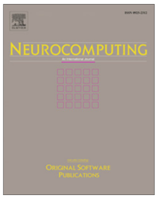
Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Towards densely clustered tiny pest detection in the wild environment

Jianming Du [a], Liu Liu [b,*], Rui Li [a], Lin Jiao [c], Chengjun Xie [a], Rujing Wang [a,d]

[a] *Hefei Institute of Intelligent Machines, Chinese Academy of Sciences, China*
[b] *Shanghai Jiao Tong University, China*
[c] *Anhui University, China*
[d] *University of Science and Technology of China, China*

ARTICLE INFO

Article history:
Received 10 August 2021
Revised 16 October 2021
Accepted 4 December 2021
Available online xxxx

Keywords:
Pest detection
Clustered tiny object
Small object detection
Image dataset

ABSTRACT

Our life is populated with many small-size objects, such as human in aerial images and tiny pests in agriculture. Current generic and small object detection methods are only focus on tackling their sizes rather than distribution. Considering this limitation, we state a Densely Clustered Tiny (DCT) object detection problem using a novel metric Object Density Level (ODL) to measure the object distribution in an image. The DCT problem allows varied densely distributed objects in the real-world captured images. In dealing with the DCT problem, we select two kinds of aphids that usually gather into cliques in the real-world agricultural environment, and build an aphid dataset APHID-4K in our task. Accompanying the DCT task, we propose a novel DCT detection network (DCTDet) to address this challenge. Specifically, a Cluster Region Proposal Network (ClusRPN) is trained to select appropriate densely distributed object cluster regions from images. These candidates are classified into different groups according to their density. A Density Merging and Partition module (DMP) merges and partitions them respectively and finally outputs cluster regions with uniform size and density to a subsequent Local Detector Group (LDG). In addition, we also use Composited Cluster data Generation (CCG) to present a large-scale dataset for ClusRPN optimization for robust training procedure and theoretically analyze their effects in detail. Experiments on APHID-4K and another clustered small object detection dataset VisDrone show that our DCTDet achieves state-of-the-art performance.

1. Introduction

Object detection based on computer vision plays a significant role in the monitoring and surveillance applications. The high-resolution small object detection has become an important field, such as object detection in high-altitude aerial images [1,8,51]. However, in some specific applications, the objects are different from those in aerial image datasets, such as the aphid detection in wild field. The in-field aphid detection is an integral part of the automatic monitoring of agricultural pests. According to the current requirements of in-field pest automatic monitoring applications, a person without any professional knowledge should be able to quickly judge the presence of pests in a photo captured by mobile cameras, as well as their quantity and location for severity level evaluation and precision applying pesticide. While aphids may only occupy about 100~400 pixels, which are much smaller than the "small" objects($32 \times 32 = 1024$ pixels) defined in MS COCO [20]. Meanwhile, they often distribute in very small areas, which are denser and smaller than objects in aerial images. Therefore, we state a new problem: densely clustered tiny(DCT) object detection. There are three main features of this problem: 1) most objects are tiny; 2) objects are densely distributed in cluster regions; 3) the cluster regions are small. In order to evaluate the distribution of small and tiny objects on 2D images, we utilize a metric named object density level(ODL) to measure their clustering degree using a sliding window. In the aphid dataset, the density levels of clustered objects are much higher than those in small object detection tasks. Therefore, we name the new task as DCT detection task.

There are two challenges for addressing the problem: 1) To the best of our knowledge, there is no publically available dataset for the DCT detection task, especially in specific applications, such as in-field aphid detection. Although there are many datasets for clustered small object detection like aerial image datasets [56,6,48], their cluster areas are large, and the density is not high enough. 2) The existing methods for clustered small object detection underperform when they are directly applied to our task. Due to the features of DCT task, it is difficult for the existing methods to accurately locate the cluster regions and select precise size through

[*] Corresponding authors.
E-mail addresses: djming@iim.ac.cn (J. Du), liuliu1993@sjtu.edu.cn (L. Liu).

https://doi.org/10.1016/j.neucom.2021.12.012
0925-2312/© 2021 Elsevier B.V. All rights reserved.

scaling processing. So, the results of region extraction are generally larger than their actual sizes(as shown in Fig. 1).

To solve the first challenge, we build a specific in-field aphid dataset named APHID-4K for the DCT detection task. This dataset contains more than four thousand wheat aphid images captured by specific mobile image collection devices. A large proportion of the aphid objects meet the features of DCT task.

To address the challenges of the insufficient methodology of the DCT detection task, we propose a Densely Clustered Tiny object Detection network (DCTDet) based on the cluster region proposal. DCTDet consists of three core components, including a Cluster Region Proposal Network (ClusRPN), a Density Merging and Partition module(DMP), and a Local Detector Group (LDG). ClusRPN predicts the density of assigned regions in an image according to the ODL and output candidates of cluster region chips. Then the chips are categorized into several density groups and sent through DMP to merge the overlapping regions and divide them into suitable sizes. Then these merged chips are sent to subsequent LDG for local object detection. Finally, the local detection results are fused with those from the global detector using a Non-Maximum Suppression (NMS) post-processing step [27]. In order to fully exploit value of the limited clustered regions in the non-large-scale dataset, such as APHID-4K due to the collection difficulties, we present a Composited Cluster data Generation approach (CCG) to generate a external large-scale dataset containing additional distribution information to optimize the prediction performance of ClusRPN in the training phase.

Compared with existing approaches, DCTDet shows several advantages: 1) Comparing with [8,15], which conducted initial detection first to obtain dense information, ClusRPN directly outputs regions with density levels and greatly reduces the computation cost; 2) By using DMP, the chips sent to fine detectors have similar sizes and density, reducing the impact of size span and non-uniform density on the local detection performance; 3) Our method only optimizes the cluster region extraction stage without any modification to the local detection network and the fusion strategy, so our method can be easily applied to the existing state-of-the-art methods, such as methods in [39,53] for further improvements.

Our contributions are summarized as follow:

1) A novel object detection task named densely clustered tiny (DCT) object detection is proposed. We state this task in detail and present a related dataset APHID-4K.
2) We propose a novel DCTDet network to address the challenges of DCT detection task. DCTDet exploits the density information to improve the performance of DCT detection.
3) The proposed method outperforms state-of-the-art methods of clustered small object detection on APHID-4K. Extensive experiments suggest that the proposed method achieves state-of-the-art performance on VisDrone.

2. Related work

Generic Object Detection. Generic object detection has developed rapidly based on the deep convolutional neural networks (CNNs) [26] and image recognition [13]. Most of the state-of-the-art object detectors [36,12,28] have a similar architecture consisting of a deep backbone network for feature extraction from images and a hierarchical neck network [18] laterally connecting for connections of semantic feature maps and a head network [34,2] for classification and localization of objects. According to the detection pipeline, the existing methods can be roughly classified into region-based [9,34,11] and region-free [23,32,19,55,33]. The main difference between these two categories is whether the detector performs a candidate region selection on the feature map before classification and localization. This stage significantly improves accuracy at the expense of reduced efficiency.

In-field Pest Detection. In-field pest control and prevention have become a top-priority task in agriculture around the world [47,16]. Due to the time–costing and labor-consuming issues of traditional manual pest monitoring, automatic pest monitoring through fixed and mobile cameras is increasingly being used, followed by a huge demand for well-performing pest image detec-

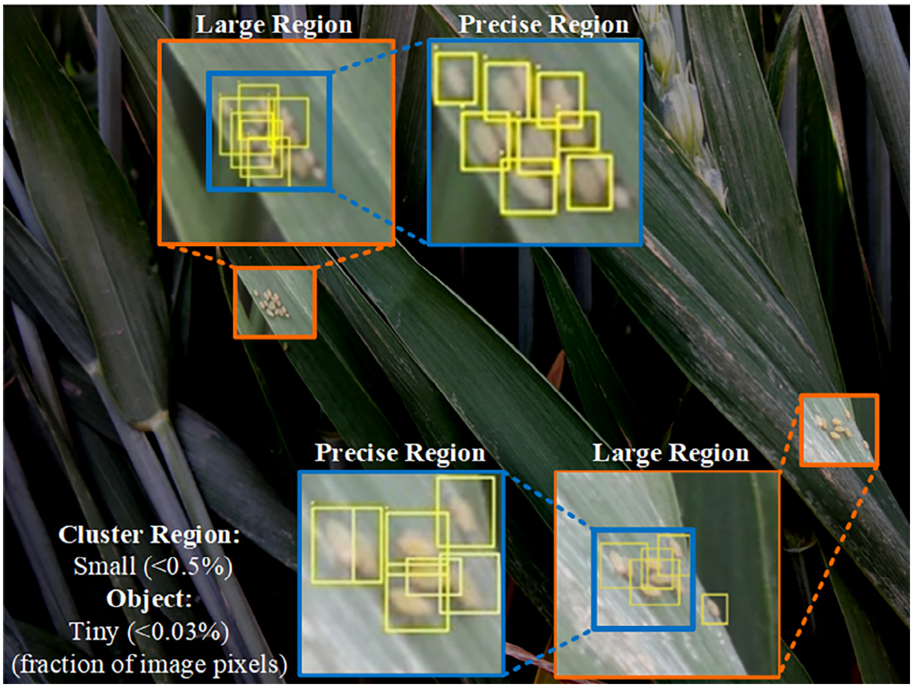


Fig. 1. The main features of aphids: 1) extremely small in size; 2) densely distributed; 3) small cluster region. In this case, the different size of region selection brings significant change in internal density and size span of inclusive objects.

tion. Compared with the early works using image processing algorithms [46,31], recent methods [49,40,47,42] inspired by CNNs and generic object detection has achieved remarkable successes. However, tiny scale and high density remain the main issues limiting the performance of pest detection [16,21].

