A New Dataset and Comparative Study for Aphid Cluster Detection

Tianxiao Zhang[1], Kaidong Li[1], Xiangyu Chen[1], Cuncong Zhong[1], Bo Luo[1], Ivan Grijalva Teran[2], Brian McCornack[2], Daniel Flippo[3], Ajay Sharda[3], Guanghui Wang*[4]

[1]Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA
[2]Department of Entomology, Kansas State University, Manhattan, KS 66506, USA
[3]Department of Biological and Agricultural Engineering, Kansas State University, Manhattan, KS 66506, USA
[4]Department of Computer Science, Toronto Metropolitan University, Toronto, ON M5B 2K3, Canada
Corresponding author: Guanghui Wang (wangcs@torontomu.ca)

Abstract

Aphids are one of the main threats to crops, rural families, and global food security. Chemical pest control is a necessary component of crop production for maximizing yields, however, it is unnecessary to apply the chemical approaches to the entire fields in consideration of the environmental pollution and the cost. Thus, accurately localizing the aphid and estimating the infestation level is crucial to the precise local application of pesticides. Aphid detection is very challenging as each individual aphid is really small and all aphids are crowded together as clusters. In this paper, we propose to estimate the infection level by detecting aphid clusters. We have taken millions of images in the sorghum fields, manually selected 5,447 images that contain aphids, and annotated each aphid cluster in the image. To use these images for machine learning models, we crop the images into patches and created a labeled dataset with over 151,000 image patches. Then, we implement and compare the performance of four state-of-the-art object detection models.

Introduction

Annually 37% of crops are lost to pest damage and around 13% of crop damage is caused by insects. Most farmers consider utilizing pesticides to eliminate insects and a tremendous amount of funding were applied to pesticides each year. While most of the pesticides are wasteful since only a small portion of used pesticides is employed directly on the insects and most of them are wasted and even pollute the environment. Under several management scenarios, only a small fraction of areas receive a justified amount of pesticide, while other areas lose yield due to delayed timing and damage by pests, and remaining areas receive a superfluous spray application when there is no pest presence. However, the development of robotic technology for insecticide application has not been explored primarily due to unavailable camera vision capabilities to locate the pest incidence and severity within a complicated crop canopy. There is an urgent demand for an intelligent application system designed to accurately spray on the infested canopy but only where infestations are present.

Object detection and recognition is one of the most critical components in agricultural robotics, and detecting small insects like aphids can be especially challenging. Convolutional neural network (CNN) was first used in (Girshick et al. 2014) for object detection and recognition. The CNN models have a wide range of applications on medical image analysis (Li et al. 2021a) and object detection (Ma, Zhang, and Wang 2021)(Zhang et al. 2020b). Nonetheless, aphids are so tiny that even state-of-the-art detection models could not accurately localize them individually. Most existing aphid detection models (Teng et al. 2022)(Li et al. 2019b)(Li et al. 2019a) attempt to detect individual aphids, but the performance of the detection models on the aphid dataset is still not that pleasant. In addition, those models are trained on ideal aphid images, it is even harder for them to detect the aphid in real-world scenarios since most aphids are clustered together on the leaves, and accurately dividing and detecting the dense aphids individually is almost impossible. In addition, different illuminations and shades in different images might cause domain shifts which also severely affect the accuracy of CNN models on the tiny aphids detection (Yang et al. 2022)(Zhang, Ma, and Wang 2021).

In this paper, we collected millions of images from the sorghum field over two seasons, and then manually selected

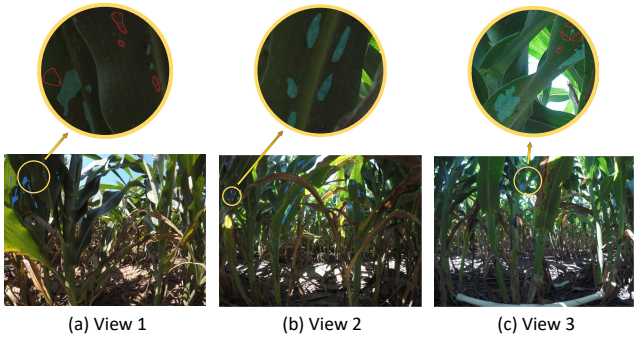
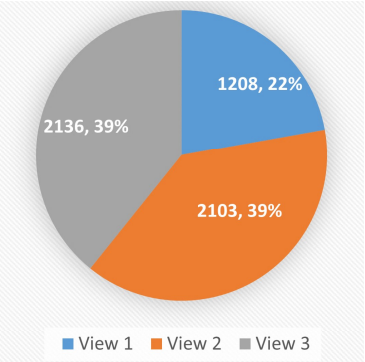


