density Prior

Fangjie Zhu, Zhenhao Chen, Haoyang Li, Qian Shi, *Senior Member, IEEE*, and Xiaoping Liu, *Member, IEEE*

***Abstract*—Individual tree segmentation (ITS) is a pivotal technique in orchard research, estimating tree counts and delineating crown contours. This method provides foundational data for assessing orchard health, nutritional composition, and predicting yield. Unmanned aerial vehicles (UAVs) have become an essential data source for (ITS) due to their capability to capture ultra-fine details. However, current deep-learning-based ITS methods struggle to accurately handle densely overlapping fruit tree distributions with similar characteristics in UAVs images, primarily due to the intricate nature of spatial arrangements in such scenarios. In this article, we propose CEDAnet, a context enhancement, and density adjustment network, to address the challenge of dense fruit trees segmentation. Specifically, a transformer-based contextual aggregation module is designed to distinguish different instances and refine the boundary of the instances. We have proposed a density-guided nonmaximum suppression method to adaptively generate sufficient candidate bounding boxes, aiming to retain more potential instances in dense trees. To evaluate the effectiveness and robustness of our proposal, we curated two ITS datasets constructed with imagery captured by UAVs, namely instance segmentation in Conghua images dataset (iSCHID) and instance segmentation in Maoming images dataset (iSMMID) based on their respective spatial characteristics. Experimental results on both two datasets demonstrated that CEDAnet yields competitive results in ITS tasks, with the bounding box AP of 0.498, segmentation AP of 0.493 in iSCHID, and the bounding box AP of 0.706, segmentation AP of 0.703 in iSMMID.**

***Index Terms*—Benchmark dataset, deep learning (DL), individual tree segmentation (ITS), instance segmentation, unmanned aerial vehicle (UAV).**

I. Introduction

INDIVIDUAL tree segmentation (ITS), which means segmenting the contour of each tree and identifying each unique tree, is essential for forests and agriculture management [1], [2], [3]. ITS involves separating the relevant pixels, forming precise masks, and creating a distinct identifier for each tree in remote sensing (RS) images. Specifically, ITS of fruit trees offers basic data to enable growth monitoring, disease detection, and other related orchard planting applications. Traditional ITS based on RS images mainly relies on manual visual interpretation [4]. Recently, automatic segmentation technology reduces abundant labor costs and time consumption, and improves the generalization and efficiency [5], [6].

In recent years, unmanned aerial vehicles (UAVs) have provided the possibility for precise ITS with the ultra-high spatial resolution images [7], [8]. Satellite-based RS has made some remarkable progress in large-scale ITS, especially for sparse forests in Africa and regular economic forests [9]. However, faced with low height and dense distribution of artificial fruit trees in orchards, satellite-based RS with relatively coarse resolution makes it difficult to characterize the detailed morphology. UAVs with higher spatial resolution of centimeter level [10] have become valuable tools in orchard management and precision agriculture [11], [12], and its fine-grained detail allows the precise analysis of individual fruit tree crowns, enabling targeted interventions and management strategies.

Although UAVs provide unprecedented clear details, classical ITS methods like morphological processing are hard to deal with the complex and redundant information in UAVs images. Morphological algorithms extract rough tree boundary information in an unsupervised manner, including local maximum [13], edge detection algorithm [14], watershed algorithm [15], and region growth algorithm [16]. However, these morphology-based approaches face the problem of complex spatial structure in UAVs images, such as the confusion with the surrounding grass and shrub pixels. Moreover, these methods often rely on manual intervention and threshold adjustment, which are difficult to be applied on a large scale. Therefore, supervised algorithms with strong spatial feature representation capability are needed to extract accurate tree information from UAV images.

Due to the powerful spatial feature mining ability and generalization performance of deep learning (DL) [17], [18], [19], ITS has achieved significant improvement by using various deep networks. Especially, the proposal-based framework has gained significant popularity in object detection systems, including both one-stage and two-/multistage methods [20], [21]. The paradigm generally has a two-step pipeline: first, generating excessive object proposals in handcraft (e.g., predefined anchors) or learnable [e.g., region proposal networks (RPNs)] manner. The success of this framework is largely attributed to the ability of the RPN to generate multiscale and translation-invariant region proposals.

Manuscript received 31 December 2023; revised 22 February 2024; accepted 1 March 2024. Date of publication 19 March 2024; date of current version 29 March 2024. This work was supported in part by the National Key R&D Program of China under Grant 2022YFB3903402, and in part by the National Natural Science Foundation of China under Grant 42222106. *(Corresponding author: Haoyang Li.)*

The authors are with the Guangdong Provincial Key Laboratory for Urbanization and GeoSimulation, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China (e-mail: zhufj6@mail2.sysu.edu.cn; chenzhh239@mail2.sysu.edu.cn; lihy256@mail2.sysu.edu.cn; shixi5@mail.sysu.edu.cn; liuxp3@mail.sysu.edu.cn).

Both two datasets and the code of CEDAnet will be publicly available at https://github.com/Lihy256/CEDAnet.

Digital Object Identifier 10.1109/JSTARS.2024.3378167

And then, predicting a single instance corresponding to each proposal box with a confidence score and a refined location. To remove duplicate predictions, methods, such as nonmaximum suppression (NMS) [22], are usually required for postprocessing. Object detection and instance segmentation offer numerous advantages over direct segmentation for ITS in DL. They enable fine-grained segmentation by accurately labeling each object instance, providing detailed information about object position, shape, and size, which facilitates object recognition and tracking tasks. In addition, instance segmentation provides semantic understanding by revealing object relationships within the scene, aiding in better scene interpretation.

However, the dense distribution of trees with similar characteristics in orchards still presents challenges for DL-based methods. Some studies have focused on achieving precise segmentation of dense instances. One approach is to improve the segmentation performance of dense instances by utilizing multiscale processing [23]. This involves using a multiscale image pyramid to capture object information at different scales and performing segmentation and fusion at different scales. To guide the model optimization process, another approach is to design effective loss functions that accurately measure the differences between predicted results and ground truth segmentation [24]. However, from the overlooked perspective of UAVs, the shapes and colors appear highly similar in the dense trees compared to natural images from a head-up perspective. As a result, it is difficult for a detector to generate distinguishing predictions for each proposal, respectively, at the areas of high overlap. Moreover, since the substantial intermingling of instances, the predictions are prone to be mistakenly suppressed by NMS, resulting in discarded instances that should have been retained. Meanwhile, few works provide annotated datasets for dense ITS, which greatly limits the algorithm development and application.

To deal with the above problems, in this article, we propose a context enhancement and density adjustment network called CEDAnet. First, in order to overcome the difficulty of segmenting dense fruit trees, we build an improved density-guided NMS module (DG-NMS) to retain more potential instance bounding boxes and reduce false positives (FPs). DG-NMS utilizes the foreground density of tree pixels within the bounding box as prior knowledge to dynamically adjust the threshold. Second, to distinguish between different instances and refine their boundaries, we construct a transformer-based contextual aggregation module (TCAM). This module is designed to aggregate contextual information on the feature map, thereby enhancing the overall context for improved boundary delineation. Two datasets, instance segmentation in Conghua images dataset (iSCHID) and instance segmentation in Maoming images dataset (iSMMID), specifically curated for dense fruit tree instance segmentation, have been annotated and made publicly available.

There are three following contributions in this article.

1) In CEDAnet, the DG-NMS algorithm achieves more precise preservation of individual tree extraction in dense orchards through dynamic threshold adjustment methods.
2) TCAM leverages the powerful context aggregation capabilities of transformers to capture long-range dependencies and contextual information, enabling precise segmentation of individual fruit trees even in dense orchards.
3) We offer two ITS datasets captured via UAVs for fruit tree instance segmentation research. The iSCHID dataset highlights densely distributed fruit trees, with experiments confirming its contributions to instance segmentation in complex scenes.