Small object detection. In the small object detection tasks, the main challenge is that the features of the edges of small objects are not obvious, so that most existing general object detection methods can hardly perceive them. Many works [10,22,44] proposed novel feature fusion methods for FPN-based detectors [18], enabling the networks to obtain more object information from shallow feature layers. Some works [52,17] also utilized the attention and context semantic information of objects from high-level feature layers. Other methods proposed new metrics [43,35,54,50] based on the IoU and AP to get rid of the sensitive evaluation of bounding boxes of small objects. In addition, Super Resolution(SR) is also an effective way to recover the information of low-resolution objects [29,5]. Although these methods performed well in small object detection tasks, the clustered small object detection task is still a challenge.

A general solution of clustered small object detection is to simulate the human vision process which divides the detection process into two stages: 1) select local regions from the global image as the local receptive fields; 2) conduct up-sampling operation and fine detection on these local regions. In the region selecting stage, the early works [7,25] relied on randomly or evenly dividing the image to obtain regions that may contain objects. Although these methods achieved good performances, they seriously waste computing resources when the image contains many sparse objects. To accurately select the regions of interest as effective receptive fields, many region searching methods for small objects have been proposed. These methods can be roughly categorized into direct region selection and indirect region selection. The direct selection is to train a network to propose candidate regions directly. [37] conducted multi-scale training by processing context-regions around ground-truth instances generated using a region proposal network. [14] designed a RPN-like network to propose regions of object of interest(ROOBI) from a large-scale aerial video frame. [51] suggested a Cluster Proposal sub-net(CPNet) aiming to address the cluster region searching before detection. The work of [39] proposed a CPEN network to locate the cluster regions by predicting the center points of clusters of small objects. The indirect selection is to perform an initial detection to obtain density information first, then select regions based on clustering algorithms. Some methods [8,45] used a coarse detector to obtain the rough object distribution information so as to locate cluster regions. [15] introduced the density map, which is popular in crowd counting tasks, into the distribution estimation and cluster localization.

However, these region searching methods have some disadvantages in the DCT detection task: 1) the initial detection of indirect methods easily mistake dense areas for sparse areas due to the missed detection of tiny objects; 2) the output candidates tend to contain more objects rather than uniformly clustered objects because the methods based on cluster region proposal usually use the number of inclusive objects to evaluate the density. These issues cause the existing methods to output larger regions but not precise regions in DCT detection task.

3. Problem statement

As mentioned earlier, the DCT detection problem setting advances the small object detection and clustered object detection in three aspects: the sizes of individual objects are tiny, the clusters are dense, and the cluster regions are small. In order to state the DCT detection problem more clearly, we propose the following process to evaluate a dataset.

Given a RGB single image I_i from the dataset $\mathscr{D} = \{I_1, I_2, \ldots, I_K\}$ as input. The objects in this image are annotated as bounding boxes $\mathscr{B} = \{b_j\}_{j=1}^{M}$ as well as their categories $\mathscr{C} = \{c^j\}_{j=1}^{M}$.

Firstly, we expand the definition of size metrics in COCO by adding a new evaluation size $16 \times 16 = 256$ pixels. The objects smaller than this threshold are defined as “tiny” objects.

Then, a square multi-size window is used sliding on the image. We use object density level (ODL) to evaluate the density degree in the current window w, which is defined as following:

$$ODL_w = \log\left(\frac{N_w}{S_w}\right) + O \tag{1}$$

where S_w is the area size of the window, N_w is the number of objects in the window, O is the offset coefficient to ensure that the level is positive within the valid area, which is 10 in this paper. We use square windows to measure ODL. For convenience, the side length L_w of windows is used instead of the area size to represent to window sizes in this paper. As shown in Fig. 3, when the window size is fixed, the larger the number of inclusive objects is, the higher the ODL is, and the denser the inclusive cluster is. We divide the ODL into five intervals to represent different clustering degrees, namely: no cluster (1–2), sparse cluster (2–2.5), medium cluster (2.5–3), general cluster (3.0–3.5) and dense cluster (3.5–4).

When the window frames objects, we calculate the ODL and expand the window until current clustering degree decreases. Then we regard the current window size as the edge size L_{ODL} of the cluster. By calculating the average edge size $\bar{L}_{ODL}$ of each degree in the image, we obtain the distribution of cluster sizes in the image.

Finally, we define that a DCT detection task should meet the following requirements:1) the majority of objects are smaller than “tiny” threshold; 2) the proportion of clusters in dense cluster degree is high; 3) the edge size of dense cluster degree is small.

4. Dataset

APHID-4 K. The dataset contains 4,294 real-world wheat aphid images (3,435 images for training and 859 images for testing) with 54,681 annotated objects of two categories of aphids located on diverse backgrounds, including leaf surface, wheatear, straw root, and ground. The resolution of images is about 1440 × 1080 pixels. We spent two years collecting aphids images at different growth stages of wheat using a specific collection device. This collection device consists of a front macro camera, a mobile data transmission terminal, and a retractable carbon-fiber bracket. The camera can easily reach into the depths of wheat crops to take pictures and automatically upload them to the server for storage and analysis. We annotate aphids according to the standard process of the COCO dataset. We do not generate additional ground-truth of clusters for training like [51] did. In order to meet our DCT detection task, we screened out the images containing fewer than five objects from our collections.

We compare the APHID-4K with the representative dataset of small object detection VisDrone [57,56], the results are shown in Fig. 5. (a) shows that the tiny objects occupy the largest proportion of APHID-4K, while the object sizes of VisDrone vary. As shown in (b), the majority of clusters in APHID-4K are in dense cluster degree(3.5–4), while VisDrone has the largest proportion in general cluster degree(3–3.5). (c) shows that the average edge size of clusters in APHID-4K is much smaller than that in VisDrone. According to our previous statement on DCT detection, APHID-4K meets the features of DCT task, while VisDrone is a typical clustered small object dataset. Fig. 4.

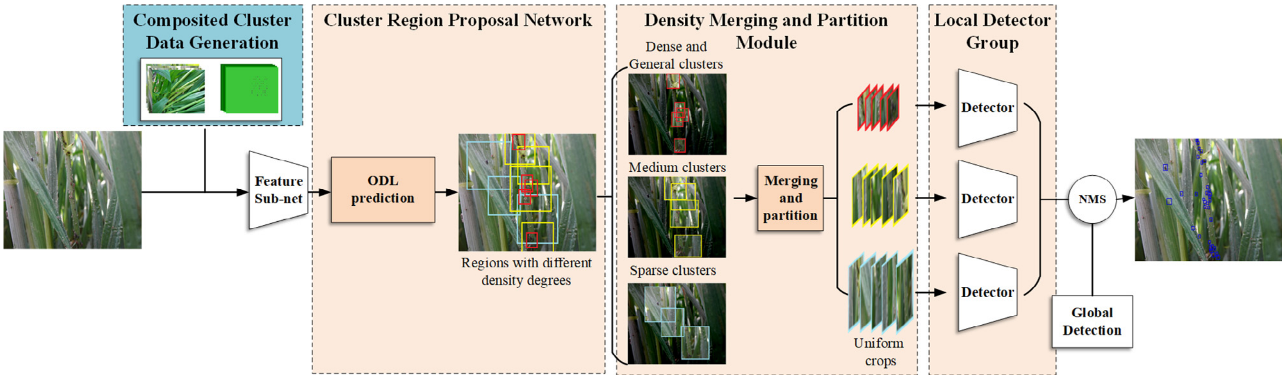

Composited Cluster Data Generation
Cluster Region Proposal Network
Density Merging and Partition Module
Local Detector Group
Feature Sub-net
ODL prediction
Regions with different density degrees
Dense and General clusters
Medium clusters
Sparse clusters
Merging and partition
Uniform crops
Detector
Detector
Detector
NMS
Global Detection

Fig. 2. Densely Clustered Tiny Object Detection network (DCTDet). DCTDet consists of three main parts: 1) a cluster region proposal network for region selecting; 2) a density merging and partition module for uniforming region chips; 3) a local detector group for fine detection. Also, a composited cluster data generation pipeline is used to generate images containing clustered objects.