Figure 1: Original images with annotations. The light blue areas represent an annotated aphid cluster. Aphid clusters are mostly tiny compared to the original image size. In (a) and (c), the areas circled by red lines represent clusters we do not need to annotate since those areas only contain very few sparsely distributed aphids. The criteria is an aphid cluster should have more than 6 closely located aphids.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.



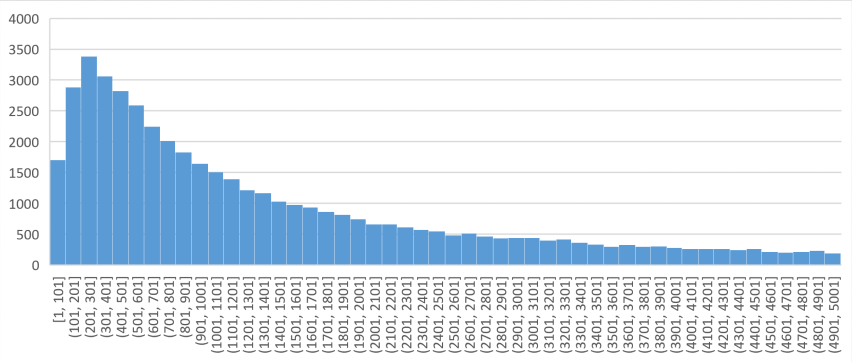


Figure 2: Statistical summary of the created dataset. (left) Dataset composition; (right) mask size distribution.

and annotated 5,447 images affected by aphids. Instead of labeling the aphid individually, we propose annotating aphid colonies as clusters and creating the bounding boxes based on the aphid clusters. Because bounding boxes are always rectangles and cover larger areas, closely located bounding boxes are merged together without affecting aphid detection. In addition, we implemented and evaluated the performance of four state-of-the-art detection models on the generated aphid dataset. The study makes it possible to estimate the aphid affection levels from real images to assist farmers to make timely control of affection. The labeled dataset and developed learning models will be freely accessible to the research community on the author's homepage.

Dataset Generation

Data Collection

Most of the aphids are located under the leaves and majority of them are densely clustered. In order to reduce the influence of occlusion among the sorghum leaves, we developed an imaging rig with three GoPro cameras that can capture the canopy leaves at three different heights corresponding to view 1, view 2, and view 3, respectively. Thus, we can enrich the dataset by taking pictures at three different views and capturing aphid clusters from different perspectives. Three sample annotated images corresponding to the three views are shown in Fig 1. Using this device, we have captured millions of images over two growing seasons of a sorghum farm. Most of the images are free of aphids. We manually examine all images and eliminate those without aphids, and eventually, select 5,447 images that contain adequate aphids. The percentages of photos corresponding to the three views are shown in Fig 2.

Data Labeling

The aphid clusters in the selected images are manually annotated by professionally trained researchers using Labelbox[1]. We first create segmentation masks for each image and then generate detection bounding boxes based on the masks. In total, we have labeled 59,767 aphid clusters.

Aphid Cluster Definition. Data labeling is a labor-intensive process. The task is distributed among 8 trained research assistants. Therefore, it is crucial to have an efficient and consistent definition of what is an aphid cluster before labeling. Ambiguous criteria will confuse deep learning models during training.

In the fields, the aphid clusters can appear in a variety of patterns (low density, high density, different sparsity) as shown in Fig 1. If we aim to label each individual aphid regardless of its density, it will take an excessive amount of time and resources will be wasted on areas without critical threat. If the threshold is set too high, areas with substantial aphid infestation might be ignored, resulting in financial loss. After discussions with agricultural experts, we define the aphid cluster as "an area with more than or equal to six closely located aphids". A further interpretation of the threshold is demonstrated in Fig 1.

Labeling. After removing some redundant images and images without aphid clusters, we have labeled in total 5,447 photos. Redundant images are those taken from very close viewpoints, resulting in visually similar images. Photos without clusters are removed because we believe deep CNN models can learn sufficient negative features from empty spaces of the remaining photos.