II. Related Work

A. Deep-Learning-Based ITS

In practice, the undertaking of ITS from images is familiar to us, and over the years, numerous researchers have developed DL methods to explore research in this domain. For instance, in 2019, Wang et al. [20] utilized faster R-CNN for the segmentation of individual rubber trees. Weinstein et al. [21] proposed a semisupervised DL method for identifying individual trees, incorporating light detection and ranging for the unsupervised generation of training samples. Ferreira et al. [25] employed a fully convolutional neural network (CNN) to extract tree crowns from images captured by UAVs over Amazon palm trees. Plesoianu et al. [26] introduced the single shot detector (SSD), a DL ensemble design. Miyoshi et al. [27] proposed a novel hyperspectral image DL method for recognizing single tree species in high-density regions. Culman et al. [28] implemented a segmentation model based on CNNs for tree segmentation in high-resolution aerial images, followed by similar work by Sun et al. [29]. In 2021, Korznilov et al. [30] used a U-Net-like network for tree recognition, while Chen et al. [31] combined voxelization strategy with the PointNet DL framework to accurately segment individual trees from point clouds. Simultaneously, Zamboni et al. [32] benchmarked DL methods based on anchor and anchor-free, such as RetinaNet and Faster R-CNN did not exhibit satisfactory performance in RS.

Furthermore, results vary significantly when detecting and identifying different tree species in different scenes. The researches conducted by some researchers has provided us with cases and methodologies for ITS in diverse scenarios and for different tree species. For instance, in different scenarios, Tang et al. [33], Iqbal et al. [34], and Safonova et al. [35] conducted extensive work in plantations. Braga et al.'s [36] research focused on ITS in tropical forests, while Ocer et al. [37], Zhang et al. [38], and Sun et al. [39] explored urban trees. Yang et al. [40] investigated trees in urban parks, and Guirado et al. [41] studied trees in arid lands. Regarding different tree species, works have been conducted on various types, such as Brazilian palm trees [25], Brazilian nut trees [42], other tree species in the Amazon [43], and lychee trees [44]. Their studies contribute valuable insights for navigating the challenges associated with ITS across varied environmental contexts and tree types.

B. Instance Segmentation

Instance segmentation, a pivotal task in computer vision, involves identifying individual object instances and performing pixelwise segmentation. DL, notably with CNNs, has significantly advanced this field. Fully convolutional network [45] introduced in 2015 addressed semantic segmentation but lacked the ability to distinguish between instances within the same category. In response, faster R-CNN, proposed by Ren et al.

[46], introduced an RPN for object detection, setting the stage for subsequent instance segmentation methods. Mask R-CNN [47], building upon faster R-CNN, integrated a mask branch for fine-grained instance segmentation masks, marking a significant stride in seamlessly combining object detection and instance segmentation within a single framework.

These methodologies demonstrated the transition of single-tree segmentation from semantic segmentation to object detection, effectively highlighting the capabilities of DL models. They established a robust foundation for addressing complex instance segmentation challenges. Concurrently, we aimed to contribute to this field by employing a transformer-based contextual enhancement for the instance segmentation of densely populated lychee trees.

III. Materials and Method

A. Dataset

In this part, we will introduce two datasets prepared for the experiment. Both datasets, captured through UAV imagery, showcase dense lychee orchards in China's Guangdong province—one situated in Guangzhou and the other in Maoming.

The selected datasets exhibit different planting densities, providing diverse perspectives and scenes, thereby enriching the diversity of training data. Such diversity facilitates the model's better understanding of the visual characteristics and growth patterns of lychee trees under varying planting densities. The model adapts to the visual and morphological variations of lychee trees across different planting densities, contributing to enhanced generalization performance. This enables the model to accurately identify and segment lychee trees of various densities in practical applications. Meanwhile, the intentional selection of diverse locations adds intrinsic value to our experiment, allowing us to explore and analyze the impact of geographical variability on our research outcomes. This geographical contrast enriches the datasets and enhances the universality of our findings, emphasizing these selected sites make a distinctive contribution to the boarder academic community. These datasets are publicly available to meet different research needs.

1) iSCHID: The iSCHID focuses on the Li-Bo Park area in Conghua District, Guangzhou City, Guangdong Province, China. The orchard covers an area of about 200 km^2, and a large number of litchi trees is planted, as shown in Fig. 1. The images of the study area were taken by DJI M30 UAV, and the flight patrol was carried out with a 75% side overlap rate, 75% heading overlap rate, and navigation height of 200 m. Finally, the captured images were processed by DJI mapping software to automatically build the digital orthophoto map of the study area.

We geospatially divided the study area into training and test areas. In the training area, we sliced the digital orthophoto map and clipped it with an overlap rate of 70%. The training and validation data were collected from the training area, the test data were collected in the test area, and the data were manually annotated on the Labelme software as shown in Fig. 1. The training and testing datasets contain the 265 and 64 pairs of data.

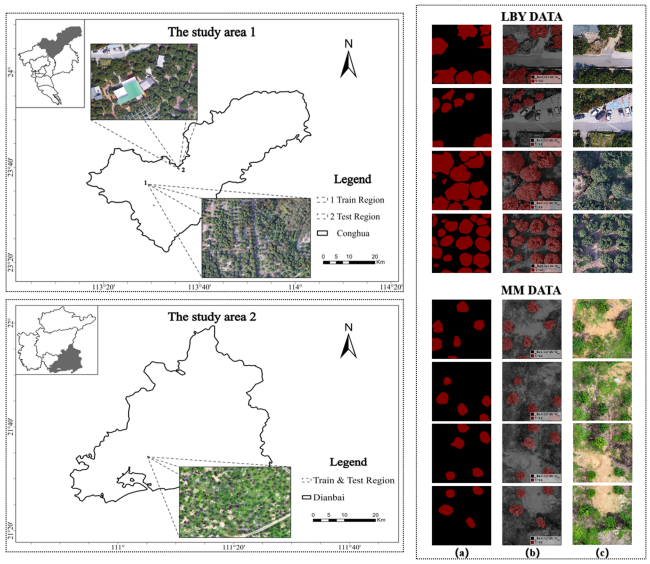

Fig. 1. Two study areas and datasets. (a) Ground truth. (b) Mask on image. (c) UAV Image.

2) iSMMID: The study area of iSMMID is located in Dianbai District of Maoming City, Guangdong Province, which covers an area of 800 km^2. A large number of litchi trees are also planted in the orchard, as shown in Fig. 1. In the same way as the iSCHID dataset, we also build the digital orthophoto map of the study area by DJI M30 UAV, then slice the digital orthophoto map and clip them with a 25% overlap rate. We manually annotate individual tree on the Labelme software. The training and testing datasets were randomly split and contained 270 and 55 pairs of images, respectively.

B. CEDAnet

The overall framework of the proposed CEDAnet method is illustrated in Fig. 2. First, given an input UAV image, we use the ResNet50 [46] to generate the feature map. Second, we utilize the TCAM for in-depth supervision of the feature map derived from ResNet50, subsequently replacing original feature map. During object detection on the new feature maps, the DG-NMS algorithm is employed. This algorithm dynamically adjusts thresholds utilizing prior density knowledge to more judiciously retain bounding boxes. Finally, using fully connected layer, the classifier outputs a score for each class, the bounding box regression predicts the precise object localization and the mask branch outputs the segmentation results. We next describe our module of the proposed framework.

1) TCAM: To further enhance the modeling and integration of multiscale information extracted by the feature extractor, we introduce a context enhancement module within CEDAnet, denoted as TCAM.

