Detecting Occluded and Dense Trees in Urban Terrestrial Views with a High-quality Tree Detection Dataset

Yongzhen Wang, Xuefeng Yan, Hexiang Bao, Yiping Chen, *Senior Member*, *IEEE*, Lina Gong, Mingqiang Wei, *Senior Member*, *IEEE*, and Jonathan Li, *Senior Member*, *IEEE*

***Abstract*—Urban trees are often densely planted along the two sides of a street. When observing these trees from a fixed view, they are inevitably occluded with each other and the passing vehicles. The high density and occlusion of urban tree scenes significantly degrade the performance of object detectors. This paper raises an intriguing learning-related question – if a module is developed to enable the network to adaptively cope with occluded and un-occluded regions while enhancing its feature extraction capabilities, can the performance of a cutting-edge detection model be improved? To answer it, a lightweight yet effective object detection network is proposed for discerning occluded and dense urban trees, called OD-UTDNet. The main contribution is a newly-designed Dilated Attention Cross Stage Partial (DACSP) module. DACSP can expand the fields-of-view of OD-UTDNet for paying more attention to the un-occluded region, while enhancing the network's feature extraction ability in the occluded region. This work further explores both the self-calibrated convolution module and GFocal loss, which enhance the OD-UTDNet's ability to resolve the challenging problem of high densities and occlusions. Finally, to facilitate the detection task of urban trees, a high-quality urban tree detection dataset is established, named UTD; to our knowledge, this is the first time. Extensive experiments show clear improvements of the proposed OD-UTDNet over twelve representative object detectors on UTD. The code and dataset are available at https://github.com/yz-wang/OD-UTDNet.**

***Index Terms*—OD-UTDNet, UTD dataset, Urban tree detection, High density and occlusion, Dilated attention cross stage partial module**

I. Introduction

THERE is a proverb in China: "Up above there is heaven; down below there are Suzhou and Hangzhou". It means the two cities have picturesque nature. Besides Suzhou and Hangzhou, many cities in the world are beautiful and attractive in which trees play an important role. The trees in the city are called the lungs of the city. There is a great demand for determining the quantity and monitoring the health status of urban trees. Accordingly, tree detection becomes an essential and fundamental prerequisite for resource appraisal and urban vegetation management [1]–[4].

Conventional tree detection methods are mainly manual and can only be conducted one by one, which is costly and time-consuming. Since then, remote sensing techniques such as aerial photography and LiDAR have been developed for assessing urban trees, both directly and indirectly [5]–[9]. Although these efforts are effective in detecting urban trees, the collection of remote sensing data is difficult and costly to commission. Many works attempt to study image-based urban tree detection in light of the convenient acquisition of RGB images. Lin et al. [10] propose a detection framework for detecting individual trees in unmanned aerial vehicle (UAV) images. Chen et al. [11] develop an improved species-based particle swarm optimization algorithm termed KDT-SPSO for palm tree detection. KDT-SPSO employs a KD-tree structure to accelerate the nearest neighbor search and obtain promising detection results. However, for these conventional wisdom of tree detection, users have to tweak parameters multiple times to obtain satisfied detection results in practical scenarios. This inconvenience heavily discounts the efficiency and user experience.

With advances in deep learning, numerous excellent neuron network-based detection frameworks have emerged in the field of object detection [12]–[16], which bring great opportunities for urban tree detection. Different from the conventional detection methods, learning-based detectors commonly employ CNNs to detect trees directly from the captured images in an end-to-end fashion. Accordingly, these algorithms can produce promising detection results in a variety of scenarios. However, since the majority of urban trees are planted along roads, coupled with heavy occlusion in complex and crowded urban spaces, most existing detectors cannot accurately detect dense urban trees. As demonstrated in Fig. 1, urban trees may be occluded by vehicles, and trees may be occluded by each other. Furthermore, most existing learning-based detectors have numerous parameters and high computational costs, making them unsuitable for deployment on various memory-constrained mobile devices. To this end, this work aims to develop a lightweight yet effective detection network for urban tree detection, especially under occluded and dense scenarios.

In this paper, we respond to an intriguing learning-related question. That is, developing a module to enable the network

Yongzhen Wang, Xuefeng Yan, Hexiang Bao and Lina Gong are with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China (e-mail: wangyz@nuaa.edu.cn, yxf@nuaa.edu.cn, bhxhhxx@nuaa.edu.cn, gonglina@nuaa.edu.cn).

Mingqiang Wei is with the Shenzhen Research Institute, Nanjing University of Aeronautics and Astronautics, Shenzhen, China (e-mail: mingqiang.wei@gmail.com).

Yiping Chen is with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China (chenyiping@xmu.edu.cn).

Jonathan Li is with the Department of Geography and Environmental Management and Department of Systems Design Engineering, University of Waterloo, Waterloo, Canada (e-mail: junli@uwaterloo.ca).

(Corresponding author: Xuefeng Yan.)

(a) RetinaNet (b) CenterNet (c) YOLOv5s (d) Ours

Fig. 1. Detection results by different methods on two typical examples of occluded and dense urban scenes. From (a) to (d): the detection results by (a) RetinaNet [14], (b) CenterNet [17], (c) YOLOv5s [18], and (d) our OD-UTDNet, respectively. Occluded and dense scenes strongly degrade the performance of various object detectors. Even humans have difficulty determining the number of trees in such challenging images. As observed, the proposed OD-UTDNet can discern ten trees from these two samples, while other detectors can only detect up to three trees, which indicates that our method can more robustly detect the trees in such scenarios with higher confidence.

to adaptively cope with occluded regions while enhancing its feature extraction capability will improve the performance of cutting-edge detection models. A novel detection network is proposed for discerning occluded and dense urban trees, called OD-UTDNet. Note that the trees here are mostly street trees and those commonly found in city parks, which are usually mature trees with clear stems. In urban spaces, in addition to complex and heavy traffic, most of the trees planted along the roads are usually quite dense, which creates a severe occlusion problem for tree detection (see Fig. 1). To address this issue, we develop a Dilated Attention Cross Stage Partial (DACSP) module to expand the receptive field of OD-UTDNet and enhance its feature extraction capabilities. In this way, DACSP can make our network pay more attention to the un-occluded areas, thus improving the detection accuracy of the model in occluded cases. Additionally, an effective feature enhancement module (i.e., self-calibrated convolutions) [19] is employed to further enlarge the fields-of-view of each convolutional layer and enrich the output features. Moreover, to cope with the problem of dense detection in this work, the GFocal loss [20] is introduced in the design of OD-UTDNet, which has been proven effective in dense object detection.

One challenge for applying deep learning techniques to urban tree detection is the need for a benchmark dataset. The lack of such datasets is a pervasive problem in the field of urban tree detection due to the expensive data collection and annotation cost. To the best of our knowledge, there is currently no public image dataset related to urban trees. Therefore, we capture and collect 1860 images of urban tree scenes and label them in the format of PASCAL-VOC to establish an urban tree detection dataset, named UTD.

Extensive experiments on the UTD benchmark demonstrate that our network outperforms twelve representative state-of-the-art object detection algorithms. The main contributions of our method are summarized as follows:

- A novel lightweight yet effective detection framework is proposed for discerning occluded and dense urban trees, called OD-UTDNet.
- To address the heavy occlusion problem in urban trees, OD-UTDNet leverages a well-designed DACSP module and self-calibrated convolutions to expand the receptive field of our model and enhance its feature extraction ability. Additionally, the GFocal loss is introduced in the model to cope with the problem of dense tree detection.
- We establish and release an object detection dataset of urban tree images, dubbed UTD. To the best of our knowledge, this is the first image dataset that focuses on urban tree detection.
- The proposed OD-UTDNet is compared with twelve representative state-of-the-art object detection algorithms via extensive experiments. Consistently and substantially, OD-UTDNet performs favorably against them.

The rest of this work is organized as follows. In Section II, the related work is briefly reviewed from three aspects: conventional tree detection techniques, deep learning-based tree detection techniques, and general object detection techniques. Section III describes the details of the proposed OD-UTDNet for occluded and dense tree detection. Section IV first exhibits the established UTD dataset and then presents the conducted experiments and discuss the results, followed by conclusions and future work in Section V.