In summary, the statistical information of the generated masks is shown in Fig 2. In total, 59,767 masks are created and the sizes of masks vary greatly. 77.0% of the masks have a size smaller than 5,000 pixels. Since masks with larger sizes are rare and sparsely distributed, we only plot the histogram of masks with less than 5,000 pixels in Fig 2b. More than half of the masks are smaller than 1,500 pixels, with the most popular size interval [201, 301]. Among all masks, the median size is 1,442 pixels and the mean is 7,867 pixels. The median is more representative, while the mean value is severely affected by the extremely large masks.

10-Fold Cross Validation. Cross validation (Stone 1974) is a resampling method to evaluate and pick models on a small dataset. Popular computer vision datasets commonly have more than $10k$ images, *e.g.,* MS COCO (Lin et al. 2014) has more than $200k$ labeled images. Our dataset only has a little over $5k$ images. Following cross validation (Stone 1974), we decide to split our dataset into 10 groups. To ensure each group has a similar percentage of images from the

[1]https://labelbox.com/

Table 1: Number of patches in each group

Sum	1	2	3	4	5	6	7	8	9	10
151,380	14,778	15,392	14,567	15,720	15,943	14,929	15,272	15,276	14,140	15,363

three different views, we separately shuffle the images and split them into 10 subgroups from each view. Then the final cross validation groups are formed by picking one subgroup from each view. Thus images from each view will be evenly distributed in each group.

Image Patches. The majority of the masks, as shown in Fig 2, have a size smaller than 1,500 pixels, which is less than 0.015% of the original image size ($3,648 \times 2,736$). In addition, most detection and segmentation models are trained and tested on much smaller images. So we crop the original high-definition images into smaller 400 patches. During this process, some masks will be separated into different patches and will have some exclusions. To ensure each mask's completeness in at least one of the final patches, the patch generation is performed with 50% overlapping, meaning the next patch overlaps 50% with the previous patch both horizontally and vertically. An original $3,648 \times 2,736$ image will generate 221 patches for detection and segmentation.

Patch generation is conducted after dividing the dataset into 10 cross validation groups, such that information from one original photo will not leak to any other groups. Also after patch generation, those patches without an aphid cluster are discarded because CNN models should have enough negative samples just from the background of other patches. In summary, the number of patches in each cross validation group is shown in Table 1.

Bounding Box Merge. Since we label the aphids based on clusters, small clusters close to each other are labeled individually with well-defined boundaries. However, the generated bounding boxes of these clusters overlap with each other, as shown in Figure 3. From the object detection point of view, these bounding boxes should be merged as they all represent aphid clusters. Otherwise, they may cause confusion during learning. In our application, we merge the bounding boxes of the clusters if their closest distance is less than or equal to 10 pixels. Our experiments show that this process will greatly boost detection accuracy.

Tiny Cluster Removal. The process of image cropping may create some extremely small clusters, and most of them are around the border or corner of the patches. In practice, these small labels are meaningless for model training and affection estimation. Thus, we remove the small cluster masks whose areas are less than 1% of the patch. The results after merging and removal are illustrated in Table 2.

Object Detection Models

In object detection, both classification and localization are required for recognizing and localizing the objects in the videos or images. Typically, the detection models have two separate branches for classification and localization, respectively. The classification branch is similar to most classification tasks which classify the contents included by the bounding boxes. The localization branch predicts the offsets to the anchor boxes for anchor-based detection models or to the anchor points for anchor-free detection models and then the offsets would be converted to the bounding box coordinates based on the anchor boxes or anchor points for final predictions. Since the IoU thresholds are extremely important for detection models, recent detection models tend to calculate the adaptive thresholds based on the statistical properties among the samples (Zhang et al. 2020a)(Zhang et al. 2021) or compute the dynamic thresholds based on the training status (Kim and Lee 2020)(Zhang et al. 2022).