TCAM, comprising a token encoder and a decoder within the transformer architecture, efficiently captures contextual information from feature maps. In the context of ITS, TCAM excels in handling intricate canopy patterns, enhancing the ability to compute losses with mask labels. This module plays a pivotal

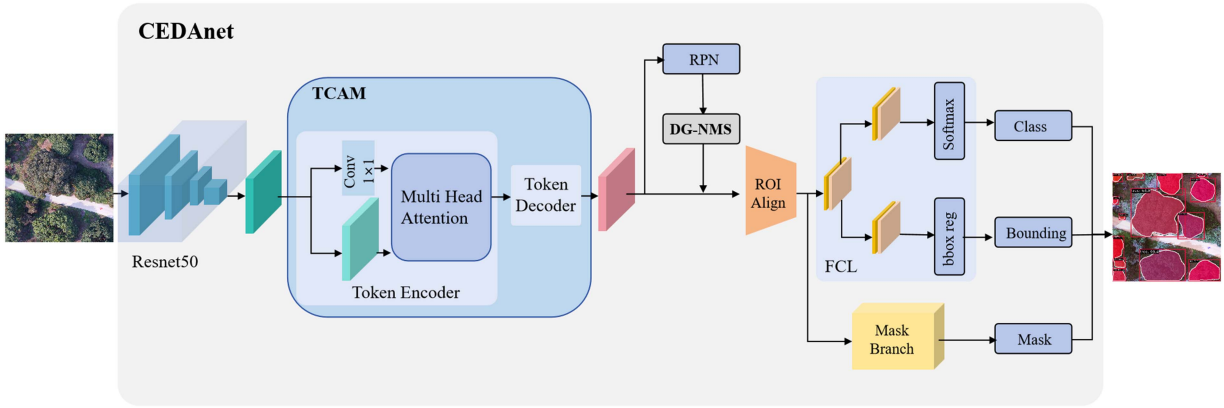

Fig. 2. Illustration of the designed CEDAnet where TCAM is a transformer-based module to aggregate context and DG-NMS is a density-guided NMS algorithm.

role in discerning fine-grained details within the canopy structure, contributing significantly to the model's overall accuracy and precision in ITS tasks.

a) Token encoder: Utilizing a context enhancement module based on the transformer to process the feature maps obtained from the backbone, the original feature maps outputted by ResNet50 are replaced.

The obtained 32 × 32 feature map is inputted to the token encoder, employing spatial attention and transformer modules for token encoding. This step is essential to enhance context and extract tree-related features. Spatial attention ensures relevant features are highlighted, contributing to the overall texture enhancement effect.

The input feature map $F \in R^{b\times c\times h\times w}$ experiences a 1 × 1 convolution to adjust channel numbers, resulting in $F' \in R^{b\times l\times h\times w}$, F' is reshaped to a token $f' \in R^{b\times l\times(h\times w)}$. Simultaneously, the original feature map, without convolution, is reshaped to $f \in R^{b\times c\times(h\times w)}$.

An einsum operation is applied to $f \in R^{b\times c\times(h\times w)}$ and $f' \in R^{b\times l\times(h\times w)}$, eliminating the last dimension to obtain $t \in R^{b\times c\times l}$, represented as

$$t_{bl} = f'(hw)\, f_{bc}(hw) \tag{1}$$

where b, c, h, w, and l represent batch size, channel number, height and width of the feature map, and token length, respectively.

Next, position embedding is introduced through elementwise addition to each position of the token t. The resulting t' is expanded using a linear layer and inputted into the multihead self-attention (MHA) module, capturing contextual information between tokens. This attention mechanism enhances the model's ability to discern intricate textures related to trees.

The architecture of the MHA module is as follows. MHA extends t' to a new embedding t using a linear layer, represented as

$$t' = tW^I, t' \in R^{b\times l\times(n\times d\times 3)} \tag{2}$$

where W^I is the weight of the linear layer, n is the head number of MHA, and d is the dimension for subsequent tensors. n and d are set for 8 and 64, respectively.

First, t' is normalized, and linear layers transform it into query ($Q \in R^{b\times n\times l\times d}$), key ($K \in R^{b\times n\times l\times d}$), and value ($V \in R^{b\times n\times l\times d}$), represented as

$$Q, K, V = t'W_i^Q, t'W_i^K, t'W_i^V \tag{3}$$

where W_i^Q, W_i^K, and W_i^V denote the weights of the linear layers to map Q, K, and V, respectively.

The scaled dot-product attention mechanism is then used to calculate attention values between query and key. The weighted sum of all t' using attention values yields attention maps, represented as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D}}\right)V. \tag{4}$$

Finally, the outputs of each head are concatenated and input to a linear layer, producing the final output of MHA, expressed as

$$\text{head}_i = \text{Attention}\left(t'W_i^Q, t'W_i^K, t'W_i^V\right),\ i \in (0, n]$$
$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_n)W^O \tag{5}$$

where W_i^Q, W_i^K, and W_i^V denote the weights of the linear layers of the ith head to map $Q, K,$ and V, respectively; and $W^O$is the weight of the last linear layer in MHA.

The final output of MHA is a concatenated representation of the outputs from each head, contributing to the overall enhancement of tree-related features.

In FFN, two linear layers and a Gaussian error linear unit activation [48] are used to further transform the learning token of MHA.

A larger, richer dictionary may enable the token encoder to better understand features in the input image, enhancing the model's ability to model image feature information and handle contextual information more effectively, achieving the desired context enhancement effect. Fig. 3 shows the token encoder module.

b) Token decoder: The token decoder receives two inputs: the convolution features F and the token embedding t from the token encoder. In the token decoder, before F and t are input to the transformer decoder, trainable parameters are added to F

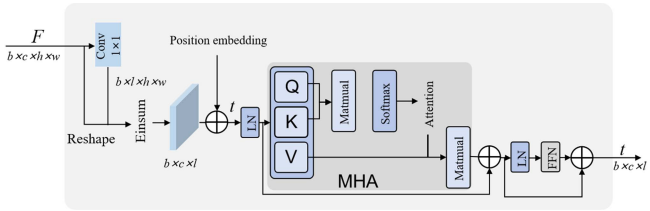

Fig. 3. Illustration of the token encoder.

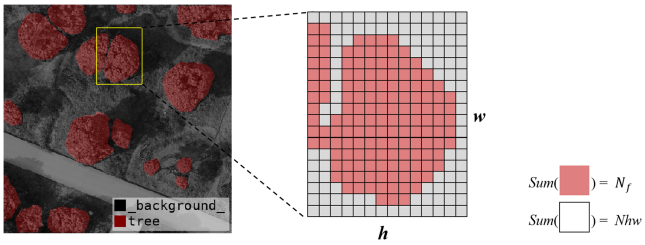

Fig. 4. Illustration of DG-NMS about foreground and background.

with the aim of reprojecting the token embedding back into pixel space and enhancing F.

In the token decoder, before the use of the MHA module, weight-sharing LN is applied to both F and t. This is similar to the MHA in the token encoder. The key distinction lies in the fact that the query is derived from F, while the key and value mappings come from t. The final output image maintains the same size as the input image.

2) DG-NMS: After generating bounding boxes in the RPN, NMS method is usually used to suppress excessive boxes. But in scenarios of substantial object overlap, the traditional NMS selectively preserves only the box with the highest score, resulting in the removal of the prediction box for the other heavily overlapped object.

To address this limitation, we introduce DG-NMS, a method that dynamically adjusts thresholds to resolve the issues inherent in traditional NMS. DG-NMS makes an important contribution by adapting to varying degrees of object overlap, ensuring more accurate preservation of prediction boxes in scenarios with substantial overlap. First, for bounding boxes (M) whose overlap with the highest scoring box exceeds the predefined threshold N_t, their confidence is diminished rather than promptly excluded. This approach allows more boxes to be retained, mitigating the occurrence of overlaps to some extent. In the vicinity of the same object, there are often multiple boxes. By consistently selecting the box with the highest score, the surrounding boxes are suppressed. The degree of suppression increases with a higher Intersection over Union (IOU) with the highest scoring box. Generally, the IOU of boxes representing the same object tends to be greater than that of boxes representing different objects. Consequently, this process retains boxes for other objects while removing those for the same object. DG-NMS reduces scores in a way that avoids forcefully setting the scores of overlapping boxes to zero. Instead, it employs a gentler suppression mechanism

$$S_i = \begin{cases} S_i\,, & \text{iou}\,(M, b_i) < N_t \\ S_i\,(1 - \text{iou}\,(M, b_i))\,, & \text{iou}\,(M, b_i) \geq N_t \end{cases} \tag{6}$$

where S_{i} represents the score of a detected bounding box, M denotes the highest score detected bounding box, and b_i represents the one detected bounding box.