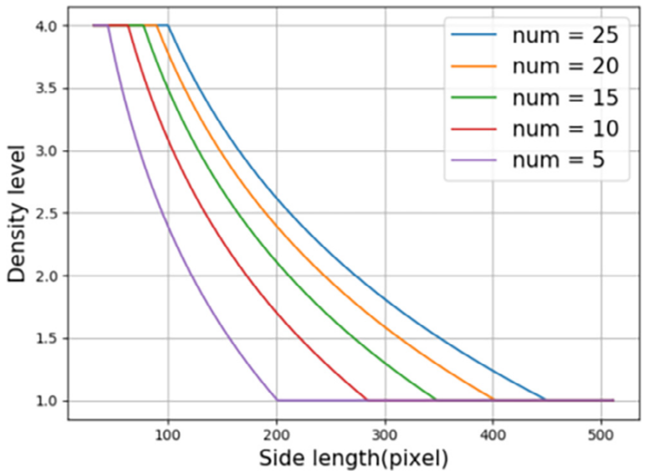


Fig. 3. Illustration of ODL. In practice, it is necessary to set maximum and minimum limits to ensure that the levels are within a valid range.

Composited Cluster Data Generation. Compared with the collection of aerial images, it is far more challenging to collect DCT data in many specific applications. We take the establishment of the APHID-4K dataset as an example. Since the time and space spans of pest occurrence are usually huge, it is labor-consuming and time–cost to obtain high-quality images of cluster pests in the field. Besides, due to the requirement of hiring agricultural experts to guide the labeling process, the overall cost of establishing a large-scale dataset is extremely high.

In order to fully exploit the value of real data in a relatively small DCT dataset to complete accurate cluster region selection, we present a large-scale dataset of additional distribution information using a general image generation method for the DCT detection task, including two approaches: CCG-R and CCG-F. The CCG-R aims to generate roughly realistic images, the CCG-F generates totally fake cluster-only images. Along with APHID-4K, we present two composited cluster data generated by each approach, containing 32,000 images, respectively.

The pipeline of the CCG-R approach is as follows:

1) randomly cut 200 real individual aphids from the original images as paste materials; 2) select a large number of crop images that do not contain aphids as paste backgrounds; 3) set a specific

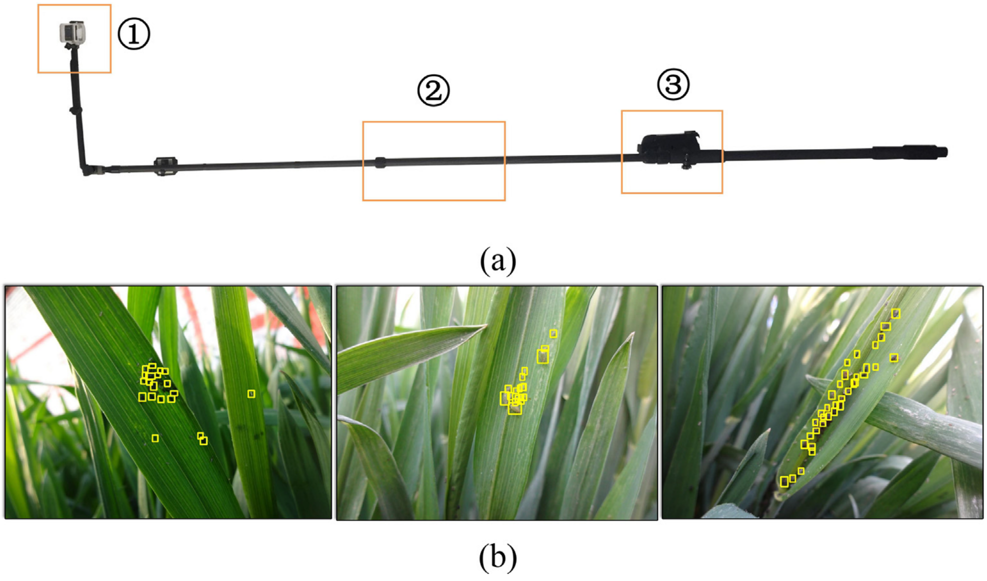


Fig. 4. (a) Collection device: 1-camera, 2-bracket, 3-mobile terminal; (b) Samples of APHID-4K. Aphids are tiny in size, and their distributions are complex and diverse.

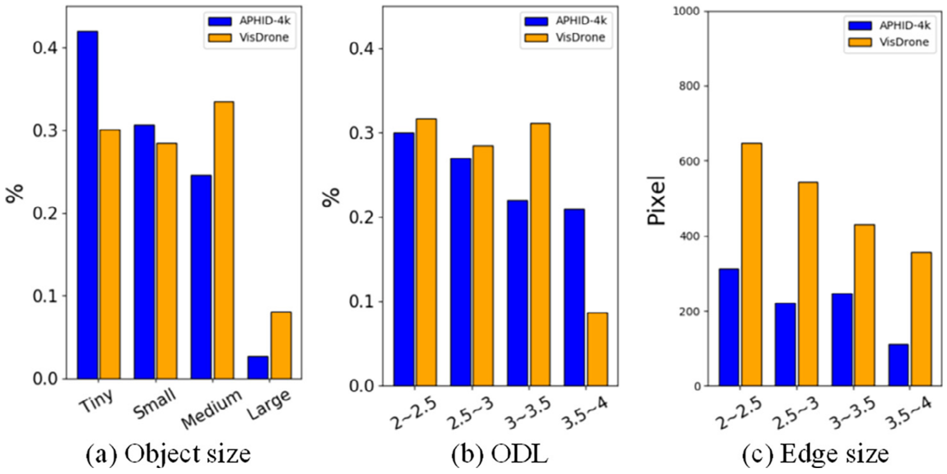


Fig. 5. Comparison between APHID-4K and VisDrone. (a) Object average size. (b) ODL distribution of clusters. (c) Average edge size. The regions in no cluster degree(less than 2 ODL) are not taken in consideration.

paste density level and calculate the paste region size L and individual number N, and then make a corresponding $M \times M$ square grid; 4) randomly select a crop leaf position in background as the paste centre; 5) randomly select individual aphids from paste materials and paste them in the grid around the paste centre; 6) use the Poisson fusion method to make the paste boundaries more naturally (Fig. 6).

In the CCG-F, aiming to generate images with density-only information, we completely ignore the geometrical and morphological features of aphids by replacing them with small elliptical spots and annotate them as a new category "fake". Then we use a solid color (green in the paper) instead of crop images as backgrounds.

Different from the general data augmentation methods that directly expand the original dataset during the whole training phase, CCG is independent of the original data and only targets the training phase of the cluster region proposal network. Because in the DCT tasks, even in a small-scale dataset, the number of individual objects is already large enough while the number of dense regions is extremely small(the gap is about 10–100 times). This imbalance makes region proposal network hard to be well trained. Therefore, the region proposal network needs additional training to obtain better region prediction performance. However, the general dataset expansion methods are prone to cause over-training of local detectors, and the image distortion caused by the simulation data could lead to potential deteriorated performance of local detectors.

We explain the CCG training mechanism in detail in Section 5.5.

5. Densely clustered tiny object detection network (DCTDet)

5.1. Overview

There are two main problems with DCT tasks: 1) the size of object is too small, which are hardly detected because the edge features of individual objects are not obvious; 2) the large size and density span between each clustered regions affect the detection accuracy of the single fine detector. As shown in Fig. 2, to address the problems, the proposed method is designed in two aspects: 1) the ClusRPN is designed to perceives the gathering regions and an up-sampling process is conducted during the fine detection, which increases the size of individual objects so that the tiny objects are easier to distinguish by enlarged edges; 2) the region chips are divided into multiple groups according to the density and the chip sizes in each group are uniformed by DMP, then the LDG is used to separately detect each chip group, which effectively avoids the problem of size and density span. Therefore, the proposed method improves the detection performance of clustered object detection networks in DCT tasks.

5.2. Cluster region proposal network (ClusRPN)

Motivated by the region proposal network (RPN), which outputs rough location candidates of objects, the ClusRPN output location candidates of cluster regions.