In this study, we implemented the following four state-of-the-art object detectors and evaluated their performance on aphid detection based on the created dataset. (1) ATSS (Adaptive Training Sample Selection) (Zhang et al. 2020a) calculates the adaptive IoU thresholds based on the mean and standard deviation of the IoUs between the candidate anchor boxes and the ground truth objects to select the positive samples instead of using fixed thresholds. (2) GFLV2 (Generalized Focal Loss V2) (Li et al. 2021b) utilizes statistics of bounding box distributions as the Localization Quality Estimation (LQE). Thus the high-quality bounding boxes could have a high probability to be kept instead of suppressed with the NMS (Non-Maximum Suppression) algorithm. (3) PAA (Kim and Lee 2020) dynamically divides the positive samples and negative samples using GMM (Gaussian Mixture Model) based on the classification and localization scores of the samples in a probabilistic way. (4) VFNet (Zhang et al. 2021) is based on ATSS (Zhang et al. 2020a) algorithm, but proposes IoU-aware Classification Score (IACS) as the classification soft target using the IoUs between the predicted bounding boxes and their corresponding ground truth objects. Thus high-quality predicted boundary boxes might have high scores than those low-quality boxes. In addition, star-shaped box feature representation is introduced to further refine the predicted boxes so that they could be closer to the ground truth objects.

The aforementioned detection models are state-of-the-art approaches that have excellent performance on COCO benchmark (Lin et al. 2014). Since the labels of our created dataset are based on the aphid clusters instead of the single aphid, we can directly apply them in this problem and train these models using the created dataset.

Model Training

All models exploit 0.001 as the initial learning rate with the total training epoch being 12. The initial learning rate is utilized for 9 epochs and then reduced by 10 times for the last 3 epochs. SGD (Stochastic Gradient Descent) is employed as the optimizer to optimize the model. The momentum and weight decay are 0.9 and 0.0005, respectively. The batch size is 16 and the warmup iterations are 500. The detection models are written by PyTorch with Python3 (Chen et al. 2019).

Table 2: The mean and standard deviation of 10-fold cross validation on state-of-the-art detection models

		original	+merge 10	+rm 0.01
VFNet	AP	41.9 ± 1.91	44.8 ± 1.89	58.3 ± 1.87
	recall	80.4 ± 0.90	83.7 ± 0.92	96.8 ± 0.35
GFLV2	AP	41.6 ± 1.84	44.7 ± 1.93	58.3 ± 1.90
	recall	79.2 ± 1.27	82.6 ± 1.09	96.2 ± 0.45
PAA	AP	41.2 ± 1.65	44.2 ± 1.68	58.7 ± 1.89
	recall	84.1 ± 0.89	87.6 ± 0.86	98.4 ± 0.23
ATSS	AP	41.8 ± 1.70	44.8 ± 1.85	59.0 ± 1.82
	recall	80.0 ± 1.00	83.3 ± 1.03	97.0 ± 0.28

The evaluation metric for detection models is Average Precision (AP) which computes the area under the PR curve. The PR curve plots the Precision rate versus the Recall rate for the detection models. The precision rate indicates the correctly predicted samples over the entire predicted samples. The recall rate represents the correctly predicted samples over the entire ground truth samples.

For object detection, only predicting the correct labels is not enough since we should also consider the bounding box accuracy. Typically, IoUs (Intersection over Unions) between the predicted bounding boxes and their corresponding ground truth boxes are utilized to judge the quality of the predicted boxes. Typically, IoU is calculated by the ratio of the intersection area over the union area of two bounding boxes. PASCAL VOC (Everingham et al. 2010) selects 0.5 as the IoU threshold which indicates that the detection is a success if the IoU between the predicted bounding box and the ground truth bounding box is over 0.5 if the classification label is correctly predicted. COCO (Lin et al. 2014) chooses the IoU threshold from 0.5 to 0.95 with the step of 0.05, and calculates the AP for each of the thresholds and finally averages them. In this paper, we utilize the IoU threshold from PASCAL VOC and the generated annotation files are also in xml format as PASCAL VOC (Everingham et al. 2010).

Results

The performance across the 10-fold cross validation among all detection models is illustrated in Table 2. AP (Average Precision) and Recall are recorded in the format of $mean \pm std$. The mean and standard deviation (std) are calculated across all 10-fold validations. In the first row, "original" indicates the detection models are applied to the originally labeled dataset; "+merge 10" illustrates the results after merging the clusters whose closest distance is within 10 pixels; "+rm 0.01" stands for the results after removing the small clusters whose areas are less than 1% of the patches.