Second, based on the position of the target box, the original image patch corresponding to the box position can be obtained, and we can apply the Otus binarization [48] method to separate the foreground and background of this image slice. The Otus algorithm is a classical algorithm used for image segmentation, aiming to automatically determine the threshold of image grayscale levels and divide the image into background and foreground parts. It is to determine the optimal threshold by maximizing the between-class variance. The between-class variance is a measure of the difference between the two segmentation parts of the image. When the between-class variance is maximized, it means that an optimal threshold has been found, which can best separate the foreground and background.

Typically, slices from densely distributed fruit trees should have a much higher foreground pixel ratio than the background pixel ratio. Therefore, the NMS threshold can be appropriately adjusted based on the foreground pixel ratio, allowing more boxes to be retained. The threshold N_t calculation can be expressed as

$$N_t = \frac{N_f}{Nhw} \tag{7}$$

where N_t represents the threshold, N_f indicates the number of foreground pixels after Otus binarization, and Nhw refers to the total number of pixels in the original image corresponding to the target. Fig. 4 shows the DG-NMS.

DG-NMS further refines this process by adjusting the factor and incorporating the DG-NMS threshold. The DG-NMS threshold is determined based on the foreground pixel count after Otus binarization. We use it as prior knowledge to achieve adaptive adjustment of NMS threshold. This adaptation ensures a balanced preservation of overlapping boxes and optimized NMS thresholds based on the foreground pixel distribution.

IV. Experiments

A. Implementation Details

We clip the digital orthophoto map of the study area to 512 × 512 slice data for train and test. Construction and preprocessing of the datasets are completed by QGIS, GDAL-python, and MATLAB.

For the model training, the optimizer is Adam; the initial learning rate is 0.01; the beta is 0.9 and 0.999; and the weight decay is 0.0001. The learning rate decay strategy is StepLR with step size as 10, gamma as 0.8, and last epoch as 200. All models are trained for 250 epochs. Our models and experiments are based on the open-source DL framework MMDetection [49]. The experimental environment is Centos 7.5. 1804. The GPU

is GeForce RTX 4090ti. The CPU is Intel(R) Xeon(R) CPU E5-2680.

B. Comparisons With Baseline Methods

1) Comparative Models:

a) Mask R-CNN [47]: Mask R-CNN is a DL-based object detection and instance segmentation model that can simultaneously predict object bounding boxes and pixel-level masks, enabling accurate object detection and fine-grained instance segmentation. By adding an additional branch to the faster R-CNN architecture [46], mask R-CNN predicts pixel-level masks for objects, achieving the capability of both detection and segmentation.

b) Cascade mask R-CNN [51]: Cascade mask R-CNN is a mask R-CNN based instance segmentation model that utilizes a cascaded structure to progressively improve the detector's accuracy. It achieves higher detection performance and lower FP rates by cascading multiple detection stages, with each stage performing stricter filtering based on the results of the prevIOU stage.

c) Mask scoring R-CNN [52]: Mask scoring R-CNN (MS R-CNN) is another mask R-CNN based on instance segmentation model that aims to improve the quality of instance segmentation masks. It introduces an MS branch to evaluate the quality of predicted masks, enabling the model to assign higher scores to more accurate masks and filter out low-quality ones, leading to enhanced segmentation results.

d) SOTR [53]: Segmenting objects with transformers is a method for object detection and segmentation based on transformer architecture. By employing row–column separated attention mechanism, it captures relationships between different positions in images. Coupled with convolutional layers and activation functions, it enables end-to-end object detection and segmentation.

e) Instaboost [54]: Instaboost is a data augmentation method for object detection and segmentation that enhances the robustness and generalization of the model by dynamically adjusting and resampling samples during training. It effectively alleviates class imbalance and hard example issues, improving the performance and stability of instance segmentation models.

2) Evaluation: In this article, we use the average precision (AP) to evaluate the results. Among different annotated datasets used by object detection challenges and the scientific community, the most common metric used to measure the accuracy of the detections is the AP. Before examining the variations of the AP, we should review some concepts that are shared among them. The most basic are the ones defined as follows: true positive: a correct detection of a ground-truth bounding box; FP: an incorrect detection of a nonexistent object or a misplaced detection of an existing object; false negative: an undetected ground-truth bounding box. It is important to note that, in the object detection context, a true negative result does not apply, as there are infinite number of bounding boxes that should not be detected within any given image. The above definitions require the establishment of what a "correct detection" and an "incorrect detection" are.

TABLE I
EXPERIMENTAL RESULTS WITH OTHER NETWORKS ON ISCHID

	bbox					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
Mask R-CNN	0.466	0.753	0.514	0.171	0.418	0.593
Cascade Mask R-CNN	0.469	0.755	0.519	0. 175	0. 421	0. 601
MS R-CNN	0.481	0.748	0. 524	0. 18	0. 432	0. 597
SOTR	0.469	0.741	0. 495	0. 19	0. 406	0. 592
Instaboost	0.453	0. 75	0. 483	0. 179	0. 407	0. 57
Ours	**0.498**	**0.769**	**0. 561**	**0. 217**	**0. 449**	**0. 63**
	Segmentation					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
Mask R-CNN	0.462	0.742	0.532	0.112	0.413	0.611
Cascade Mask R-CNN	0.466	0.748	0.541	0.12	0.433	0.625
MS R-CNN	0.466	0.741	0. 525	0. 12	0. 406	0. 595
SOTR	0.467	0.713	0. 484	0. 144	0. 451	0. 606
Instaboost	0.433	0.727	0. 469	0. 099	0. 389	0. 566
Ours	**0.493**	**0.793**	**0. 571**	**0. 19**	**0. 458**	**0. 636**

A common way to do so is using the IOU. It is a concept usually used in object detection and semantic segmentation, and is the overlap rate between the generated candidate bound and the original ground truth bound, that is the ratio of their intersection to union. In this experiment, we utilize IOU within the DG-NMS framework to compute the overlap between two candidate bounding boxes and determine whether to retain them based on a threshold. Here, the calculation of IOU will affect the results of AP.

The numerator part calculates the number of pixels to correctly predict in the foreground, and the denominator part calculates the number of pixels in the images and sets the real foreground and predicted foreground. The process of calculating a debit note can be expressed as

$$\text{IOU} = \frac{\text{TP}}{\text{FN} + \text{TP} + \text{FP}}. \tag{8}$$

AP is a commonly used precision evaluation metric in object detection tasks. It measures the accuracy and recall of the detection results at different confidence thresholds. First, based on the model's output, the detection results are sorted in descending order of confidence. Then, for different confidence thresholds, the precision and recall are calculated. Precision represents the proportion of correctly predicted positive samples among all samples predicted as positive by the model. Recall represents the proportion of correctly predicted positive samples among all actual positive samples. Next, the precision and recall values are