II. Related Work

This section roughly divides the discussion into three parts: conventional tree detection techniques, deep learning-based tree detection techniques, and general object detection techniques.

A. Conventional Tree Detection Techniques

As a long-standing and fundamental task in both remote sensing and computer vision, tree detection has attracted a great deal of research attention in academia and industry [7], [9], [21]–[23]. Traditional tree detection methods are

commonly based on manual field measurements, which are costly and time-consuming. Afterward, aerial photography and LiDAR techniques are often employed to directly or indirectly assess urban trees [5], [6], [8], [24]. However, given that the collection of remote sensing data is difficult and expensive, many efforts have begun to study image-based urban tree detection, particularly with RGB images. Srestasathiern et al. [25] propose a palm tree detection approach based on high-resolution satellite images and achieve a detection rate of about 90%. Jiang et al. [26] propose a GPU-accelerated scale-space filtering (SSF) algorithm to detect the papaya trees with UAV images. SSF shows a clear improvement over other algorithms in both speed and accuracy. Donmez et al. [7] employ a Connected Components Labeling (CCL) algorithm to detect the citrus trees based on the high-resolution UAV images and achieve a high accuracy rate.

B. Tree Detection via Deep Learning

Advances in deep learning bring a big opportunity for automatic tree detection. Iqbal et al. [27] present a deep learning approach for the detection and segmentation of coconut trees in aerial imagery. They employ a Mask R-CNN model with a ResNet50/ResNet101-based architecture and achieve an overall 91% mean average precision for coconut tree detection. Hartling et al. [28] conduct extensive experiments and reveal that DenseNet is more effective for urban tree classification and detection, outperforming the popular RF and SVM techniques. Dong et al. [29] develop a single-tree detection algorithm for high-resolution remote sensing images based on a cascade neural network. They design a classifier with a back propagation (BP) neural network and analyze the differences between tree and non-tree samples. Liu et al. [30] propose a point-based neural network named LayerNet for tree species classification in forest regions. LayerNet can divide multiple overlapping layers in Euclidean space to extract the local structural features of the tree. Briechle et al. [31] propose a dual-CNN-based network called Silvi-Net for the classification of 3D trees. Experimental results demonstrate that Silvi-Net outperforms other state-of-the-art 3D tree detection approaches, especially in the case of small samples. Ferreira et al. [32] develop a fully convolutional neural network model for individual tree detection and species classification in Amazonian regions. They adopt a low-cost UAV to capture RGB images and then detect trees in these images. Xie et al. [33] present an end-to-end trainable framework for street tree detection based on the Fast R-CNN network. Experimental results show a clear improvement of their method over other object detectors.

C. Object Detection

Object detection aims at predicting both the class and bounding box of the target object, which is an important research area in computer vision [34], [35]. Recently, with the rapid development of CNNs, learning-based algorithms have dominated modern object detection for years.

Current object detection techniques can roughly fall into two categories, i.e., one-stage and two-stage. For two-stage approaches, they first adopt the region proposal methods [36] to produce a sparse set of candidate proposals and then refine their locations and predict their specific categories. The most representative two-stage detector is R-CNN [35], which is the first successful attempt to replace the classifier with a CNN, achieving large improvements in detection accuracy. Since then, numerous variants based on this framework have been developed, including Fast R-CNN [37], Faster R-CNN [38], Cascade R-CNN [39] and Grid R-CNN [40]. Despite achieving remarkable detection accuracy, two-stage detectors are not satisfactory in terms of inference speed. Accordingly, to achieve a better speed-performance trade-off, various one-stage detectors are proposed for real-time detection. Representative algorithms including YOLO series [13], [18], [41]–[43], RetinaNet [14], CenterNet [17], SSD [12], etc. In general, the speed of the one-stage detector is relatively faster, but the detection performance is slightly weaker than that of the two-stage detector.

III. OD-UTDNET

Recall a time when you walk in a street. You may find that the urban trees planted along the two sides of the street are dense and occluded with each other and the passing vehicles. Thus, you can just observe a part of these trees from your view, potentially leading to misunderstanding these trees. Similarly, the high density and occlusion of urban tree scenes significantly degrade the performance of cutting-edge object detectors, since these detectors can only perceive a part of the trees and pay much attention to the occluded region.

Is it possible to make a detector pay more attention to the un-occluded region while enhancing its feature extraction ability in the occluded region? If the answer is positive, the performance of cutting-edge detection models for handling the high density and occlusion problems can be improved. It is the focus of this work.

In this section, we first describe the overall architecture of the urban tree detection network, namely, OD-UTDNet. After that, the proposed Dilated Attention Cross Stage Partial module (DACSP) is elaborated to demonstrate how we address the heavy occlusion problem in the task of urban tree detection. Finally, the self-calibrated convolutions and GFocal loss are introduced to optimize OD-UTDNet to further enhance the detection accuracy in dense tree scenes.

A. Overview of OD-UTDNet

For urban tree detection, the first consideration is how to address the problem of heavy occlusion in dense urban spaces, which greatly degenerates the detection accuracy of existing detectors.

As known, YOLO series (e.g., the latest version YOLOv5) [13], [18], [41]–[43] have been successfully applied in object detection. Although YOLOv5 has achieved promising results in various object detection benchmarks (e.g., MSCOCO [44], PASCAL-VOC [45]), there are still many challenging yet unsolved problems. First, the YOLOv5 family is originally designed for object detection in general yet easy scenarios, without considering how to cope with the non-trivial dense object detection scenes. Second, like most existing detectors,