We can see from Table 2 that all detection models achieve similar results in terms of average precision, however, the recall rate of PAA (Kim and Lee 2020) is slightly higher than the other detectors since the number of predicted bounding boxes of PAA is higher than that of other detection models.

Figure 3: Some examples of merging clusters that are close enough (10 pixels) to each other. The segmentation masks are utilized to illustrate the changes in the bounding boxes before and after the merging process. The white areas denote the masks for aphid clusters. The green bounding boxes in the masks are the originally labeled boxes and the red bounding boxes in the masks are the merged results. It is very challenge for the detection models to detect individual cluster in thess scenarios. Thus, they are merged to form a more meaningful cluster.

Both AP and recall rate have been greatly increased after cluster merging (+merge 10) and removal (+rm 0.01). The above results are obtained with IoU setting to 0.5. We have also tested the influence of other IoU thresholds. In general, lower IoU yields better average accuracy and vice versa. If the locations of the aphid clusters are not concerned, a lower IoU threshold may be applied.

Conclusion

In this paper, we have selected thousands of aphid-affected images from millions of images captured in the fields and annotated the aphids based on clusters instead of individual aphids for generic usage. Due to the irregular shapes and sizes of the aphid clusters, the initial annotations might not be suitable for the training of learning models. Thus, we merge the bounding boxes of the neighboring clusters and remove the extremely small clusters. We have also evaluated and compared the performance of four state-of-the-art detectors and created a baseline of aphid detection using the created dataset. The experiments have also demonstrated the effectiveness of cluster merging and removal. The created dataset and trained detection models could be used to help farmers to estimated the aphid infestation levels in the field so as to provide timely and precise pesticide application. We hope our created dataset and analysis could inspire more work on aphid detection.

Acknowledgments

This work was partly supported in part by NSERC (RGPIN-2021-04244) and USDA (2019-67021-28996).

References

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

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.

Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 580–587.

Kim, K.; and Lee, H. S. 2020. Probabilistic anchor assignment with iou prediction for object detection. In *European Conference on Computer Vision*, 355–371. Springer.

Li, K.; Fathan, M. I.; Patel, K.; Zhang, T.; Zhong, C.; Bansal, A.; Rastogi, A.; Wang, J. S.; and Wang, G. 2021a. Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations. *Plos one*, 16(8): e0255809.

Li, R.; Wang, R.; Xie, C.; Liu, L.; Zhang, J.; Wang, F.; and Liu, W. 2019a. A coarse-to-fine network for aphid recognition and detection in the field. *Biosystems Engineering*, 187: 39–52.

Li, W.; Chen, P.; Wang, B.; and Xie, C. 2019b. Automatic localization and count of agricultural crop pests based on an improved deep learning pipeline. *Scientific reports*, 9(1): 1–11.

Li, X.; Wang, W.; Hu, X.; Li, J.; Tang, J.; and Yang, J. 2021b. Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11632–11641.

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

Ma, W.; Zhang, T.; and Wang, G. 2021. Miti-detr: Object detection based on transformers with mitigatory self-attention convergence. *arXiv preprint arXiv:2112.13310*.

Stone, M. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2): 111–133.

Teng, Y.; Wang, R.; Du, J.; Huang, Z.; Zhou, Q.; and Jiao, L. 2022. TD-Det: A Tiny Size Dense Aphid Detection Network under In-Field Environment. *Insects*, 13(6): 501.

Yang, Y.; Zhang, T.; Li, G.; Kim, T.; and Wang, G. 2022. An unsupervised domain adaptation model based on dual-module adversarial training. *Neurocomputing*, 475: 102–111.

Zhang, H.; Wang, Y.; Dayoub, F.; and Sunderhauf, N. 2021. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8514–8523.

Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; and Li, S. Z. 2020a. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9759–9768.

Zhang, T.; Luo, B.; Sharda, A.; and Wang, G. 2022. Dynamic Label Assignment for Object Detection by Combining Predicted IoUs and Anchor IoUs. *Journal of Imaging*, 8(7): 193.

Zhang, T.; Ma, W.; and Wang, G. 2021. Six-channel Image Representation for Cross-domain Object Detection. In *International Conference on Image and Graphics*, 171–184. Springer.

Zhang, X.; Zhang, T.; Yang, Y.; Wang, Z.; and Wang, G. 2020b. Real-time golf ball detection and tracking based on convolutional neural networks. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2808–2813. IEEE.