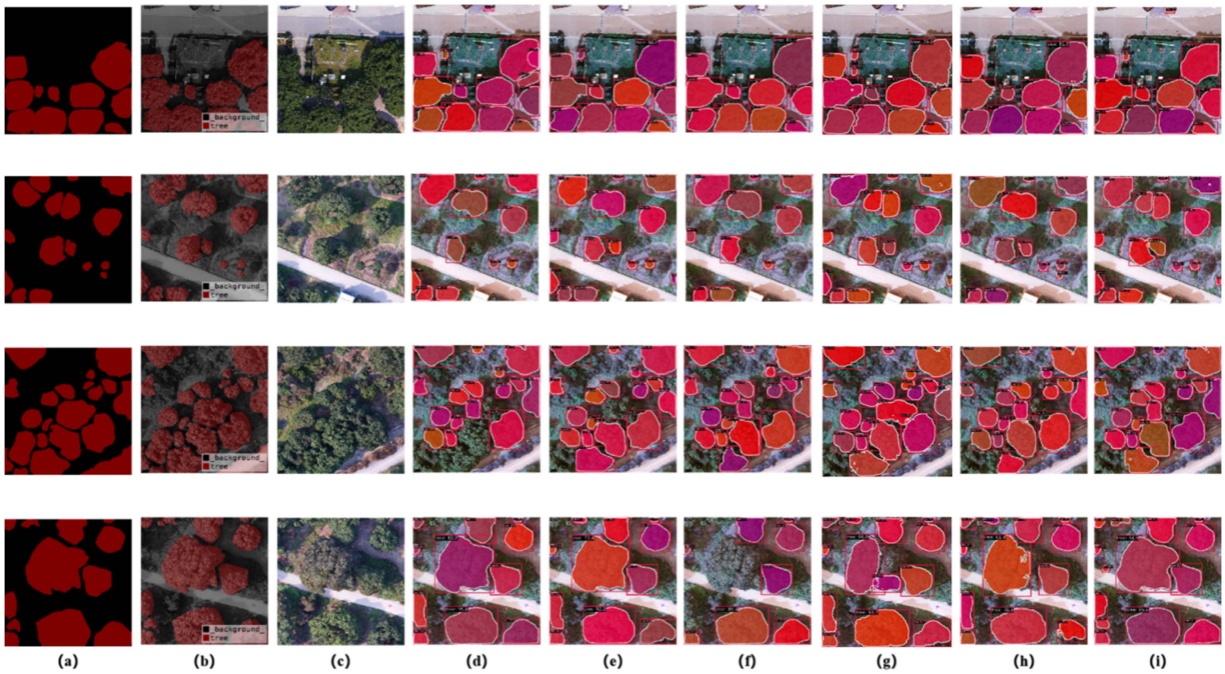

Fig. 5. Some visualization results on iSCHID. (a) GT. (b) Mask. (c) UAV image. (d) Mask R-CNN. (e) Cascade R-CNN. (f) MS mask R-CNN. (g) SOTR. (h) Instaboost. (i) Ours.

calculated for each confidence threshold, and a precision–recall curve is plotted. Finally, the area under the curve, known as AP, is computed. The value of AP ranges from 0 to 1, with higher values indicating more accurate detection results. Typically, AP decreases as the confidence threshold increases. The AP is evaluated with different IOU. It can be calculated for 10 IOU varying in a range of 50% to 95% with steps of 5%, usually reported as AP@50:5:95. It also can be evaluated with single values of IOU, where the most common values are 50% and 75%, reported as $AP_{50}$and AP_{75}, respectively. The AP is determined for objects in three different sizes: small (with area $<32^2$ pixels), medium (with $32^2<$ area $<96^2$ pixels), and large (with area $>96^2$ pixels)

3) Comparison of iSCHID: We tested our own model on the dataset and compared it with state-of-the-art instance segmentation models. As can be seen from Table I, the proposed method CEDAnet outperforms all baselines on the iSCHID, achieving the highest AP scores on test datasets. Our method has achieved good results in both target box detection and target segmentation, especially in the detection and segmentation of small targets, which can also prove that our weak supervision module plays a certain role. By comparing with other instance segmentation models, our model demonstrated the highest accuracy in terms of numerical evaluation. As Table I shows, Instaboost obtains the lowest AP^{bbox} and AP^{seg} of 0.453 and 0.433 among the compared methods, which is followed by mask R-CNN. Mask R-CNN obtains an AP^{bbox} of 0.466 and an AP^{seg} of 0.462. SOTR obtains the AP^{bbox} and AP^{seg} of 0.469 and 0.467, which shows the SOTR can slightly improve the model accuracy here. Our method achieved the highest AP^{bbox} of 0.498 and an AP^{seg} of 0.493 among all the compared methods, which denote that context enhancement module is capable of capturing and leveraging features of ITS in the image more effectively, thereby enhancing segmentation performance and providing more precise results for practical applications.

Below, we will showcase some details on the result images. Fig. 5 provides more intuitive pictures of each network's performance on the iSCHID. Mask R-CNN which has the ability to segment relatively apparent individual fruit trees, but not do well in extracting the high-density boundaries of trees and small trees. Among the deep networks, cascade mask R-CNN and MS R-CNN can get more detailed detection and segmentation results. However, the models still struggle to effectively segment individual trees in large areas with dense clusters of fruit trees. SOTR, as an instance segmentation method based on transformer, demonstrates suboptimal performance in the experiments. Our proposal gives more correct bounding box results and refined boundaries of trees (row 1 of Fig. 5).

4) Comparison of iSMMID: The spatial distribution of fruit trees in this dataset is not as dense as iSCHID. Similarly, our method still achieved promising results.

As can be seen from Table II, the proposed method also outperforms all baselines on the iSMMID dataset, achieving the highest AP^{bbox} and AP^{seg} scores of 0.706 and 0.703, respectively. The second-ranked MS R-CNN obtains an AP^{bbox} of 0.699 and an AP^{seg} of 0.694, which further proves the advancement of context enhancement. The mask R-CNN scores 0.691 AP^{bbox} and 0.667 AP^{seg}, while the cascade mask R-CNN scores 0.695 AP^{bbox} and 0.671 AP^{seg}. This shows that the aggregation of features can slightly improve the model accuracy here. SOTR obtains the AP^{bbox} and AP^{seg} scores of 0.701 and 0.693, which shows that proposal boxes have significant advantages in improving accuracy.

TABLE II
EXPERIMENTAL RESULTS WITH OTHER NETWORKS ON ISMMID

iSMMID						
	bbox					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
Mask R-CNN	0.691	0.973	0. 894	−1	0. 674	0. 714
Cascade Mask R-CNN	0.695	0.979	0. 899	−1	0. 681	0. 723
MS R-CNN	0.699	0.977	0. 889	−1	0. 689	0. 738
SOTR	0.701	0.979	0. 892	−1	0. 685	0. 751
Instaboost	0.701	0.977	0. 886	−1	0. 69	0. 745
Ours	**0.706**	**0.983**	**0. 899**	**−1**	**0. 697**	**0. 752**
	Segmentation					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
Mask R-CNN	0.667	0.959	0. 963	−1	0. 635	0. 701
Cascade Mask R-CNN	0.671	0.963	0. 961	−1	0. 643	0. 712
MS R-CNN	0.694	0.977	0. 913	−1	0. 68	0. 739
SOTR	0.693	0.988	0. 938	−1	0. 669	0. 715
Instaboost	0.688	0.986	0. 904	−1	0. 678	0. 722
Ours	**0.703**	**0.988**	**0. 971**	**−1**	**0. 661**	**0. 745**

Fig. 6 demonstrates the behavior of different methods on the iSMMID. Our method can precisely segment small instances (row 4 of Fig. 6). It can be observed that our model achieves more precise delineation of boundaries for small objects. Mask R-CNN can roughly segment the boundaries of trees, but there is still a significant gap between their boundary contours and the annotated mask. Because of the refinement of spatial context information during feature extraction, the cascade mask R-CNN is better to keep precise boundaries of these small objects, while it is relatively rough and supersaturated compared to the ground truth. SOTR, as an instance segmentation method based on transformer, do not show good performance on this dataset as well. This demonstrates that the integration of our modules can help to restore the spatial information and improve the ITS accuracy.