In the training phase, a window of different sizes slides on the images and calculates ground-truth ODL based on the window size

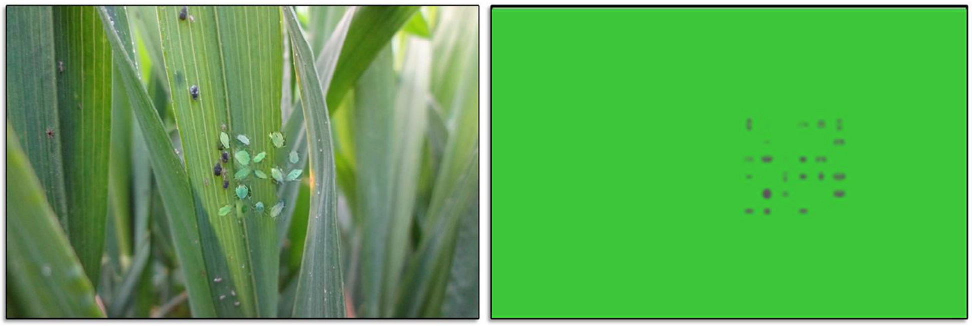


Fig. 6. Examples of two generation approaches (enlarged for best view). These approaches are very simple and do not require very realistic results.

and the number of inclusive objects in the current region. Then ClusRPN calculates the loss of current region based on the ground-truth ODL and the predicted ODL. The loss function of ODL for an image is defined as:

$$\mathscr{L}(\{s_i\}) = \frac{1}{N_{clus}} \sum_i \mathscr{L}_{clus}\left(s_i, s_i^*\right) \tag{2}$$

where s_i is the predicted ODL of the i-th region, and s_i^* is the ground-truth ODL. The loss is normalized by the number of regions N_{clus}. And the loss $\mathscr{L}_{clus}\left(s_i, s_i^*\right)$ is defined as:

$$\mathscr{L}_{clus}\left(s_i, s_i^*\right) = \mathrm{SmoothL1}\left(s_i, s_i^*\right) \tag{3}$$

Although ClusRPN shares a similar idea with RPN, they are very different. In terms of purpose, RPN predicts whether there are individual objects at certain positions in a picture, while ClusRPN predicts the object density(ODL) of each position in a picture. In terms of architectures, RPN is composed of a classification branch and a regression branch, while ClusRPN is composed of only one ODL prediction branch. Additionally, since ClusRPN focuses on the region density, the training of ClusRPN is independent to that of object detectors. Therefore, the use of artificial CCG dataset will improve ClusRPN without any impact on subsequent fine detectors.

5.3. Density merging and partition module (DMP)

```
Algorithm 1: Density Merging Process
Input: Initial clusters $\mathscr{B} = \{b_1, \ldots, b_N\}$, corresponding
    clustering degree $\mathscr{D} = \{d_1, \ldots, d_N\}$, merging threshold $N_t$
Note: $OL(a,b) = (a \cap b)/min(a,b)$ means the proportion of
    overlap in the smaller region. $MERGE(\mathscr{T})$ is the operation of
    merging the regions and outputs merged regions with its
    clustering degree.
Output: Merged clusters $\mathscr{M}_b$ and corresponding clustering
    degrees $\mathscr{M}_d$
$\mathscr{M}_b, \mathscr{M}_d \leftarrow \{\}$
while $\mathscr{B} \neq empty$ do
$k \leftarrow \mathrm{argmax}\mathscr{D}$; $\mathscr{T}_b \leftarrow \{\}$;$\mathscr{T}_d \leftarrow \{\}$
    for$b_i$ in $\mathscr{B}$ do
        if $OL(b_i, b_k) \geqslant N_t$ & $d_i = d_k$ then
            $\mathscr{T}_b \leftarrow \mathscr{T}_b \cup \{b_i\}$;$\mathscr{T}_d \leftarrow \mathscr{T}_d \cup \{d_i\}$
        end if
    end for
    if $\mathscr{T}_b \neq empty$ then
        $\mathscr{T} \leftarrow \{\mathscr{T}_b, \mathscr{T}_d\}$
        $b'_k, d'_k \leftarrow MERGE(\mathscr{T})$
        $\mathscr{B} \leftarrow (\mathscr{B} - \mathscr{T}_b) \cup \{b'_k\}$;$\mathscr{D} \leftarrow (\mathscr{D} - \mathscr{T}_d) \cup \{d'_k\}$
    else
        $\mathscr{M}_b \leftarrow \mathscr{M}_b \cup \{b_k\}$;$\mathscr{M}_d \leftarrow \mathscr{M}_d \cup \{b_k\}$
    end if
end while
return $\mathscr{M}_b$, $\mathscr{M}_d$
```

As shown in Fig. 2, the outputs of ClusRPN are a series of candidate cluster regions with various density levels and scales. Many of them with similar density levels are highly overlapped. In order to reduce the computation burden, we need to merge the highly-overlapped regions. In this merging process, a series of clustering degrees defined in Section 3 are used and only those within a same degree are merged to ensure the density in the merged region is still uniform. Let $\mathscr{B} = \{b_1, \ldots, b_N\}$ represent the set of bounding boxes of cluster regions detected by ClusRPN, where $b_i = \{x_i^{min}, y_i^{min}, x_i^{max}, y_i^{max}\}$ are the upper left and bottom right coordinates. $\mathscr{D} = \{d_1, \ldots, d_N\}$ is the corresponding clustering degrees. We use $OL(a,b) = (a \cap b)/min(a,b)$ to measure the overlap between two regions and set a pre-defined merging threshold N_t to determine whether merging is needed. $MERGE(\mathscr{T})$ is a simple operation for merging regions in set $\mathscr{T} = \{b_1, \ldots b_T\}$, which finds the minimum $x_{t\in[1,T]}^{min}, y_{t\in[1,T]}^{min}$ and maximum $x_{t\in[1,T]}^{max}, y_{t\in[1,T]}^{max}$ as the merged region coordinates. The clustering degree of the merged region is the one of the set $\mathscr{T}$.

The implementation process of merging is shown in Alg.1. Briefly, the merging process starts from high to low clustering degrees. It picks a region from $\mathscr{B}$ and merges it with other highly overlapped areas within the same clustering degree. The merged region is put back into the original set $\mathscr{B}$ unless there are not regions to merge. This process will be repeated until all regions in the candidate set $\mathscr{B}$ are picked out. After that, we perform a partition operation for overly large regions to limit the size span of different candidate regions. In this paper, we directly use bisection operation for those regions exceeding the size threshold L_t, and keep some overlapping areas at the edge of the segmentation.

5.4. Local detector group (LDG)

Through DMP, region chips are divided into similarly sized groups according to their ODL. We established a local detector group consisting of several parallel detectors to detect these groups of chips separately. Each detector dedicates the detection at a specific object density in a similar size. Therefore they will not be affected by the large size spans during the resizing operations. Most existing detectors can be utilized as local detectors.

The objective function of a single fine detector can be represented as,

$$L_{fine} = min \frac{1}{N} \sum_{n=1}^{N} \left(\mathscr{L}_{cls}^{n} + \mathscr{L}_{reg}^{n}\right) \tag{4}$$

where N is the number of training samples. When the multiple fine detectors LDG are utilized for different density groups, the objective function is defined as,

$$\begin{aligned} L_{LDG} &= min \sum_{g=1}^{G} L_{fine}^{g} \\ &= min\left(\frac{1}{N^1} \sum_{n^1=1}^{N^1} \left(\mathscr{L}_{cls}^{n^1} + \mathscr{L}_{reg}^{n^1}\right) + \ldots + \frac{1}{N^G} \sum_{n^G=1}^{N^G} \left(\mathscr{L}_{cls}^{n^G} + \mathscr{L}_{reg}^{n^G}\right)\right) \end{aligned} \tag{5}$$

where G denotes the number of density groups, $N = \sum_{g=1}^{G} N^g$ denotes the sum of the training samples in each density group. We assume that the data distribution of each density group is obviously different from others while similar within their own group. Then, using multiple detectors to fit the data distribution of different density groups is intuitively better than using a single detector to fit. Therefore, we obtain a better objective function to improve the entire performance of fine detection.

5.5. Training with composited cluster data

When the cluster regions in the dataset are insufficient during the training phase, ClusRPN relying only on the original datasets usually output larger scale with imprecise density levels (as shown in Fig. 7). To fully exploit these cluster information in dataset to improve the performance of ClusRPN, we use CCG together in ClusRPN training phase.