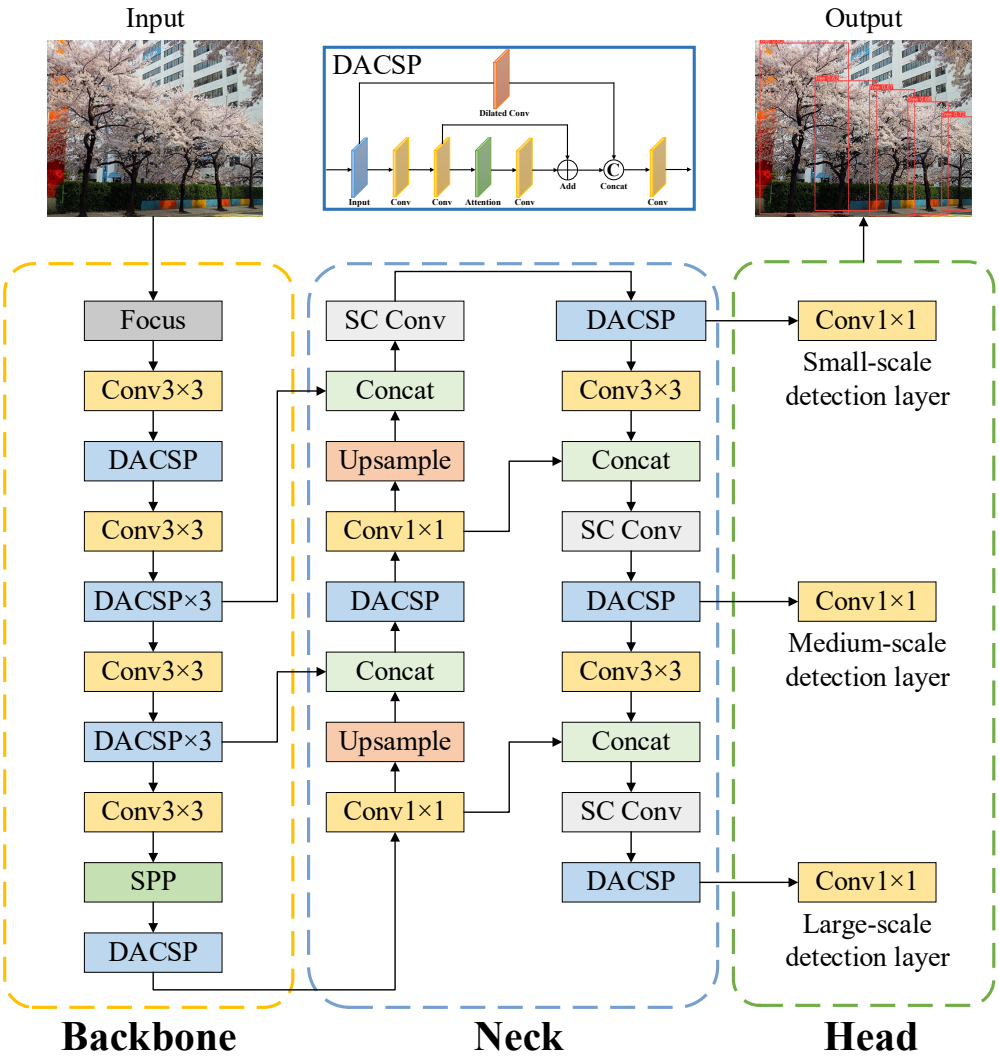


Fig. 2. The overall architecture of OD-UTDNet. It mainly consists of three parts: the Backbone network, the Neck module, and the Detection head. Given an urban tree image, we first employ the Backbone network to extract the complex and deep features from the input image. After that, the neck module is adopted to further fuse the multi-scale features and transmit them to the detection head to predict the final detection result. DACSP refers to Dilated Attention Cross Stage Partial Network, and SC Conv refers to self-calibrated convolutions. The main contribution is a newly-designed DACSP module. DACSP expands the fields-of-view of the network for paying more attention to the un-occluded region, and enhances the network's feature extraction ability in the occluded region.

the YOLOv5 family is susceptible to the occlusion problems in the urban tree detection task, resulting in a significant decrease in detection accuracy. Third, YOLOv5s is lightweight and efficient, which is promising for resource-limited mobile devices in the YOLOv5 family. But its detection accuracy is thus largely degenerated for the high density and occluded urban tree scenes.

To this end, we start from these three aspects and propose an enhanced object detection network based on YOLOv5s for occluded and dense urban tree detection. Here, the proposed OD-UTDNet is built on top of the latest version of the YOLO series, namely, YOLOv5s [18] (the smallest version of YOLOv5). Actually, our OD-UTDNet has the potential to benefit from a more complex version of YOLOv5 to further improve its performance, such as YOLOv5m and YOLOv5l. However, we choose the smallest version because a lightweight model is more desirable for deploying automatic tree detection on many memory-constrained devices.

As illustrated in Fig. 2, the proposed network consists of three main components, i.e., the backbone network, the neck module, and the detection head. Given an input image, we first leverage the Focus operation in the backbone network to divide the image into different granularities and aggregate them together. Then, several enhanced Cross Stage Partial (CSP) modules [46], i.e., Dilated Attention CSP (DACSP), are employed to extract the complex and deep features from the restructured image. DACSP is a novel feature enhancement module developed to expand the receptive field of our model, thus making the network pay more attention to the un-occluded areas to reduce the impact of heavy occlusions. After that, the neck module is adopted to produce the feature pyramids based on the Path Aggregation Network (PANet) [47] and then transmits the feature maps to the detection head. Finally, the detection head module is employed to generate the final class

probability score, bounding boxes, and confidence score.

Moreover, to further boost the detection performance of OD-UTDNet and address the dense detection problem in urban spaces, we introduce a more recent multi-scale feature enhancement module (self-calibrated convolutions) and an up-to-date GFocal loss into our network. Both of them have been widely used in object detection tasks and have been demonstrated to be effective in improving detection accuracy, which will be described in the following subsections.

B. Dilated Attention Cross Stage Partial network

Theoretically, the feature extraction ability of the network directly determines the performance of the model. Thus, we argue that there are two solutions to reduce the impact of occlusions on urban tree detection. One is to expand the receptive field of the network to help the model detect the trees from the un-occluded areas. The other is to enhance the feature extraction ability of the network so that trees can be detected directly from the occluded areas. To improve the feature extraction capability of the network, natural thinking is to deepen the number of network layers or employ more complex network architectures (e.g., GNNs [48] and Transformers [49]). However, a lightweight model is more desirable to deploy for automatic tree detection tasks due to the limited hardware and computing ability of intelligent detection systems (e.g., intelligent vehicles, UAVs). Given this, we consider employing a dilated convolution module (expanding the receptive field) and a lightweight attention network (enhancing feature extraction ability) in the design of the CSP network to achieve a better parameter-performance trade-off. This enhanced CSP network is called Dilated Attention CSP (DACSP), and its architecture is depicted in Fig. 3.

Dilated convolutions can expand the receptive field of the network without increasing the computational effort, which has been demonstrated to be effective in improving the performance of networks. However, due to the special calculation method of dilated convolution, the continuous information of the image will be destroyed, resulting in the loss of partial features. To address such a problem, we combine dilated convolutions with conventional convolutions to expand the receptive field of the network while ensuring that the image information will not be lost. Specifically, we first employ the dilated convolution module and the conventional convolution network to calculate the input feature maps and then concatenate them through a residual connection to extract multi-scale features while expanding the receptive field of the CSP network (see Fig. 3). In this way, our model can focus on more un-occluded areas, thus reducing the impact of occlusions on detection accuracy.

Attention mechanisms have been widely used in improving the performance of neural networks [50], [51]. We consider adopting a lightweight attention module to boost the feature extraction capability of the CSP network, thus enhancing the detection accuracy of OD-UTDNet. Motivated by Qin et al. [52], an up-to-date Frequency Channel Attention Network (FCANet) is employed in the design of the CSP network to improve the detection performance of our model, as exhibited in Fig. 3. FCANet is extended on the basis of SENet [51] and combined with Discrete Cosine Transform (DCT) to develop a novel multi-spectral channel attention mechanism. It enables the CSP network to learn the weights from different features adaptively, thus enhancing the detection performance of OD-UTDNet.

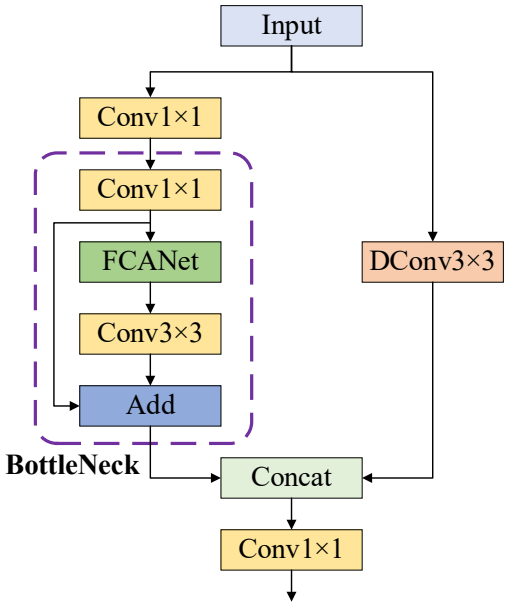


Fig. 3. The architecture of Dilated Attention CSP network. In our designation, we employ a dilated convolution module to expand the receptive field of the convolutional layer and a lightweight attention network to enhance the feature extraction ability of the CSP network. Both of them can help the model address severe occlusions in urban spaces and improve the detection accuracy of OD-UTDNet. FCANet represents the frequency channel attention network. DConv3×3 refers to a dilated convolution network with a kernel size of 3 (dilation rate = 3).

C. Self-Calibrated Convolutions

The self-calibrated convolution network [19] is an enhanced CNN structure, which is employed to build long-range spatial and inter-channel dependencies around each spatial location. That is, it can enlarge the receptive field of each convolutional layer and improve the feature extraction ability of CNNs. Therefore, we consider employing the self-calibrated convolution network as a feature enhancement module to help OD-UTDNet address the aforementioned occlusion problem and boost the detection accuracy of the model.

The architecture of self-calibrated convolutions is demonstrated in Fig. 4. Given an input feature map X with channel C, we first split it into two feature maps X_1 and X_2 with channel $C/2$. Then, X_1 is sent to the self-calibrated branch for feature transformation and fusion. In this branch, three filters (i.e., K_2, K_3, and K_4) are adopted to extract and fuse multi-scale features from X_1. Next, the filter K_1 is employed to transmit and extract features from X_2 to obtain the other half of result Y_2. Finally, we concatenate Y_1 and Y_2 to produce the final output Y. In our designation, the self-calibrated convolutions are introduced into the neck module of OD-UTDNet to expand the fields-of-view of the convolutional layer and extract multi-scale features, thus enforcing the network to pay more

attention to the un-occluded areas. In this way, our model can well address the unpredictable occlusion scenarios in urban tree detection tasks.