C. Ablation Experiment

CEDANet integrates both TCAM blocks and DG-NMS module for accurate instance segmentation. We design ablation experiments to verify the improvement of context enhancement and prior knowledge for instance segmentation. In Tables III and IV, the "Base" baseline denotes the basic model mask R-CNN. The "Base+TCAM" model means mask R-CNN with context aggregator to context enhancement. The "Base+DG-NMS" model means mask R-CNN with DG-NMS algorithm to dynamically adjust the NMS threshold by acquiring prior knowledge of densely populated trees. The "CEDAnet" model is the proposal of this article with the transformer-based context aggregator module and dense-guided NMS algorithm.

As can be seen from Table III, the incorporation of both TCAM and DG-NMS can improve the model performance on

TABLE III
EXPERIMENTAL RESULTS WITH OTHER NETWORKS ON ISCHID

	bbox					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
Base	0.466	0.753	0.514	0.171	0.418	0.593
Base+TCAM	0.475	0.756	0.528	0.215	0.434	0.619
Base+DG-NMS	0.492	0.767	0.552	0.205	0.448	0.605
CEDAnet	**0.498**	**0.769**	**0.561**	**0.217**	**0.449**	**0. 63**
	Segmentation					
	AP^{seg}	AP^{seg}_{50}	AP^{seg}_{75}	AP^{seg}_{S}	AP^{bseg}_{M}	AP^{seg}_{L}
Base	0.462	0.742	0.532	0.112	0.413	0.611
Base+TCAM	0.478	0.752	0 557	0.157	0.43	0.639
Base+DG-NMS	0.473	0.75	0.521	0.13	0.418	0.636
CEDAnet	**0.493**	**0.793**	**0.571**	**0.19**	**0.458**	**0.636**

TABLE IV
EXPERIMENTAL RESULTS WITH OTHER NETWORKS ON ISMMID

	bbox					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
Base	0.691	0.973	0.894	−1	0.674	0.714
Base+TCAM	0.697	0.987	0.894	−1	0.679	0.725
Base+DG-NMS	0.697	0.987	0.894	−1	0.679	0.725
CEDAnet	**0.706**	**0.983**	**0.899**	**−1**	**0.697**	**0.752**
	Segmentation					
	AP^{seg}	AP^{seg}_{50}	AP^{seg}_{75}	AP^{seg}_{S}	AP^{bseg}_{M}	AP^{seg}_{L}
Base	0.667	0.959	0.963	−1	0.635	0.722
Base+TCAM	0.678	0.987	0.865	−1	0.654	0.722
Base+DG-NMS	0.685	0.988	0.889	−1	0.661	0.725
CEDAnet	**0.703**	**0.988**	**0.971**	**−1**	**0.661**	**0.745**

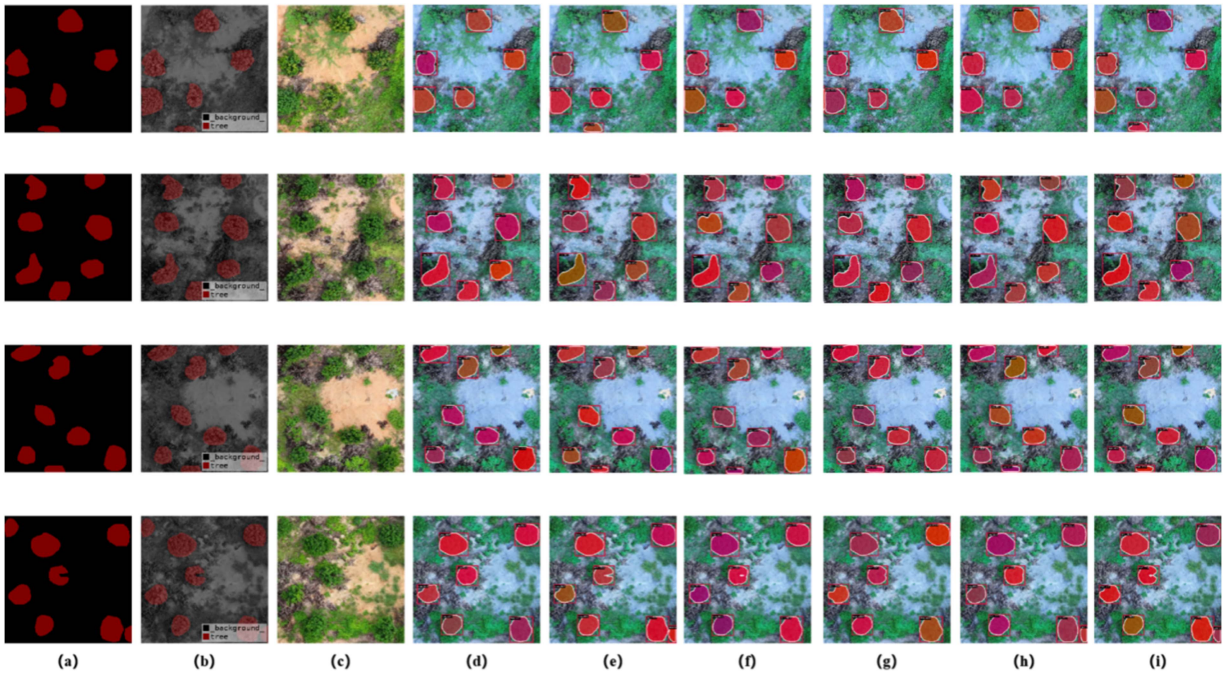

Fig. 6. Some visualization results on iSMMID. (a) GT. (b) Mask. (c) UAV image. (d) Mask R-CNN. (e) Cascade R-CNN. (f) MS mask R-CNN. (g) SOTR. (h) Instaboost. (i) Ours.

both datasets. More specifically, the AP^{bbox} and AP^{seg} can be improved by 1.9% and 3.4% on the iSCHID and by 0.86% and 1.6% on the iSMMID after adding TCAM layers. It shows that deep supervision on the feature map does enhance the ability of the model. Besides, the DG-NMS can improve the AP^{bbox} and AP^{seg} of the iSCHID by 5.6% and 2.3% and those of the iSMMID by 1.6% and 2.7%, respectively; this indicates that DG-NMS can contribute substantially to the subsequent metric learning by making the feature pairs more distinguishable from each other. Notably, compared to the "Base" model, the AP^{bbox} and AP^{seg} of the CEDAnet with both TCAM and DG-NMS integrated are increased by 6.9% and 6.7% on iSCHID, and 2.2% and 5.4% on iSMMID, respectively. The great improvement of the CEDAnet not only further proves the effectiveness of TCAM and DG-NMS, but also proves the gain effect of their combination.

TABLE V
EXPERIMENTAL RESULTS WITH OTHER NETWORKS ON ISCHID

	bbox					
	AP^{bbox}	AP^{bbox}_{50}	AP^{bbox}_{75}	AP^{bbox}_{S}	AP^{bbox}_{M}	AP^{bbox}_{L}
t=0. 3	0.423	0.724	0.426	0.146	0.383	0.535
t=0. 4	0.422	0.739	0.475	0.148	0.403	0.58
t=0. 5	0.432	0.736	0.473	0.157	0.406	0.571
t=0. 6	0.466	0.753	0.514	0.171	0.418	0.593
t=0. 7	0.459	0.742	0.492	0.167	0.429	0.61
t=0. 8	0.448	0.738	0.492	0.152	0.411	0.598
Base+DG-NMS	**0.492**	**0.767**	**0.552**	**0.205**	**0.448**	**0.605**

	Segmentation					
	AP^{seg}	AP^{seg}_{50}	AP^{seg}_{75}	AP^{seg}_{S}	AP^{bseg}_{M}	AP^{seg}_{L}
t=0. 3	0.453	0.753	0.498	0.108	0.412	0.583
t=0. 4	0.463	**0.758**	0.508	0.110	0.41	0.618
t=0. 5	0.461	0.75	0.503	0.109	0.408	0.624
t=0. 6	0.462	0.742	**0. 532**	0.112	0.413	0.611
t=0. 7	0.458	0.742	0.518	0.110	0.413	0.635
t=0. 8	0.458	0.741	0.516	0.108	0. 414	0.624
Base+DG-NMS	**0.473**	0.75	0.521	**0.13**	**0.418**	**0.636**

V. DISCUSSION

A. NMS Threshold

The threshold for NMS is a hyperparameter in this experiment that directly determines whether multiple bounding boxes of dense regions are retained, thus directly affecting the effectiveness of dense instance segmentation. Therefore, we also designed an experiment to verify the rationality of dynamically adjusting the NMS threshold. We manually adjusted the NMS threshold to determine the optimal value on iSCHID. As can be seen from Table V, when the threshold is set to 0.6, the majority of the accuracy reaches its peak. However, setting the threshold to other values also yields optimal accuracy for certain cases. When we dynamically adjust the threshold by adding our DG-NMS module to the mask R-CNN model, all bounding box accuracy is significantly improved, and all segmentation accuracy is improved while the threshold is set to 0.4 that the

AP_{50}^{seg} gets highest and the threshold is set to 0.6 that AP_{75}^{seg} gets highest. In general, this demonstrates the rationality of adaptively adjusting the NMS threshold based on prior knowledge.