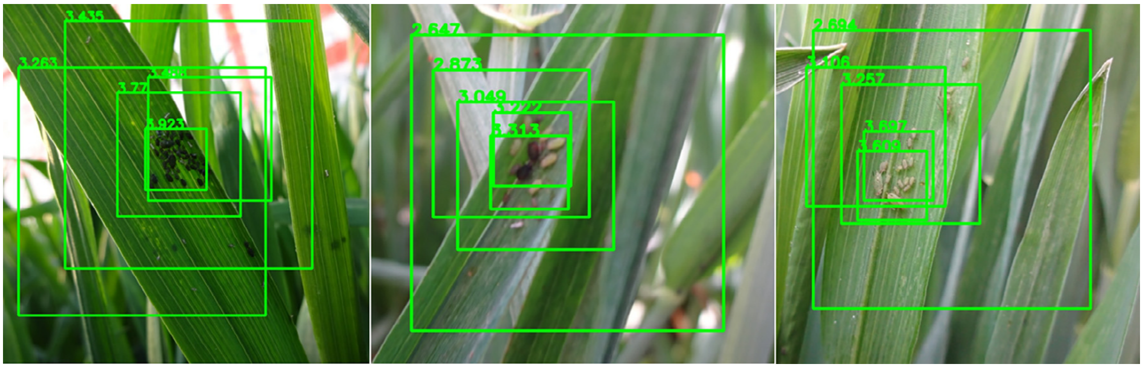


Fig. 7. ClusRPN under-performs due to insufficient training(enlarged for visualization). We find that the network has already been able to select dense cluster areas correctly, but there is a significant deviation of density level prediction. The predicted density levels of low-density areas are much higher than their ground-truth, which causes the size of the final proposed regions to become larger.

CCG provides two kinds of cluster data generation approaches: CCG-R and CCG-F. CCG-R is a common data augmentation method, similar to the Mask Resampling Module in [39]. While CCG-F does not consider the reality of images and only keeps cluster information. In order to better illustrate the feasibility of them, we analyze the effect of CCG on ClusRPN.

According to the function of ClusRPN, it has two main capability: the first capability of ClusRPN is to distinguish the foreground objects from the background, which is similar to the classifier branch of RPN, the second capability is to accurately predict the ODL of the foreground objects. So the integral capability of ClusRPN can be represent as:

$$C_{ClusRPN} = \left\{ C_{cls}^{K},\ C_{pred}^{K} \middle| \ K = \{k_1, k_2, \ldots\} \right\} \tag{6}$$

where K is the set of foreground object categories, C_{cls}^{K} and C_{pred}^{K} denote the capability of classification and ODL prediction.

CCG-R uses the existing object category $k_m \in K$ to generate images, so as these images enable the network to conduct an additional training process on k_m. Since ClusRPN does not distinct categories within K, it is regarded as a further training process of ODL prediction on the whole set K. In this way, C_{pred}^{K} is improved. The issue of the imbalance of categories along with the additional training of classification on category k_m can be resolved by generating objects of multiple categories. As long as the generated objects are realistic enough, C_{cls}^{K} is not affected much. However, in order to ensure the reality of generated clusters, some processing like Poisson fusion are utilized. Particularly in the database of VisDrone, objects of many categories are complex, which makes the implementation of this approach difficult to avoid the distortion of simulated images.

CCG-F generates cluster data containing a fake category $k_{fake} \notin K$. The new set is represented as $\widetilde{K} = \{k_1, k_2, \ldots, k_{fake}\}$. Similar to the first approach, $C_{pred}^{\widetilde{K}}$ is improved by conducting additional training. Different from the first approach, since the added category k_{fake} is extremely different from all other categories, $C_{cls}^{\widetilde{K}}$ for the original set K is almost unaffected. After ignoring the reality of objects, the CCG approach is greatly simplified and brings similar improvement to the CCG-R.

Despite using paste, the idea of CCG-F is totally different from the idea of CCG-R, which is a general data augmentation method. 1) In term of method, CCG-F does not rely on any dataset, which means it does not require any real object and background from real dataset. 2) In term of purpose, CCG-F is to optimize the capability of ClusRPN to find specific density regions from a small-scale data-

Table 1
The detection performance on APHID-4K. We compare our DCT detection methods with other two groups: generic detectors and clustered small object detectors. The inference time is measured on an RTX Titan GPU.

Method	Backbone	CCG	DMP	LDG	AP	AP_{50}	AP_{75}	s/img(GPU)
Generic Detection methods								
FCOS [41]	ResNet101				22.0	61.9	8.7	0.049
Yolov3 [33]	DarkNet53				17.6	52.0	5.7	0.022
RetinaNet [19]	ResNet101				17.5	51.3	6.5	0.054
Faster RCNN [34]	ResNet101				23.6	63.2	10.8	0.062
Faster RCNN + DCN [4]	ResNet101				23.6	62.8	11.4	0.068
Libra Faster RCNN [30]	ResNet101				23.8	61.5	11.7	0.065
Cascade RCNN [2]	ResNet101				24.5	64.6	12.0	0.072
EfficentDet-D5 [38]	EfficentNet				20.5	49.2	9.5	0.139
Clustered small object Detection methods								
SNIPER [37]	ResNet101				25.6	52.1	7.3	0.220
ClusDet [51]	ResNet101				28.5	64.9	8.6	0.297
DMNet [15]	ResNet101				30.1	58.3	15.7	0.323
*DCT Detection methods(**Ours**)*								
DCTDet(Baseline)	ResNet101				25.1	67.4	13.1	0.240
DCTDet	ResNet101	✓			27.0	68.6	13.6	0.240
DCTDet	ResNet101		✓	✓	30.5	71.8	16.3	0.316
DCTDet	ResNet101	✓	✓	✓	33.2	75.3	18.4	0.316
DCTDet + YOLOv3	DarkNet53	✓	✓	✓	**33.9**	**75.8**	**22.1**	**0.218**

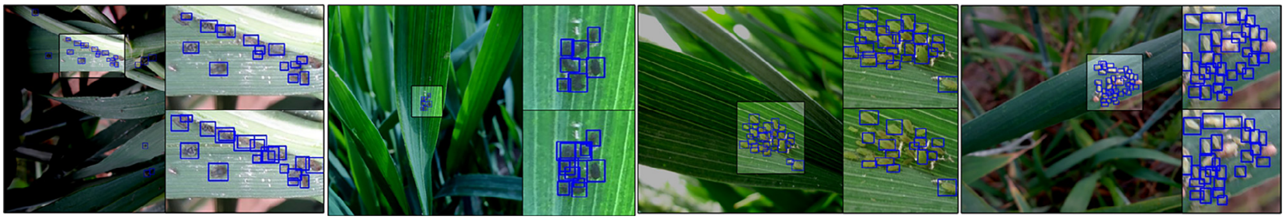


Fig. 8. Qualitative results of DCTDet and ClusDet [51] on APHID-4K. Cluster regions are enlarged for visualization. The uppers are results of DCTDet, the lowers are results of ClusDet. There are obvious challenges of missing detection and inaccurate detection in the results of ClusDet while our DCTDet could perform well.

Table 2
Quantitative results on the validation set of VisDrone dataset. The * denotes the multi-scale inference and bounding box voting are utilized in test phase.

Method	Backbone	AP	AP_{50}	AP_{75}
ClusDet [51]	ResNeXt101	28.4	53.2	26.4
RRNet [3]	Stacked Hourglass	29.1	55.8	27.2
DMNet [15]	ResNeXt101	29.4	49.3	30.6
CPEN + CenterNet [39]	Hourglass-104	29.3	38.7	32.4
CRENet [45]	Hourglass-104	33.7	54.3	33.5
HRDNet [24]	ResNeXt50 + 101	33.5	56.3	**34.0**
DCTDet	ResNeXt101	32.8	56.6	30.8
DCTDet*	ResNeXt101	**33.9**	**57.7**	32.9

set rather than using synthetic realistic images to increase the dataset itself. Therefore, it should be regarded as an optional part of the ClusRPN for non-large-scale dataset training. 3) In term of generality, CCG-F has an extremely simple implementation and can by applied to almost any DCT dataset. Also it perfectly avoiding the potential performance issue of image distortion caused by simulation data.

In Section 6.3, we conduct a series of ablation experiments on this part and analyze the effects in detail.

6. Experiments and analysis

We implement an RTX Titan GPU to train and test our proposed model. On APHID-4K, we use ResNet101 and Cascade R-CNN with ClusRPN as the baseline model of DCTDet. Although DCTDet is a specifically designed method for DCT detection tasks, we apply it to the representative clustered small detection dataset VisDrone for performance comparison to verify its generality. Cascade R-CNN and ResNeXt101 is used in the model. We train the detectors for 24 epochs. The learning rate is 5.0×10^{-3} and the batchsize is 4 in accordance with the setting of ClusNet [51] and DMNet [15]. Other parameters are all followed the default configurations of MMDetection.