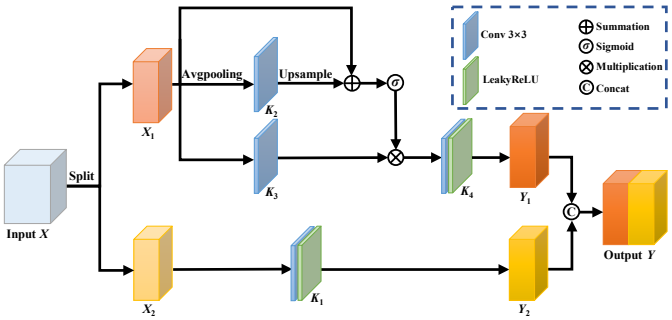


Fig. 4. The architecture of self-calibrated convolution network. It can build long-range spatial and inter-channel dependencies around each spatial location, thus expanding the receptive field of each convolutional layer and enhancing the feature extraction ability of CNNs.

D. Generalized Focal Loss

GFocal loss [20] is an improved version of Focal loss [14], which aims to solve the problem of imbalance between positive and negative samples in dense object detection tasks. In the training process, Focal loss reduces the weight of numerous simple negative samples, making the network pay more attention to difficult samples (i.e., dense objects), thus improving the detection accuracy of dense objects. Therefore, natural thinking is to employ Focal loss in the design of our framework to cope with dense scenes in urban tree detection tasks. However, Focal loss itself has many limitations; for example, it can only handle discrete labels such as 0 or 1 but cannot deal with continuous labels between 0 and 1. In addition, the inconsistency of the classification and regression calculation methods during training and inference will degenerate the detection accuracy of the model. To this end, Li et al. [20] develop an enhanced Focal loss (GFocal loss) to address the above problems and further improve the accuracy of dense object detection tasks. Gfocal loss mainly improves Focal loss from two aspects: one is to incorporate the quality estimation into the class prediction vector to eliminate the risk of inconsistency between the training and inference phases, and the other is to leverage a vector instead of the Dirac delta distribution to accurately depict the flexible distribution in real data.

GFocal loss consists of Quality Focal loss (QFL) and Distribution Focal loss (DFL). To address the above-mentioned inconsistency problem between the training and test phases, QFL adopts a joint representation of localization quality (IoU score) and classification score, where the standard one-hot category label $y \in \{0, 1\}$ turns to a possible float target $y \in [0, 1]$ on the corresponding category. Note that $y = 0$ represents the negative samples with quality score being equal to 0, and $0 < y \leq 1$ refers to the positive samples with the target IoU score y. Additionally, QFL employs the IoU score between the estimated bounding box and its corresponding ground-truth label to represent the localization quality label y, which takes a value between 0 and 1. However, considering that the original Focal loss only supports $\{0, 1\}$ discrete labels, but the new joint representation labels contain decimals. QFL loss extends the Focal loss for enabling the successful training under the new continuous labels as:

$$\mathbf{QFL}(\sigma) = -|y - \sigma|^{\beta}((1 - y)\log(1 - \sigma) + y\log(\sigma)), \quad (1)$$

where $-((1 - y)\log(1 - \sigma) + y\log(\sigma))$ denotes the complete cross-entropy loss function, $y \in [0, 1]$ denotes the new quality label and $\sigma \in (0, 1)$ represents the predicted result. Note that the sigmoid operators $\sigma(\cdot)$ is employed for multi-class implementation, with the output being denoted as σ for simplicity. Similar to the Focal loss, the term $|y - \sigma|^{\beta}$ is employed to adjust the weights for positive/negative samples during the training phase. As the quality estimation becomes accurate ($\sigma \rightarrow y$), the modulating factor goes to 0 and the weight for well-estimated examples is reduced, where the parameter β is employed to control the weighting rate (β = 2 works best in our experiments). QFL still retains the classification vector, but the physical meaning of the corresponding category position confidence is no longer a classification score but a quality prediction score. Furthermore, to accurately describe the distribution in real data, DFL is employed to accelerate the network's training.

DFL first converts the integral over the continuous domain into a discrete representation, i.e., $\{y_0, y_1, \ldots, y_i, \ldots, y_n\}$. Therefore, given the discrete distribution property $\sum_{i=0}^{n} P(y_i)$=1, the predicted regression value $\hat{y}$ ($y_0 \leq \hat{y} \leq y_n$) is formulated as:

$$\hat{y} = \sum_{i=0}^{n} P(y_i)\, y_i. \quad (2)$$

Hence, the General distribution $P(x)$ can be directly implemented through a $Softmax$ $\mathcal{S}(\cdot)$ layer with $n + 1$ units. For simplicity, the output of $P(y_i)$ is marked as $\mathcal{S}_i$. In this way, $\hat{y}$ can be trained in an end-to-end manner with existing loss objectives like IoU loss [53]. Furthermore, considering that the most appropriate underlying location would not be far away from the coarse label, the DFL is employed to help the network rapidly focus on the values near label y, by enlarging the probabilities of y_i and y_{i+1} ($y_i \leq y \leq y_{i+1}$). Consequently, DFL is extended on the basis of the complete cross-entropy part in QFL as:

$$\begin{aligned}\mathbf{DFL}(\mathcal{S}_i, \mathcal{S}_{i+1}) = -((y_{i+1} - y)\log(\mathcal{S}_i) \\ +(y - y_i)\log(\mathcal{S}_{i+1})).\end{aligned} \quad (3)$$

Intuitively, DFL is designed to focus on enlarging the probabilities of the values around y (i.e., y_i and y_{i+1}), thus can improve the training efficiency. Finally, QFL and DFL can be unified to form the GFocal loss. Assume that a network estimates probabilities for two variables y_l, y_r ($y_l < y_r$) as p_{y_l}, p_{y_r} ($p_{y_l} \geq 0, p_{y_r} \geq 0, p_{y_l} + p_{y_r} = 1$), with a final estimation of their liner combination being $\hat{y} = y_l p_{y_l} + y_r p_{y_r}$ ($y_l \leq \hat{y} \leq y_r$). Note that the corresponding continuous label y for the prediction $\hat{y}$ also satisfies $y_l \leq y \leq y_r$. Taking

the absolute distance $|y-\hat{y}|^{\beta}$ as modulating factor, GFL is formulated as:

$$\begin{aligned}\mathbf{GFL}\left(p_{y_l}, p_{y_r}\right) = -|y-(y_l p_{y_l}+y_r p_{y_r})|^{\beta}\left((y_r-y)\log\left(p_{y_l}\right)\right. \\ \left.+(y-y_l)\log\left(p_{y_r}\right)\right).\end{aligned} \tag{4}$$

Through extensive experiments, we find that DFL provides our model with relatively limited performance gains while adding additional computational overhead. Hence, only QFL is adopted in the design of OD-UTDNet.

IV. Experiments and Discussion

In this section, we evaluate OD-UTDNet on the proposed UTD benchmark and conduct comparisons with SSD300 [12], RetinaNet [14] YOLOv3 [43], Centernet [17], Grid R-CNN [40], Cascade R-CNN [39], YOLOv4 [42], YOLOF [54], Sparse R-CNN [55], and YOLOv5 [18]. After that, a comprehensive ablation study is implemented to analyze the effectiveness of each component designed in our OD-UTDNet. We also describe the details of the established UTD dataset and the specific training settings. The details are as follows.

A. Dataset

Since there is no publicly available urban tree image dataset, to train and evaluate the proposed network, a new object detection dataset of urban trees is established, called UTD. To the best of our knowledge, UTD is the first image dataset that focuses on urban tree detection research. Specifically, our UTD contains a total of 1860 tree images captured in various urban scenes. These images are collected from the Internet or captured by us. Considering that an accurately labeled dataset is essential for object detection tasks, we employ the Colabeler tool [56] to manually annotate our UTD in the format of PASCAL-VOC. Moreover, to further ensure the accuracy of the labels, experts from Nanjing Forestry University are invited to proofread all the annotated images. The task of annotating is difficult due to the heavy occlusion of trees in some images, and it takes about 5-10 minutes to label each image. After labeling these 1860 images, we obtain 5540 ground truth bounding boxes.