B. Application

Based on the segmentation results of ITS of fruit trees, it is possible to estimate the yield of orchards. By calculating the canopy area based on plant masks and determining the latitude and longitude coordinates using plant bounding boxes, it becomes feasible to estimate the fruit yield of each plant. By capturing aerial photos of fruit trees during the fruiting period using UAV and applying object detection algorithms to detect fruits, the number of fruits within the field of view can be counted. By combining the field of view angle parameters of the UAV and the canopy area of the specific plant, it becomes possible to collectively estimate the quantity or yield of fruits for that particular plant. This integrated approach leverages the power of spatial data analysis and RS technology to provide accurate and efficient yield estimation for fruit tree orchards. In the future, this article will conduct a yield assessment of the orchard.

VI. Conclusion

In this article, we have proposed a novel framework to tackle the problem of dense distribution in ITS. The key idea of the proposed method is to utilize a TCAM to encode and decode contextual information of instance features, facilitating feature fusion and enhancement. In addition, the DG-NMS algorithm is employed during the candidate box generation stage, dynamically adjusting thresholds to adaptively generate a sufficient number of candidates bounding boxes, thereby improving the performance of instance segmentation.

Experimental validations on two different ITS datasets (iSCHID and iSMMID) demonstrate the effectiveness of the proposed CEDAnet in segmenting individual trees within lychee orchards of varying density. Further ablation experiments confirm the effectiveness of integrating TCAM and DG-NMS. Specifically, TCAM is capable of acquiring and more fully utilizing information regarding fruit tree morphology, while DG-NMS enhances segmentation accuracy. Hence, our method is applicable to ITS of fruit trees, demonstrating its capability for instance segmentation in such complex and dense scenarios. In future article, we will continue to enhance this framework to improve its detection and segmentation performance.

References

[1] Y. Xiong, X. Zeng, Y. Chen, J. Liao, W. Lai, and M. Zhu, "An approach to detecting and mapping individual fruit trees integrated YOLOv5 with UAV remote sensing," Preprints, Apr. 2022, doi: 10.20944/preprints202204.0007.v2.

[2] H. Tian et al., "Extraction of citrus trees from UAV Remote sensing imagery using YOLOv5s and coordinate transformation," *Remote Sens.*, vol. 14, no. 17, Jan. 2022, Art. no. 4208, doi: 10.3390/rs14174208.

[3] Y. Mu et al., "Characterization of peach tree crown by using high-resolution images from an unmanned aerial vehicle," *Horticulture Res.*, vol. 5, no. 1, Dec. 2018, Art. no. 74, doi: 10.1038/s41438-018-0097-z.

[4] A. Ozdarici-Ok and A. Ö. Ok, "Using remote sensing to identify individual tree species in orchards: A review," *Scientia Horticulturae*, vol. 321, Nov. 2023, Art. no. 112333, doi: 10.1016/j.scienta.2023.112333.

[5] C. Xiao, R. Qin, X. Huang, and J. Li, "Individual tree detection from multi-view satellite images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2018, pp. 3967–3970.

[6] X. Guo et al., "Tree recognition on the plantation using UAV images with ultrahigh spatial resolution in a complex environment," *Remote Sens.*, vol. 13, no. 20, pp. 4122–4122, Oct. 2021, doi: 10.3390/rs13204122.

[7] H. Ren, Y. Zhao, W. Xiao, and Z. Hu, "A review of UAV monitoring in mining areas: Current status and future perspectives," *Int. J. Coal Sci. Technol.*, vol. 6, no. 3, pp. 320–333, Aug. 2019, doi: 10.1007/s40789-019-00264-5.

[8] C. Zhou, Z. He, A. Lou, and A. Plaza, "RGB-to-HSV: A frequency-spectrum unfolding network for spectral super-resolution of RGB videos," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5609318, doi: 10.1109/TGRS.2024.3361929.

[9] M. Popp and J. M. Kalwij, "Consumer-grade UAV imagery facilitates semantic segmentation of species-rich savanna tree layers," *Sci. Rep.*, vol. 13, no. 1, Aug. 2023, Art. no. 13892, doi: 10.1038/s41598-023-40989-7.

[10] K. Themistocleous, "The use of UAV platforms for remote sensing applications: Case studies in Cyprus," *Proc. SPIE*, vol. 9229, pp. 205–213, Aug. 2014, doi: 10.1117/12.2069514.

[11] F. Tong, H. Tong, R. Mishra, and Y. Zhang, "Delineation of individual tree crowns using high spatial resolution multispectral WorldView-3 satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 7751–7761, 2021, doi: 10.1109/jstars.2021.3100748.

[12] M. Wulder et al., "Individual tree recognition from multiple high spatial resolution image sources," in *Proc. Int. Geosci. Remote Sens. Symp.*, 2001, pp. 771–773. [Online]. Available: https://ieeexplore.ieee.org/document/976631

[13] J. Fan, M. Farmen, and I. Gijbels, "Local maximum likelihood estimation and inference," *J. Roy. Statist. Soc. Ser. B: Statist. Methodol.*, vol. 60, no. 3, pp. 591–608, Sep. 1998.

[14] D. Ziou and S. Tabbone, "Edge detection techniques—An overview," *Pattern Recognit. Image Anal.: Adv. Math. Theory Appl.*, vol. 8, no. 4, pp. 537–559, 1998.

[15] F. Meyer, "Color image segmentation," in *Proc. Int. Conf. Image Process. Appl. IET*, 1992, pp. 303–306.

[16] W. Xu and I. Cumming, "A region-growing algorithm for InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 124–134, Jan. 1999.

[17] L. Fang, Y. Jiang, H. Yu, Y. Zhang, and J. Yue, "Point label meets remote sensing change detection: A consistency-aligned regional growth network," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, Jan. 2024, Art. no. 5603911, doi: 10.1109/tgrs.2023.3348459.

[18] Y. Yan et al., "When vectorization meets change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, Jan. 2023, Art. no. 4401114, doi: 10.1109/tgrs.2023.3347661.

[19] J. Wu, D. Zhu, L. Fang, Y. Deng, and Z. Zhong, "Efficient layer compression without pruning," *IEEE Trans. Image Process.*, vol. 32, pp. 4689–4700, Jan. 2023, doi: 10.1109/tip.2023.3302519.

[20] J. Wang et al., "Individual rubber tree segmentation based on ground-based LiDAR data and faster R-CNN of Deep learning," *Forests*, vol. 10, no. 9, Sep. 2019, Art. no. 793, doi: 10.3390/f10090793.

[21] B. G. Weinstein, S. Marconi, S. Bohlman, A. Zare, and E. White, "Individual tree-crown detection in RGB imagery using semi-supervised deep learning neural networks," *Remote Sens.*, vol. 11, no. 11, Jun. 2019, Art. no. 1309, doi: 10.3390/rs11111309.