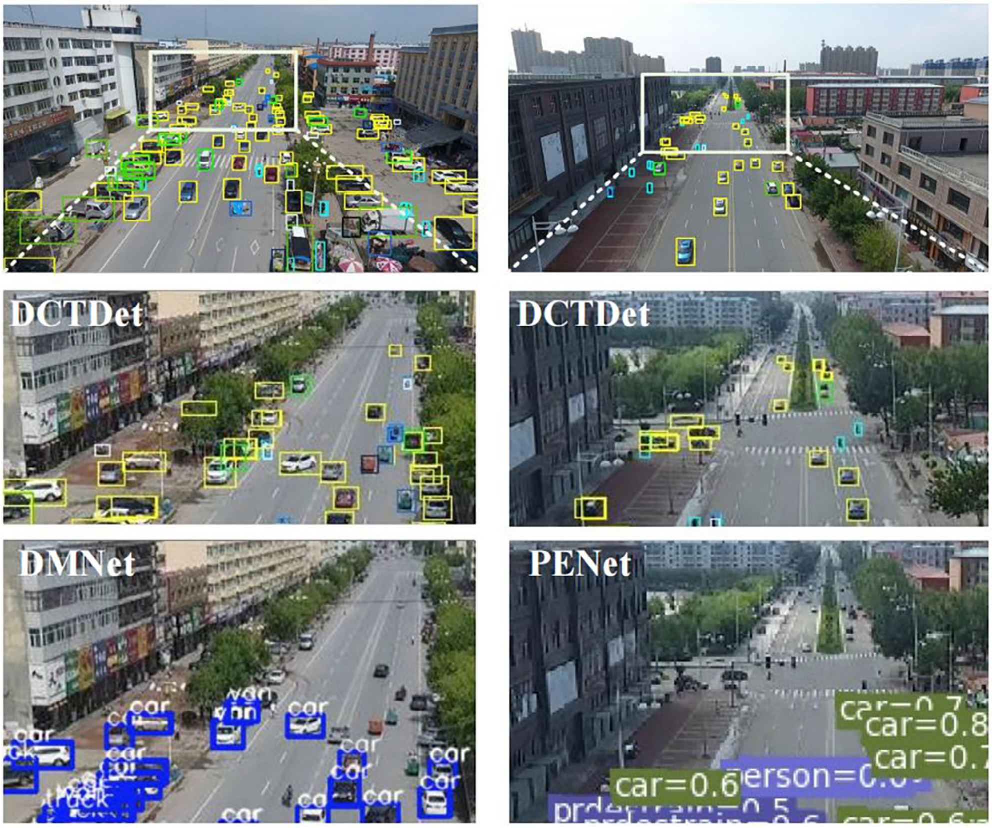


Fig. 9. Visualization of DCTDet detection results on VisDrone. We choose pictures used for visualization results in DMNet [15] (left) and PENet [39] (right). The first row is the visualization results of DCTDet. The second row and the third row show the comparison detection performance between DCTDet and these methods in the tiny. object regions.

6.1. Results on APHID-4K

Table 1 shows that our method achieves 33.9% AP on APHID-4K dataset, which is 9.4% higher than the state-of-the-art generic object detection methods and 3.8% higher than the state-of-the-art clustered small object detection methods. The inference time of the proposed method outperforms state-of-the-art clustered small object detection methods. When using Yolo as fine detectors, our method obtains the best inference speed.

When CCG is added to the baseline of DCTDet, the performance of ClusRPN improves due to the more precise predicted clustered regions. The 1.9% increase in AP demonstrates that the performance of the subsequent fine detection benefits from the precision of region cropping. In addition, DMP and LDG contribute 5.4% improvement in AP to the baseline, which verifies the assumption in Section 5.3 and Section 5.4: the density grouping with uniform size and multiple fine detectors can greatly improve the overall detection performance. When CCG, DMP and LDG are all utilized at the same time, the fine detectors can obtain both precise region prediction and accurate density grouping, DCTDet achieves a further 2.7% improvement. Fig. 8.

6.2. Results on VisDrone

VisDrone contains a lot of densely clustered and sparsely distributed small objects. As shown in Table 2, by using CCG-F to improve ClusRPN and LDG for multiple fine detection, the performance of proposed method achieves comparable performance, which shows that our method performs well in such complex distributed datasets. Need noting that DCTDet is not designed for multi-scale tasks, so it is acceptable that its overall AP is slightly lower than some multi-scale detection methods. From the visualization results in Fig. 9, we can clearly find that the proposed method can perceive tiny objects near the horizon in the distance very well, which are prone to be missed by other detection methods. The inference time of the proposed method is 0.672 s/img and 0.971 s/img when utilizing multi-scale inference and bounding box voting.

6.3. Ablation study

We perform ablation studies to analyze the effects of two important parameters of CCG on the ClusRPN performance: the quantity ratio and the density level. We use three metrics to evaluate the effect: 1) correct rate: the proportion of candidate regions that correctly contain target object; 2) average edge size: the average edge size of candidates of the corresponding density group; 3) average deviation: the difference between the predicted density level and the ground-truth density level, evaluating the overall performance of the network.

Effect of Quantity Ratio. When the quantity ratio is low, the additional training to ClusRPN is not enough. On the contrary, high quantity ratio leads to the network deterioration due to the overwhelming amount of the fake data. We experimentally explore the trade-off between the two issues. As shown in Fig. 10(a), their best performing ratios are level 4 and level 2, respectively. Although the CCG-R is better than CCG-F, the difference is marginal. Overall, the two CCG approaches all markedly improve ClusRPN performance.

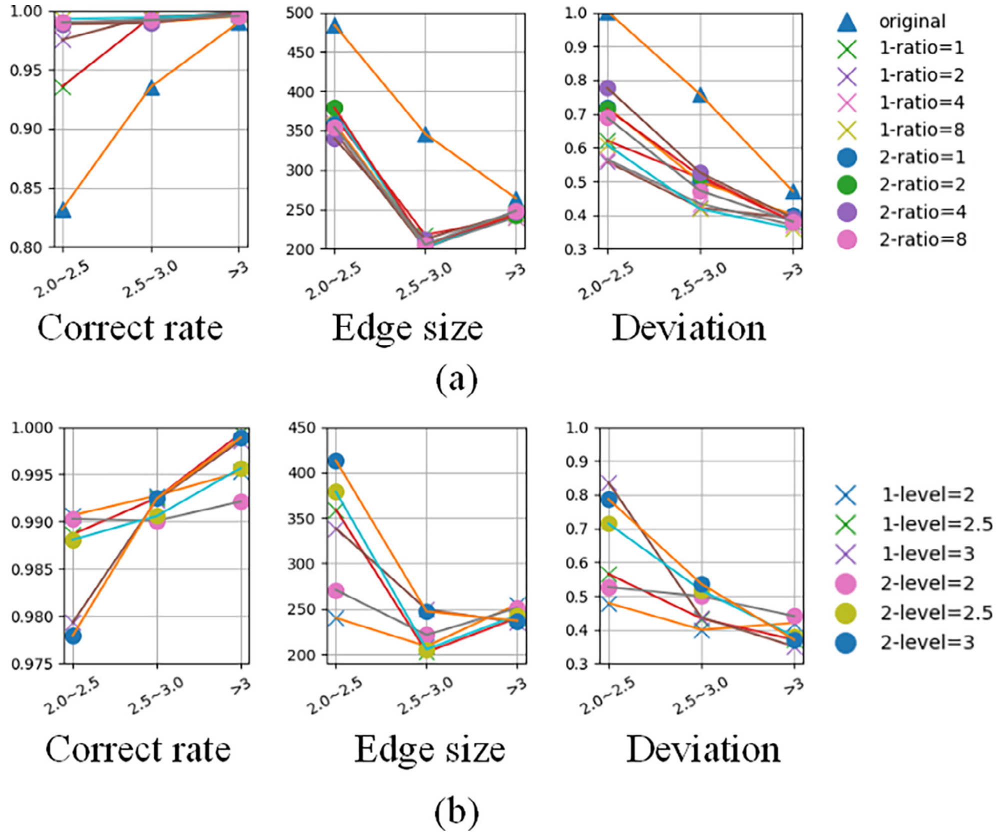


Fig. 10. Comparative result of ablation studies of CCG on APHID-4K. (a) The effect of quantity ratio; (b) The effect of ODL. A performance comparison with the original data is also added to (a). The "1-" and "2-" means the use of the CCG-R and CCG-F.