To train our OD-UTDNet, the dataset is divided into three groups, i.e., the training set, the validation set, and the test set, as exhibited in Table I. Additionally, to demonstrate our dataset more intuitively, four image examples from the UTD dataset are exhibited in Fig. 5, each of them containing more than three urban trees. Although the proposed UTD contains various categories of urban trees, all the different categories of the tree are cast as one class here, since we mainly focus on the tree detection task in this work. In the future, we will further classify these detected trees in subsequent work. Furthermore, The established UTD will be released on our GitHub website to accelerate the research on automatic tree detection tasks.

B. Implementation Details

Training details. The proposed OD-UTDNet is implemented using PyTorch 1.9 on a system with an Intel(R) Core(TM) i7-9700 CPU, 16 GB RAM, and an Nvidia GeForce RTX 3090 GPU. To optimize our model, we employ the SGD optimizer with a mini-batch size of 32, where the momentum parameter and weight decay are set to 0.937 and 0.0005, respectively. The total number of epochs and the initial learning rate are set to 100 and 0.01, respectively. Considering that training the network on pretrained weights can accelerate the convergence speed of the model, our OD-UTDNet is trained on the UTD with the MSCOCO [44] pretrained weights.

TABLE I
Detailed information about UTD dataset.

Group	Training	Validation	Test
Number of images	1530	170	160
Number of tress	4643	516	381

Fig. 5. Examples of different scenes in the UTD dataset, where the red boxes refer to the ground truths.

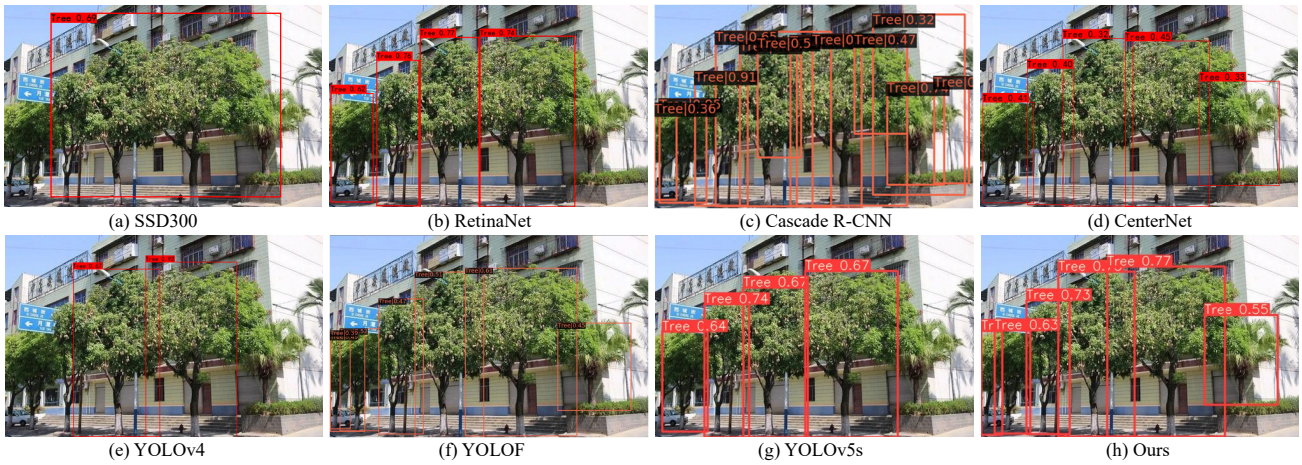
Evaluation Settings. To quantitatively evaluate the performance of OD-UTDNet, the Average Precision (AP) and AP_{50} are employed as the evaluation metrics, which are the most widely used evaluation indexes in object detection tasks. The proposed OD-UTDNet is compared with several state-of-the-art object detection methods. These detectors can be classified into two categories: 1) two-stage-based Grid R-CNN [40] and Cascade R-CNN [39]; and 2) one-stage-based SSD300 [12], RetinaNet [14], YOLOv3 [43], Centernet [17], YOLOv4 [42], YOLOF [54] and YOLOv5 [18]. Moreover, we also compare OD-UTDNet with a recent sparse-based object detection framework Sparse R-CNN [55].

C. Comparison with State-of-the-arts

We report the AP, AP_{50}, and Frames Per Second (FPS) metrics of twelve state-of-the-art detection algorithms (including three versions of YOLOv5) on the established UTD test set in Table II. To make a fair comparison, all compared approaches are retrained on the UTD dataset, following the settings in their papers. As observed, the proposed OD-UTDNet achieves the best performance with 37.7 AP and 78.5 AP_{50} compared to SOTA approaches. OD-UTDNet realizes an excellent parameter-performance trade-off, since its parameter

TABLE II
COMPARISON OF OD-UTDNET WITH STATE-OF-THE-ART OBJECT DETECTORS ON THE UTD TEST SET. WE EVALUATE THE RUN TIME WITH BATCH SIZE 1 ON A SINGLE NVIDIA GEFORCE RTX 3090 GPU.

Method	Publication	Params	GFlops	AP	AP_{50}	FPS
SSD300	ECCV'16	34.3M	386.3	31.3	71.2	45
RetinaNet	ICCV'17	33.7M	239.3	29.7	70.7	22
YOLOv3	arXiv'18	61.5M	155.1	28.0	68.8	63
Grid R-CNN	CVPR'19	64.2M	307.3	29.4	69.5	8
Cascade R-CNN	TPAMI'19	68.9M	195.8	35.8	76.7	19
CenterNet	CVPR'19	14.2M	51.3	25.5	77.3	65
YOLOv4	arXiv'20	63.9M	128.4	30.9	71.8	41
YOLOF	CVPR'21	44.0M	86.0	35.1	75.4	33
Sparse R-CNN	CVPR'21	105.9M	149.9	31.4	71.4	21
YOLOv5s	[18]	7.3M	16.3	30.8	71.1	**67**
YOLOv5m	[18]	21.4M	50.3	33.6	74.8	60
YOLOv5l	[18]	47.0M	114.1	34.3	75.1	47
OD-UTDNet	ours	8.7M	19.6	**37.7**	**78.5**	62



(a) SSD300 (b) RetinaNet (c) Cascade R-CNN (d) CenterNet

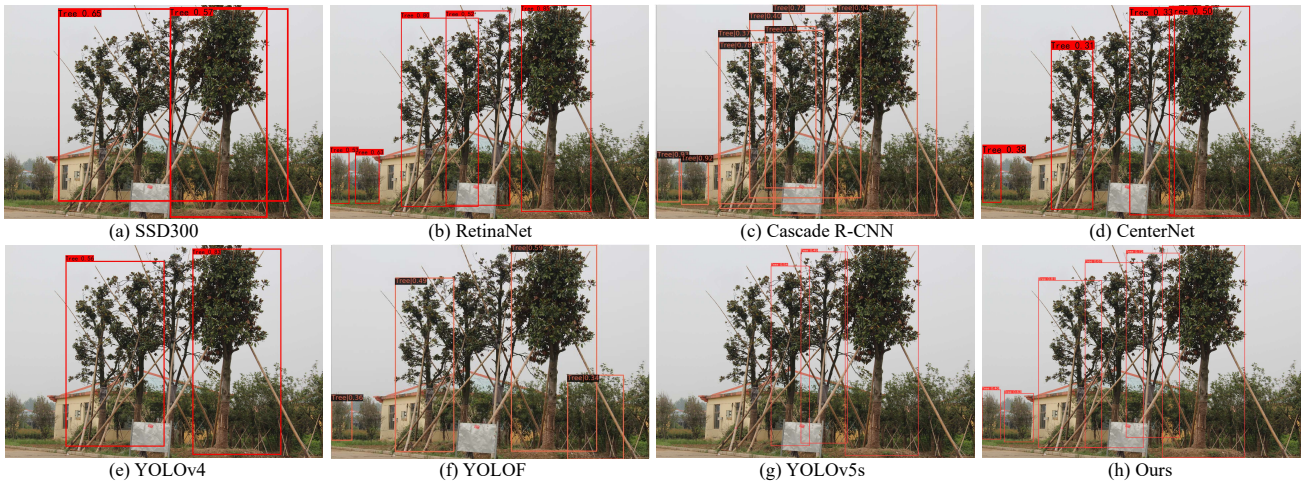
(e) YOLOv4 (f) YOLOF (g) YOLOv5s (h) Ours