[22] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. Int. Conf. Pattern Recognit.*, 2006, pp. 850–855.

[23] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.

[24] X. Chu, A. Zheng, X. Zhang, and J. Sun, "Detection in crowded scenes: One proposal, multiple predictions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12211–12220.

[25] M. P. Ferreira et al., "Individual tree detection and species classification of Amazonian palms using UAV images and deep learning," *Forest Ecol. Manage.*, vol. 475, Nov. 2020, Art. no. 118397.

[26] A. I. Pleşoianu et al., "Individual tree-crown detection and species classification in very high-resolution remote sensing imagery using a deep learning ensemble model," *Remote Sens.*, vol. 12, no. 15, Jul. 2020, Art. no. 2426.

[27] G. T. Miyoshi et al., "A novel deep learning method to identify single tree species in UAV-based hyperspectral images," *Remote Sens.*, vol. 12, no. 8, Apr. 2020, Art. no. 1294.
[28] M. Culman, S. Delalieux, and K. Van Tricht, "Individual palm tree detection using deep learning on RGB imagery to support tree inventory," *Remote Sens.*, vol. 12, no. 21, Oct. 2020, Art. no. 3476, doi: 10.3390/rs12213476.
[29] Y. Sun, Q. Xin, J. Huang, B. Huang, and H. Zhang, "Characterizing tree species of a tropical wetland in southern China at the individual tree level based on convolutional neural network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 11, pp. 4415–4425, Nov. 2019, doi: 10.1109/jstars.2019.2950721.
[30] K. A. Korznikov et al., "Using U-Net-like deep convolutional neural networks for precise tree recognition in very high resolution RGB (red, green, blue) satellite images," *Forests*, vol. 12, no. 1, Jan. 2021, Art. no. 66, doi: 10.3390/f12010066.
[31] X. Chen et al., "Individual tree crown segmentation directly from UAV-borne LiDAR data using the PointNet of deep learning," *Forests*, vol. 12, no. 2, Jan. 2021, Art. no. 131, doi: 10.3390/f12020131.
[32] P. Zamboni et al., "Benchmarking anchor-based and anchor-free state-of-the-art deep learning methods for individual tree detection in RGB high-resolution images," *Remote Sens.*, vol. 13, no. 13, Jun. 2021, Art. no. 2482, doi: 10.3390/rs13132482.
[33] Z. Tang, M. Li, and X. Wang, "Mapping tea plantations from VHR images using OBIA and convolutional neural networks," *Remote Sens.*, vol. 12, no. 18, Sep. 2020, Art. no. 2935, doi: 10.3390/rs12182935.
[34] M. S. Iqbal, H. Ali, S. N. Tran, and T. Iqbal, "Coconut trees detection and segmentation in aerial imagery using mask region-based convolution Neural network," *IET Comput. Vis.*, vol. 15, no. 6, pp. 428–439, Apr. 2021, doi: 10.1049/cvi2.12028.
[35] 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, Feb. 2021, Art. no. 1617, doi: 10.3390/s21051617.
[36] J. R. Braga et al., "Tree crown delineation algorithm based on a convolutional neural network," *Remote Sens.*, vol. 12, no. 8, pp. 1288–1288, Apr. 2020, doi: 10.3390/rs12081288.
[37] N. E. Ocer, G. Kaplan, F. Erdem, D. K. Matci, and U. Avdan, "Tree extraction from multi-scale UAV images using mask R-CNN with FPN," *Remote Sens. Lett.*, vol. 11, no. 9, pp. 847–856, Jun. 2020, doi: 10.1080/2150704x.2020.1784491.
[38] L. Zhang, H. Lin, and F. Wang, "Individual tree detection based on high-resolution RGB images for urban forestry applications," *IEEE Access*, vol. 10, pp. 46589–46598, 2022, doi: 10.1109/access.2022.3171585.
[39] 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," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 106, Feb. 2022, Art. no. 102662, doi: 10.1016/j.jag.2021.102662.
[40] M. Yang et al., "Detecting and mapping tree crowns based on convolutional neural network and Google Earth images," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 108, Apr. 2022, Art. no. 102764, doi: 10.1016/j.jag.2022.102764.
[41] E. Guirado et al., "Mask R-CNN and OBIA fusion improves the segmentation of scattered vegetation in very high-resolution optical sensors," *Sensors*, vol. 21, no. 1, Jan. 2021, Art. no. 320, doi: 10.3390/s21010320.
[42] M. P. Ferreira, R. G. Lotte, F. V. D'Elia, C. Stamatopoulos, D. Kim, and A. R. Benjamin, "Accurate mapping of Brazil nut trees (Bertholletia Excelsa) in Amazonian forests using worldview-3 satellite images and convolutional neural networks," *Ecol. Inform.*, vol. 63, Jul. 2021, Art. no. 101302, doi: 10.1016/j.ecoinf.2021.101302.
[43] H. F. P. Veras, M. P. Ferreira, E. M. Da Cunha Neto, E. O. Figueiredo, A. P. D. Corte, and C. R. Sanquetta, "Fusing multi-season UAS images with convolutional neural networks to map tree species in Amazonian forests," *Ecol. Inform.*, vol. 71, Nov. 2022, Art. no. 101815, doi: 10.1016/j.ecoinf.2022.101815.
[44] J. Mo et al., "Deep learning-based instance segmentation method of Litchi canopy from UAV-acquired images," *Remote Sens.*, vol. 13, no. 19, Sep. 2021, Art. no. 3919, doi: 10.3390/rs13193919.
[45] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[46] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/tpami.2016.2577031.
[47] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020, doi: 10.1109/tpami.2018.2844175.
[48] S. R. Dubey et al., "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing, 503*, pp. 92–108, 2021.
[49] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979, doi: 10.1109/tsmc.1979.4310076.
[50] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," Jun. 17, 2019, Accessed: Jun. 26, 2023. [Online]. Available: https://arxiv.org/abs/1906.07155
[51] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.
[52] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6409–6418.
[53] R. Guo, D. Niu, L. Qu, and Z. Li, "SOTR: Segmenting objects with transformers," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 7137–7146, doi: 10.1109/iccv48922.2021.00707.
[54] H.-S. Fang, J. Sun, R. Wang, M. Gou, Y.-L. Li, and C. Lu, "Instaboost: Boosting instance segmentation via probability map guided copy-pasting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 682–669.

Fangjie Zhu received the B.S. degree in geographic information science, in 2023, from the Sun Yat-sen University, Guangzhou, China, where he is currently working toward the M.S. degree in cartography and geographic information system with the School of Geography and Planning.

His research interests include agricultural remote sensing, change detection, and multimodal data fusion.

Zhenhao Chen is currently working toward the B.S. degree in geographic information science with the School of Geography and Planning, Sun Yat-sen University, Guangzhou, China.

His research interests include agricultural remote sensing, deep learning, and change detection.

Haoyang Li received the B.S. degree in geographic information science, in 2022, from the Sun Yat-sen University, Guangzhou, China, where he is currently working toward the Ph.D. degree in cartography and geographic information system with the School of Geography and Planning.

His research interests include VHR images LULC mapping, agricultural remote sensing, and multimodal data fusion.

Qian Shi (Senior Member, IEEE) received the B.S. degree in sciences and techniques of remote sensing from the Wuhan University, Wuhan, China, in 2010, and the Ph.D. degree in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, in 2015.

She is currently a Professor with the School of Geography and Planning, Sun Yat-sen University, Guangzhou, China. Her research interests include remote sensing image classification, including deep learning, active learning, and transfer learning.

Xiaoping Liu (Member, IEEE) received the B.S. degree in geography and the Ph.D. degree in remote sensing and geographical information sciences from the Sun Yat-sen University, Guangzhou, China, in 2002 and 2008, respectively.

He is currently a Professor with the School of Geography and Planning, Sun Yat-sen University. He has authored 2 books and more than 100 articles. His research interests include image processing, artificial intelligence, and geographical simulation.