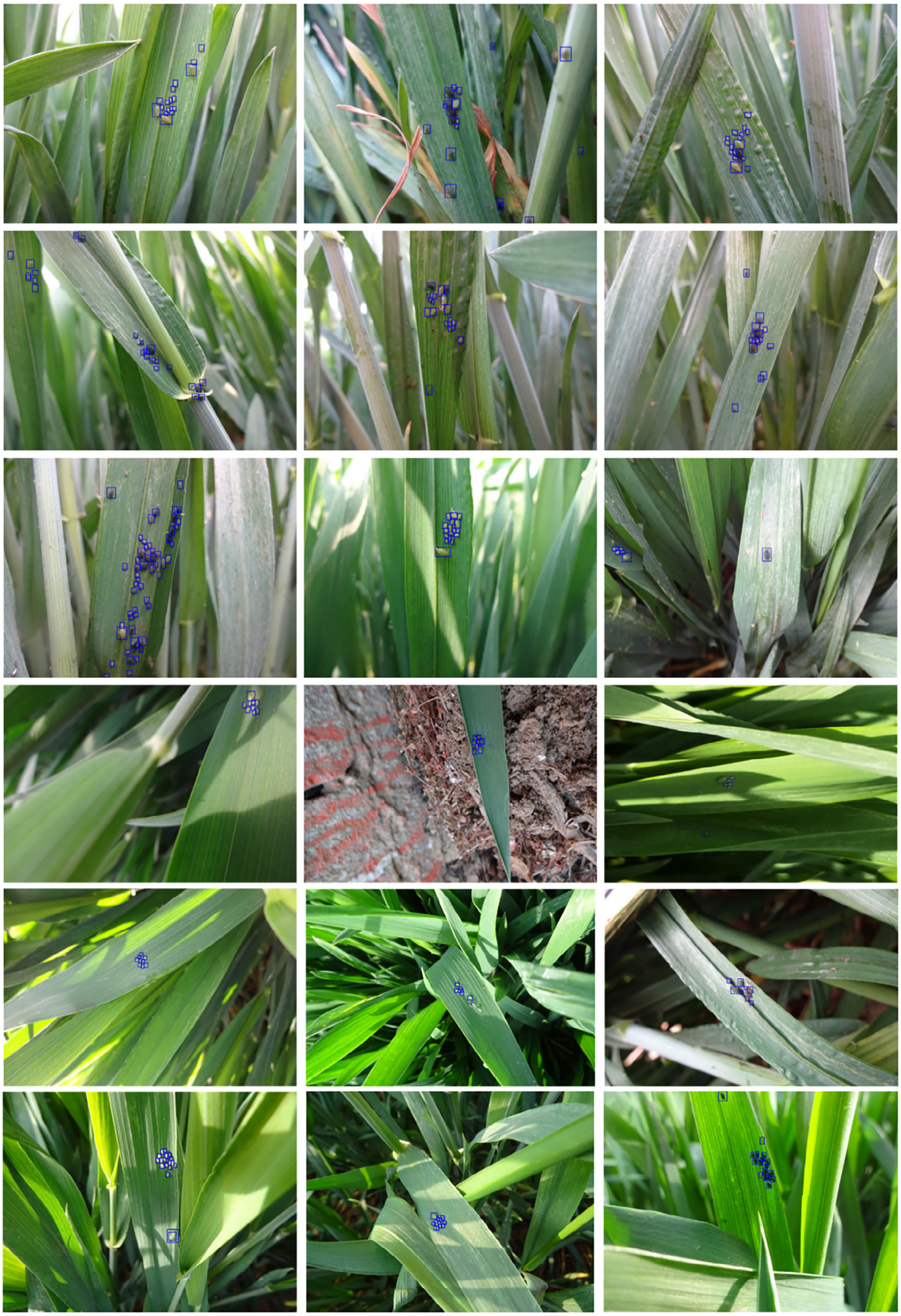


Fig. 11. Visualization results on APHID-4K.

Effect of Density Level. The improvement of ClusRPN varies according to different ODL of generated cluster data. In order to ensure the prediction accurate in ClusRPN, we need to carefully select the ODL of composited data. As shown in Fig. 10(b), the CCG approaches improve the performance of ClusRPN at all density levels. Meanwhile, images generated at specific density levels show better effects on the region proposal performance at the corresponding density levels. In general, for APHID-4K, ClusRPN performs better when the ODL of CCG is 3 (Figs. 11 and 12).

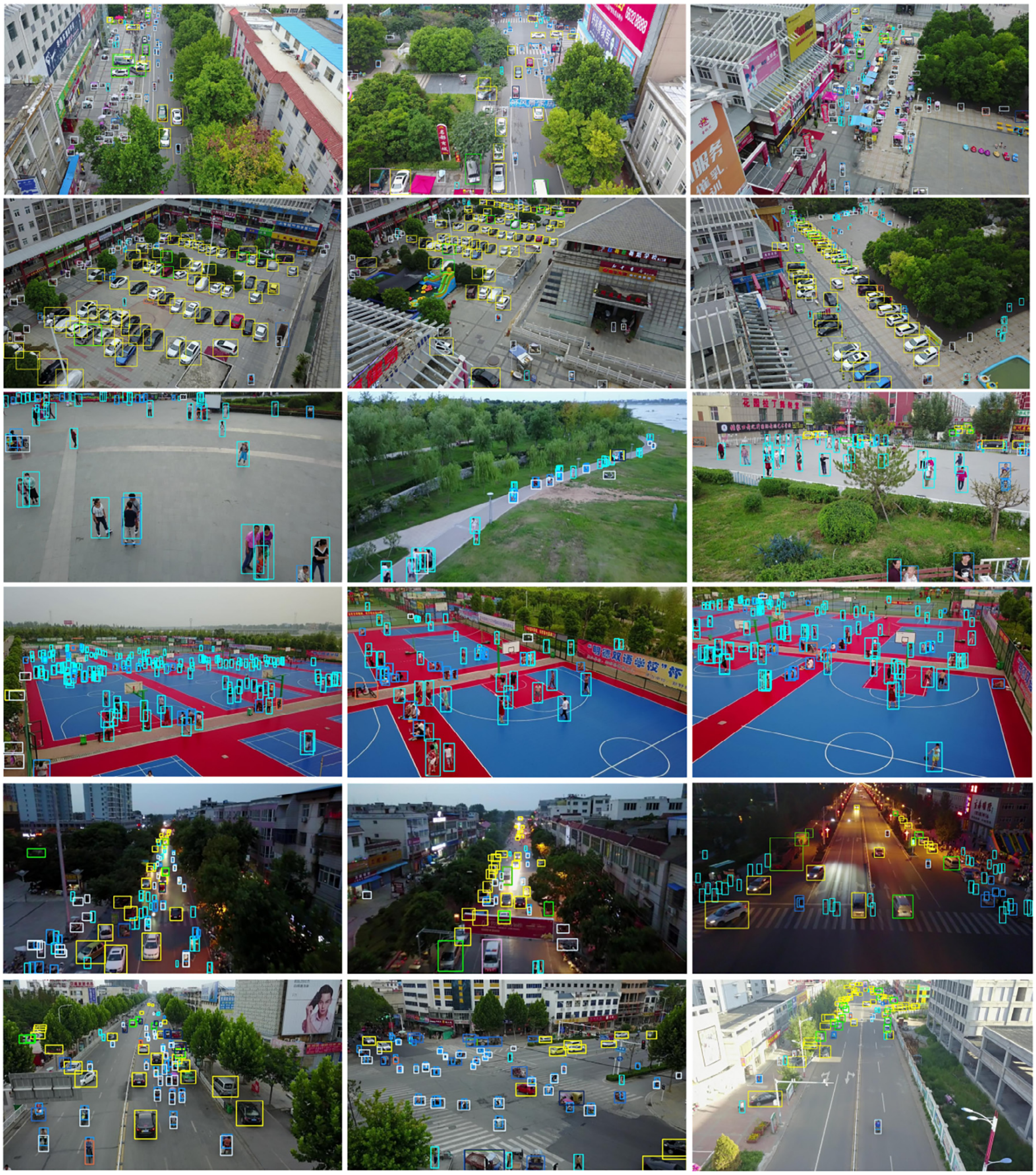


Fig. 12. Visualization results on VisDrone.

7. Conclusion

In the paper, we put forward the DCT detection problem and present a relative dataset APHID-4K for this task. To address this problem, we propose a DCTDet where a ClusRPN is designed to directly predict the object density levels of the sliding window on the image to find the densely clustered regions. In addition, we present a data generation method CCG for optimizing the ClusRPN, which is verified effective for small-scale DCT datasets. We also prove that using multiple fine detectors LDG for region chips of different densities can effectively improve the local detection performance. Experimental results demonstrate that DCTDet significantly improves the performance of other popular detectors in DCT tasks. Our method also achieves state-of-the-art performance on VisDrone.

CRediT authorship contribution statement

Jianming Du: Conceptualization, Methodology, Software, Data curation, Writing – original draft. **Liu Liu:** Conceptualization, Methodology, Software, Writing – review & editing. **Rui Li:** Methodology, Software, Writing – review & editing. **Lin Jiao:** Writing – review & editing. **Chengjun Xie:** Supervision, Writing – review & editing. **Rujing Wang:** Resources, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under grant No.32171888 and No.31671586, the Youth Foundation of Natural Science Foundation of Anhui Province under grant No. 1908085QE202.

References

[1] Y. Bai, Y. Zhang, M. Ding, B. Ghanem, Sod-mtgan: Small object detection via multi-task generative adversarial network, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 206–221.