Fig. 6. Detection results by different methods in a heavy occlusion case. From (a) to (h): the detection results by (a) SSD300 [12], (b) RetinaNet [14], (c) Cascade R-CNN [39], (d) CenterNet [17], (e) YOLOv4 [42], (f) YOLOF [54], (g) YOLOv5s [18], and (h) our OD-UTDNet, respectively. As can be seen, there are a total of six trees in the image, and only three methods, namely, Cascade R-CNN, YOLOF, and OD-UTDNet can detect all the trees. However, the Cascade R-CNN and YOLOF suffer from the mis-detection problem as they discern these six trees as eleven and seven trees, respectively. In contrast, our OD-UTDNet can effectively address unpredictable occlusion scenarios.

amount and calculation cost are acceptable and rank second among the compared methods. In addition, the inference speed of OD-UTDNet is fast, and it can detect 62 images per second, which is suitable for deploying on memory-constrained devices.

To further evaluate the effectiveness of OD-UTDNet under occluded and dense scenes, we test our method and other approaches in two heavy occlusion cases. Visual comparisons of the detection results are depicted in Fig. 6 and Fig. 7. As observed, the captured images contain severe occlusion, where trees and trees occluded each other to form a whole. It is practically impossible for a human to count the quantity of trees in these images. As demonstrated in Fig. 6 and Fig. 7, most compared detection algorithms miss some trees due to heavy occlusion. For Cascade R-CNN and YOLOF, although they have overcome the impact of occlusion to some extent, the confidence of the detected trees is quite low. In addition, the Cascade R-CNN method suffers from significant mis-detection, and it can easily identify other objects as trees. In contrast, the proposed DASCP module enhances the feature extraction capability of OD-UTDNet while expanding the receptive field of the model, thus reducing the impact of occlusions on detection accuracy. Consequently, our OD-UTDNet exhibits better results in detecting occluded and dense trees.

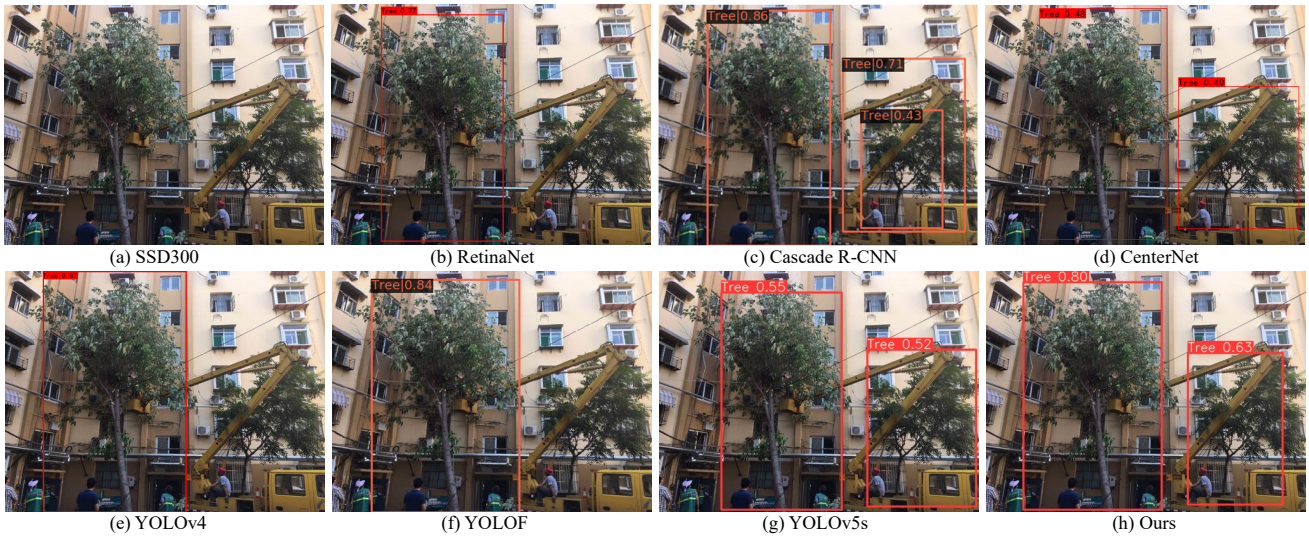
Likewise, to verify the generalization ability of OD-UTDNet in general scenes (without severe occlusion), we present a visual comparison of the detection results in the cases of few occlusions, as exhibited in Fig. 8 and Fig. 9. It can be observed that even in the case of only a small amount of occlusion, many detection algorithms still miss some trees. Similar to the results in Fig. 6 and Fig. 7, the detection results produced by Cascade R-CNN still have the problem of mis-detection, and one tree is detected as multiple trees. Compared with these SOTA



(a) SSD300 (b) RetinaNet (c) Cascade R-CNN (d) CenterNet

(e) YOLOv4 (f) YOLOF (g) YOLOv5s (h) Ours

Fig. 7. Detection results by different detectors in a heavy occlusion case. Only the Cascade R-CNN and our OD-UTDNet can detect all the trees in the image, while other detectors have more or less missed trees. Similar to the results in Fig. 6, the Cascade R-CNN still has the problem of mis-detection. Comparatively, our method can handle occlusion cases well.



(a) SSD300 (b) RetinaNet (c) Cascade R-CNN (d) CenterNet

(e) YOLOv4 (f) YOLOF (g) YOLOv5s (h) Ours

Fig. 8. Detection results by different approaches in a case of few occlusions. The results of SSD300, RetinaNet, YOLOv4, and YOLOF obviously miss some trees. For CenterNet and YOLOv5s, although they can discern the two trees in this image, the confidence of the detected trees is quite low. Similar to the previous cases, the Cascade R-CNN identifies these two trees as three. In contrast, our OD-UTDNet can generalize well in both heavy occlusion and few occlusion situations.

detectors, our OD-UTDNet can detect more trees with higher confidence, which demonstrates that our model performs well in both heavy occlusion and few occlusion situations.

D. Ablation Study

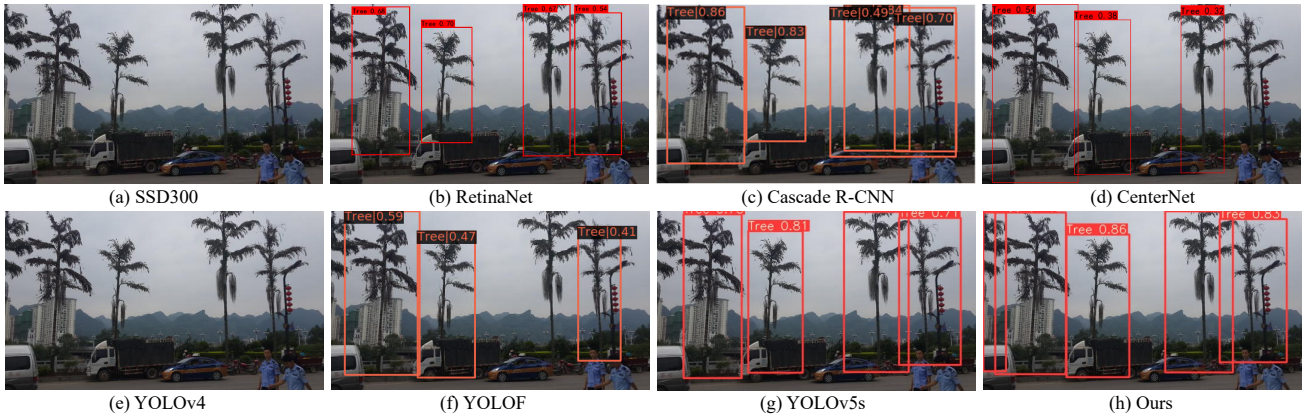
OD-UTDNet exhibits superior detection performance compared to twelve representative state-of-the-art methods. To further validate the effectiveness of the proposed OD-UTDNet, comprehensive ablation studies are conducted to analyze different components, including the Dilated Attention CSP module, self-calibrated convolutions, and GFocal loss.