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

[3] C. Chen, Y. Zhang, Q. Lv, S. Wei, X. Wang, X. Sun, J. Dong, Rrnet: A hybrid detector for object detection in drone-captured images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019.

[4] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 764–773.

[5] C. Deng, M. Wang, L. Liu, Y. Liu, Y. Jiang, Extended feature pyramid network for small object detection, IEEE Trans. Multimedia (2021).

[6] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, Q. Tian, The unmanned aerial vehicle benchmark: Object detection and tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 370–386.

[7] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable object detection using deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 2147–2154.

[8] M. Gao, R. Yu, A. Li, V.I. Morariu, L.S. Davis, Dynamic zoom-in network for fast object detection in large images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6926–6935.

[9] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[10] Y. Gong, X. Yu, Y. Ding, X. Peng, J. Zhao, Z. Han, Effective fusion factor in fpn for tiny object detection, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1160–1168.

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

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

[13] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012) 1097–1105.

[14] R. LaLonde, D. Zhang, M. Shah, Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4003–4012.

[15] C. Li, T. Yang, S. Zhu, C. Chen, S. Guan, Density map guided object detection in aerial images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 190–191.

[16] R. Li, R. Wang, C. Xie, L. Liu, J. Zhang, F. Wang, W. Liu, A coarse-to-fine network for aphid recognition and detection in the field, Biosyst. Eng. 187 (2019) 39–52.

[17] J.-S. Lim, M. Astrid, H.-J. Yoon, S.-I. Lee, Small object detection using context and attention, in: 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). IEEE, 2021, pp. 181–186.

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

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

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

[21] L. Liu, R. Wang, C. Xie, P. Yang, F. Wang, W. Liu, Deep learning based automatic multi-class wild pest monitoring approach using hybrid global and local activated features with stationary trap devices, IEEE Trans. Industr. Inf. 99 (2020).

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

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision. Springer, 2016, pp. 21–37.

[24] Z. Liu, G. Gao, L. Sun, Z. Fang, Hrdnet: High-resolution detection network for small objects, in: 2021 IEEE International Conference on Multimedia and Expo (ICME), IEEE Computer Society, Los Alamitos, CA, USA, 2021, pp. 1–6. doi:10.1109/ICME51207.2021.9428241.

[25] Y. Lu, T. Javidi, S. Lazebnik, Adaptive object detection using adjacency and zoom prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2351–2359.

[26] T. Moranduzzo, F. Melgani, Detecting cars in uav images with a catalog-based approach, IEEE Trans. Geosci. Remote Sens. 52 (10) (2014) 6356–6367.

[27] A. Neubeck, L. Van Gool, Efficient non-maximum suppression, in: 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, IEEE, 2006, pp. 850–855..

[28] A. Newell, K. Yang, J. Deng, Stacked hourglass networks for human pose estimation, in: European conference on computer vision. Springer, 2016, pp. 483–499.

[29] J. Noh, W. Bae, W. Lee, J. Seo, G. Kim, Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9725–9734.

[30] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, D. Lin, Libra r-cnn: Towards balanced learning for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 821–830.

[31] Y. Qing, D.-X. Xian, Q.-J. Liu, B.-J. Yang, G.-Q. Diao, T. Jian, Automated counting of rice planthoppers in paddy fields based on image processing, J. Integr. Agric. 13 (8) (2014) 1736–1745.

[32] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[33] J. Redmon, A. Farhadi, Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018..

[34] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149.

[35] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 658–666.

[36] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014..

[37] B. Singh, M. Najibi, L.S. Davis, Sniper: Efficient multi-scale training. arXiv preprint arXiv:1805.09300, 2018..

[38] M. Tan, R. Pang, Q.V. Le, Efficientdet: Scalable and efficient object detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[39] Z. Tang, X. Liu, B. Yang, Penet: Object detection using points estimation in high definition aerial images, in: 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2020, pp. 392–398..

[40] K. Thenmozhi, U.S. Reddy, Crop pest classification based on deep convolutional neural network and transfer learning, Comput. Electron. Agric. 164 (2019) 104906.

[41] Z. Tian, C. Shen, H. Chen, T. He, Fcos: Fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9627–9636.

[42] F. Wang, R. Wang, C. Xie, P. Yang, L. Liu, Fusing multi-scale context-aware information representation for automatic in-field pest detection and recognition, Computers and Electronics in Agriculture 169 (2020) 105222.

[43] J. Wang, W. Yang, H. Guo, R. Zhang, G.-S. Xia, Tiny object detection in aerial images, in: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 3791–3798.

[44] X. Wang, S. Zhang, Z. Yu, L. Feng, W. Zhang, Scale-equalizing pyramid convolution for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13359–13368.

[45] Y. Wang, Y. Yang, X. Zhao, Object detection using clustering algorithm adaptive searching regions in aerial images, in: European Conference on Computer Vision. Springer, 2020, pp. 651–664.

[46] C. Wen, D. Guyer, Image-based orchard insect automated identification and classification method, Comput. Electron. Agric. 89 (2012) 110–115.

[47] X. Wu, C. Zhan, Y.-K. Lai, M.-M. Cheng, J. Yang, Ip102: A large-scale benchmark dataset for insect pest recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 8787–8796.

[48] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, L. Zhang, Dota: A large-scale dataset for object detection in aerial images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3974–3983.

[49] C. Xie, R. Wang, J. Zhang, P. Chen, W. Dong, R. Li, T. Chen, H. Chen, Multi-level learning features for automatic classification of field crop pests, Compute. Electron. Agric. 152 (2018) 233–241.

[50] C. Xu, J. Wang, W. Yang, L. Yu, Dot distance for tiny object detection in aerial images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1192–1201.

[51] F. Yang, H. Fan, P. Chu, E. Blasch, H. Ling, Clustered object detection in aerial images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8311–8320.

[52] Y. Yuan, H. Ning, X. Lu, Bio-inspired representation learning for visual attention prediction, IEEE Trans. Cybern. (2019)..

[53] J. Zhang, J. Huang, X. Chen, D. Zhang, How to fully exploit the abilities of aerial image detectors, in: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019.

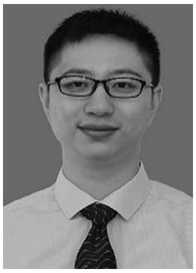
[54] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-iou loss: Faster and better learning for bounding box regression, Proceedings of the AAAI Conference on Artificial Intelligence. 34 (2020) 12993–13000.

[55] Zhou, X., Wang, D., Krähenbühl, P., 2019. Objects as points. arXiv preprint arXiv:1904.07850..

[56] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, H. Ling, Vision meets drones: Past, present and future. arXiv preprint arXiv:2001.06303, 2020..

[57] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, H. Wu, Q. Nie, H. Cheng, C. Liu, et al., Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018.

Jianming Du received his M.S. and Ph.D. degrees in information technology and signal processing from Bauman Moscow State Technical University, Russia in 2014 and 2020, respectively. He is currently a post-doctor in Hefei Institutes of Physical Science, Chinese Academy Sciences. His current research interest is deep learning and computer vision.



Liu Liu received his M.S. degree in advanced computer science from University of Manchester, Manchester, United Kingdom in 2016 and his Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei in 2020. He is currently a post-doctor in Department of Computer Science of Shanghai Jiao Tong University, China. His current research interest is computer vision and intelligent robotic.

Rui Li received his M.S. degree in computer applied technology from Hefei University of Technology, Hefei, China in 2013, his Ph.D. degree in electronic information from University of Science and Technology of China, Hefei in 2021. He is currently a post-doctor in Hefei Institutes of Physical Science, Chinese Academy Sciences. His current research interest is deep learning and computer vision. Biography of the author(s) Click here to access/download;Biography of the author(s);bios-DCTDet.pdf

Lin Jiao received her M.S. degree at the College of Mechanical and Electronic Engineering, Northwest A&F University, Shaanxi, China in 2018, and her Ph.D. degree from University of Science and Technology of China, Hefei in 2021. Her current research interests include image processing and computer vision.

Chengjun Xie received his M.S. degree in software engineering from the Hefei University of Technology, Hefei, China, in 2008, and Ph.D. degree in image processing from in the Hefei University of Technology, Anhui, China, in 2011. He is currently working in the Institute of Intelligent Machinery of the Chinese Academy of Sciences as Associate Researcher. His research interests include crop disease and pest image recognition, agricultural big data, agricultural Internet of Things.

Rujing Wang received the B.E. degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 1987, and M.S. degree in electronic engineering from Dalian University of Technology, Dalian, China, in 1990, and Ph.D. degree in pattern recognition and intelligent system from University of Science and Technology of China, Hefei, China, in 2005. He is currently working with the Institute of Intelligent Machinery of the Chinese Academy of Sciences as Professor and Researcher. His main research interests include intelligent agriculture, agricultural internet of things, Agricultural knowledge engineering.