We first construct the base model with the original YOLOv5s as the baseline of the detection network and then train this model through the implementation details mentioned above. Subsequently, different modules are incrementally added into the base model as:

1) base model + Dilated Attention CSP module $\rightarrow V_1$,
2) V_1 + self-calibrated convolutions $\rightarrow V_2$,
3) V_2 + GFocal loss $\rightarrow V_3$ (our full model).

All these models are retrained in the same way as before and tested on the UTD test set. The performances of these variants are exhibited in Table III.

As observed, each module in our OD-UTDNet contributes to object detection, especially the well-designed Dilated Attention CSP network, which achieves a 4.2 AP and 4.3 AP_{50} improvement over our base model. The introduction of self-calibrated convolutions and GFocal loss also greatly improve the performance of the network. Finally, it can be found that a committee with all three modules produces the highest detection performance, which indicates that the three modules complement each other.



(a) SSD300 (b) RetinaNet (c) Cascade R-CNN (d) CenterNet

(e) YOLOv4 (f) YOLOF (g) YOLOv5s (h) Ours

Fig. 9. Detection results by different methods in a case of few occlusions. It can be observed that only our OD-UTDNet can detect all five trees in this figure, while none of the compared algorithms can detect the far-left tree. For SSD300 and YOLOv4, they cannot even detect one tree in the image. Once again, the Cascade R-CNN detector suffers from mis-detection problems.

TABLE III
ABLATION STUDY ON OD-UTDNET. AS OBSERVED, OUR FULL MODEL (V_3) OUTPERFORMS OTHER ALTERNATIVES.

Variants	Base	V_1	V_2	V_3
DACSP	w/o	✓	✓	✓
SC Conv	w/o	w/o	✓	✓
GFocal loss	w/o	w/o	w/o	✓
AP	30.8	35.0	36.6	**37.7**
AP_{50}	71.1	75.4	77.6	**78.5**

V. CONCLUSIONS AND FUTURE WORK

In this work, for the first time, we have established a high-quality urban tree image dataset to facilitate research on the automatic detection of trees. Accordingly, a lightweight yet effective detection framework is proposed, called OD-UTDNet, for automatically detecting occluded and dense urban trees. To cope with the heavy occlusion in tree detection tasks, a Dilated Attention Cross Stage Partial (DACSP) module is developed to expand the receptive field of the model while enhancing its feature extraction capability. Additionally, the self-calibrated convolution network is introduced to further enlarge the fields-of-view of each convolutional layer and enrich the output features, thus reducing the impact of occlusion on tree detection. Moreover, the GFocal loss is introduced to address dense scenes in urban tree detection. Finally, extensive evaluations demonstrate that our OD-UTDNet performs favorably against twelve representative state-of-the-art algorithms.

In the future work, we plan to employ other powerful network architectures, such as GCNs or Transformers, to develop a more effective urban tree detection framework. In addition, we will further label the trees in UTD according to their species and conduct a classification study.

Acknowledgements. The authors thank the anonymous reviewers for their careful reading and valuable comments. This work was supported in part by the National Natural Science Foundation of China (No. 62172218), the Joint Fund of National Natural Science Foundation of China and Civil Aviation Administration of China (No. U2033202), the 14th Five-Year Planning Equipment Pre-Research Program (No. JCKY2020605C003), and the Free Exploration of Basic Research Project, Local Science and Technology Development Fund Guided by the Central Government of China (No. 2021Szvup060).

REFERENCES

[1] L. Wallace, A. Lucieer, and C. S. Watson, "Evaluating tree detection and segmentation routines on very high resolution UAV lidar data," *IEEE Trans. Geosci. Remote. Sens.*, vol. 52, no. 12, pp. 7619–7628, Dec. 2014.

[2] A. Fekete and M. Cserép, "Tree segmentation and change detection of large urban areas based on airborne lidar," *Comput. Geosci.*, vol. 156, p. 104900, Nov. 2021.

[3] D. Chi, J. Degerickx, K. Yu, and B. Somers, "Urban tree health classification across tree species by combining airborne laser scanning and imaging spectroscopy," *Remote. Sens.*, vol. 12, no. 15, p. 2435, Jul. 2020.

[4] A. Harikumar, F. Bovolo, and L. Bruzzone, "A local projection-based approach to individual tree detection and 3-d crown delineation in multistoried coniferous forests using high-density airborne lidar data," *IEEE Trans. Geosci. Remote. Sens.*, vol. 57, no. 2, pp. 1168–1182, Feb. 2019.

[5] M. Hirschmugl, M. Ofner, J. Raggam, and M. Schardt, "Single tree detection in very high resolution remote sensing data," *Remote. Sens. Environ.*, vol. 110, no. 4, pp. 533–544, Oct. 2007.

[6] W. Yao and Y. Wei, "Detection of 3-d individual trees in urban areas by combining airborne lidar data and imagery," *IEEE Geosci. Remote. Sens. Lett.*, vol. 10, no. 6, pp. 1355–1359, Nov. 2013.

[7] C. Donmez, O. Villi, S. Berberoglu, and A. Cilek, "Computer vision-based citrus tree detection in a cultivated environment using UAV imagery," *Comput. Electron. Agric.*, vol. 187, p. 106273, Aug. 2021.

[8] E. M. da Cunha Neto, F. E. Rex, H. F. P. Veras, M. M. Moura, C. R. Sanquetta, P. S. Käfer, M. N. I. Sanquetta, A. M. A. Zambrano, E. N. Broadbent, and A. P. D. Corte, "Using high-density uav-lidar for deriving tree height of araucaria angustifolia in an urban atlantic rain forest," *Urban. For. Urban. Gree.*, vol. 63, p. 127197, Aug. 2021.

[9] E. G. Parmehr, M. Amati, E. J. Taylor, and S. J. Livesley, "Estimation of urban tree canopy cover using random point sampling and remote sensing methods," *Urban. For. Urban. Gree.*, vol. 20, pp. 160–171, Dec. 2016.

[10] Y. Lin, M. Jiang, Y. Yao, L. Zhang, and J. Lin, "Use of uav oblique imaging for the detection of individual trees in residential environments," *Urban. For. Urban. Gree.*, vol. 14, no. 2, pp. 404–412, Jun. 2015.

[11] Z. Y. Chen, I. Y. Liao, and A. Ahmed, "Kdt-spso: A multimodal particle swarm optimisation algorithm based on k-d trees for palm tree detection," *Appl. Soft Comput.*, vol. 103, p. 107156, May. 2021.

[12] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, Oct. 2016, pp. 21–37.

[13] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[15] D. Liang, Q. Geng, Z. Wei, D. A. Vorontsov, E. L. Kim, M. Wei, and H. Zhou, "Anchor retouching via model interaction for robust object detection in aerial images," *IEEE Trans. Geosci. Remote. Sens.*, vol. 60, pp. 1–13, Mar. 2022.

[16] M. Li, X. Zhao, J. Li, and L. Nan, "Comnet: Combinational neural network for object detection in uav-borne thermal images," *IEEE Trans. Geosci. Remote. Sens.*, vol. 59, no. 8, pp. 6662–6673, Aug. 2021.

[17] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv:1904.07850*, Apr. 2019. [Online]. Available: https://arxiv.org/abs/1904.07850

[18] G. Jocher, A. Stoken, and J. Borovec, "Ultralytics/yolov5: V4.0–nn.silu() activations, weights & biases logging, pytorch hub integration," Jan, 2021. [Online]. Available: https://zenodo.org/record/4418161

[19] J.-J. Liu, Q. Hou, M.-M. Cheng, C. Wang, and J. Feng, "Improving convolutional networks with self-calibrated convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10 096–10 105.

[20] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, Dec. 2020.

[21] R. E. McRoberts and E. O. Tomppo, "Remote sensing support for national forest inventories," *Remote. Sens. Environ.*, vol. 110, no. 4, pp. 412–419, Oct. 2007.

[22] H. Kaartinen, J. Hyyppä, X. Yu, M. Vastaranta, H. Hyyppä, A. Kukko, M. Holopainen, C. Heipke, M. Hirschmugl, F. Morsdorf, E. Næsset, J. Pitkänen, S. C. Popescu, S. Solberg, B. Wolf, and J. Wu, "An international comparison of individual tree detection and extraction using airborne laser scanning," *Remote. Sens.*, vol. 4, no. 4, pp. 950–974, Mar. 2012.

[23] S. Malek, Y. Bazi, N. Alajlan, H. Al-Hichri, and F. Melgani, "Efficient framework for palm tree detection in UAV images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, vol. 7, no. 12, pp. 4692–4703, Dec. 2014.

[24] J. Secord and A. Zakhor, "Tree detection in urban regions using aerial lidar and image data," *IEEE Geosci. Remote. Sens. Lett.*, vol. 4, no. 2, pp. 196–200, Apr. 2007.

[25] P. Srestasathiern and P. Rakwatin, "Oil palm tree detection with high resolution multi-spectral satellite imagery," *Remote. Sens.*, vol. 6, no. 10, pp. 9749–9774, Oct. 2014.

[26] H. Jiang, S. Chen, D. Li, C. Wang, and J. Yang, "Papaya tree detection with UAV images using a gpu-accelerated scale-space filtering method," *Remote. Sens.*, vol. 9, no. 7, p. 721, Jul. 2017.

[27] 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," *arXiv:2105.04356*, May. 2021. [Online]. Available: https://arxiv.org/abs/2105.04356

[28] S. Hartling, V. Sagan, P. Sidike, M. Maimaitijiang, and J. Carron, "Urban tree species classification using a worldview-2/3 and lidar data fusion approach and deep learning," *Sensors*, vol. 19, no. 6, p. 1284, Mar. 2019.

[29] T. Dong, Z. Jian, G. Sibin, S. Ying, and F. Jing, "Single-tree detection in high-resolution remote-sensing images based on a cascade neural network," *ISPRS Int. J. Geo Inf.*, vol. 7, no. 9, p. 367, Sep. 2018.

[30] M. Liu, Z. Han, Y. Chen, Z. Liu, and Y. Han, "Tree species classification of lidar data based on 3d deep learning," *Measurement*, vol. 177, p. 109301, Jun. 2021.

[31] S. Briechle, P. Krzystek, and G. Vosselman, "Silvi-net - A dual-cnn approach for combined classification of tree species and standing dead trees from remote sensing data," *Int. J. Appl. Earth Obs. Geoinformation*, vol. 98, p. 102292, Jun. 2021.

[32] M. P. Ferreira, D. R. A. de Almeida, D. de Almeida Papa, J. B. S. Minervino, H. F. P. Veras, A. Formighieri, C. A. N. Santos, M. A. D. Ferreira, E. O. Figueiredo, and E. J. L. Ferreira, "Individual tree detection and species classification of amazonian palms using uav images and deep learning," *For. Ecol. Manag.*, vol. 475, p. 118397, Nov. 2020.

[33] Q. Xie, D. Li, Z. Yu, J. Zhou, and J. Wang, "Detecting trees in street images via deep learning with attention module," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 8, pp. 5395–5406, Aug. 2020.

[34] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[35] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[36] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[37] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[38] 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 (Nips)*, vol. 28, pp. 91–99, Dec. 2015.

[39] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, May. 2019.

[40] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid r-cnn," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7363–7372.

[41] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[42] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv:2004.10934*, Apr. 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[43] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv:1804.02767*, Apr. 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[44] 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 *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, Sep. 2014, pp. 740–755.

[45] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2010.

[46] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 390–391.

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

[48] W. Shi and R. Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1711–1719.

[49] D.-J. Chen, H.-Y. Hsieh, and T.-L. Liu, "Adaptive image transformer for one-shot object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12 247–12 256.

[50] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 286–301.

[51] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7132–7141.

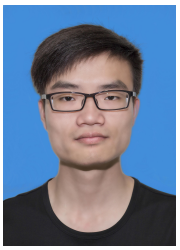
[52] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: Frequency channel attention networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 783–792.

[53] L. Tychsen-Smith and L. Petersson, "Improving object localization with fitness NMS and bounded iou loss," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 6877–6885.

[54] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13 039–13 048.

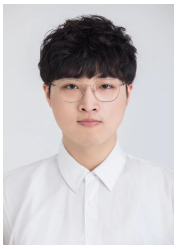
[55] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14 454–14 463.

[56] "Colabeler," May, 2021. [Online]. Available: http://www.colabeler.com



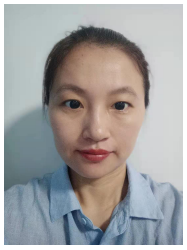
Yongzhen Wang received the M.S. degree in 2019. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His research interests include deep learning, image processing and computer vision, particularly in the domains of object detection, image dehazing, and image deraining. He has served as a PC member for AAAI 2022.

Xuefeng Yan is a professor of the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), China. He obtained his PhD degree from Beijing Institute of Technology in 2005. He was the visiting scholar at Georgia State University in 2008 and 2012. His research interests include intelligent computing, MBSE/complex system modeling, simulation and evaluation.



Hexiang Bao received the B.E. degree from Hefei University, in 2017. He is currently pursuing the M.S. degree with computer science and technology from Nanjing University of Aeronautics and Astronautics (NUAA). His research interests include deep learning, image processing, and object detection.

Yiping Chen (M'11–SM'20) received the Ph.D. degree in information and communications engineering from the National University of Defense Technology, Changsha, China, in 2011. She is a research associate professor with the Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, China. From 2007 to 2011, she was an Assistant Researcher with Chinese University of Hong Kong, China. Her current research interests include remote sensing image processing, mobile laser scanning data analysis, 3D point cloud computer vision and autonomous driving. She has published more than 70 papers in referred journals, including IEEE Transactions on Intelligent Transportation Systems, IEEE Transactions on Geoscience and Remote Sensing, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, and conferences, including CVPR, IGARSS, and ISPRS. She was a receipt of the 2020 Best Reviewer of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.



Lina Gong is currently a lecturer in the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. She received her Ph.D. degree in the Computer software and theory from China University of Mining and Technology, China. She also studied as a visitor one year in the Software Analysis and Intelligence Lab (SAIL), School of Computing, Queen's University, Canada. Her research interests include Deep learning, software analysis, software testing and mining software repositories.

Mingqiang Wei received his Ph.D degree (2014) in Computer Science and Engineering from the Chinese University of Hong Kong (CUHK). He is a professor at the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA). Before joining NUAA, he served as an assistant professor at Hefei University of Technology, and a postdoctoral fellow at CUHK. He was a recipient of the CUHK Young Scholar Thesis Awards in 2014. He is now an Associate Editor for the Visual Computer Journal, Journal of Electronic Imaging, Journal of Image and Graphics, and a Guest Editor for IEEE Transactions on Multimedia. His research interests focus on 3D vision, computer graphics, and deep learning.

Jonathan Li (Senior Member, IEEE) received the Ph.D. degree in geomatics engineering from the University of Cape Town, South Africa, in 2000. He is a Professor with the Department of Geography and Environmental Management and cross-appointed with the Department of Systems Design Engineering, University of Waterloo, Canada and a Fellow of the Engineering Institute of Canada. His main research interests include image and point cloud analytics, mobile mapping, and AI-powered information extraction from LiDAR point clouds and earth observation images. He has co-authored over 500 publications, including 300+ in refereed journals and 200+ in conference proceedings. Dr. Li is a recipient of the 2021 Geomatica Award, 2020 Samuel Gamble Award, and 2019 Outstanding Achievement Award in Mobile Mapping Technology. He is currently serving as the Editor-in-Chief of International Journal of Applied Earth Observation and Geoinformation, Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and Canadian Journal of Remote Sensing.