Fusing Global and Local Information Network for Tassel Detection in UAV Imagery

Jianxiong Ye and Zhenghong Yu

***Abstract*—Unmanned aerial vehicles (UAVs), equipped with sensors, have made a significant impact in the field of agricultural analysis. Maize, being one of the most vital crops worldwide, is intricately linked to its yield and the growth of tassels. Leveraging UAV imagery for the automatic monitoring of maize tassels holds the potential to drive the development of intelligent maize cultivation. Current research methods, nevertheless, are limited and lack robustness. To address the challenge of tassel detection in UAV images, we propose an innovative network, termed FGLNet. This network models the backbone with a 16x down-sampling to retain richer pixel information and enhances performance by effectively fusing global and local information through weighted mechanisms. Moreover, the scarcity of tassel data presents a substantial constraint. In this article, we publicly release a new dataset, named the maize tassels detection and counting UAV (MTDC-UAV), featuring annotated bounding boxes, to advance research in the agricultural domain. Although tassel detection and counting in aerial images pose formidable challenges, our approach demonstrates remarkable accuracy in evaluations based on the MTDC-UAV dataset. It achieves a detection AP_{50} of 0.837 and a counting R_2 of 0.9409, all while maintaining a parameter count of just 0.77 M. This level of performance considerably outperforms other state-of-the-art computer vision methods. Overall, this research not only introduces innovative concepts but also provides worthwhile references and a solid data foundation for future studies.**

***Index Terms*—Computer vision, detection and counting, information fusion, maize tassel, unmanned aerial vehicle (UAV).**

I. Introduction

MAIZE, as one of the world's most important crops, serves multiple purposes, providing food, feed, and industrial raw materials, thereby exerting a profound impact on agricultural economies [1], [2]. The performance of maize yield is closely tied to the growth of tassels. Consequently, monitoring the growth status of maize tassels is critically important for agricultural activities, such as breeding, field management, phenological observations, and yield prediction [3], [4]. Traditional agricultural methods typically rely on manual labor for this task, which is subjective, time-consuming, labor-intensive, and inefficient. Fortunately, with computer vision technology advancing rapidly, crop monitoring is moving from manual to automated image processing solutions. This approach offers advantages like noninvasiveness, continuity, and intuitiveness. Some widely studied cases include the detection of wheat heads [5], counting maize tassels [6], and recognizing crop seedlings [7].

In recent years, high-performance graphics processing units (GPUs) and ever-increasing computational capabilities have had a significant impact on the analysis of maize tassels in agricultural fields using sensor-equipped unmanned aerial vehicles (UAVs). Researchers can easily access high-resolution plant growth images for automated phenotypic trait analysis. Some typical research cases include Liu et al. [8], who used ResNet as the backbone for Faster R-CNN to detect tassels in high-resolution images. Regrettably, Faster R-CNN only utilized a single layer of feature mapping, limiting detection accuracy due to feature representation and a small receptive field. To improve processing speed, Song et al. [9] embedded channel attention into the YOLOX model to detect tassels in low-altitude UAV images. Attention, however, only provided partial foreground guidance and did not directly enhance small object detection accuracy. In order to address this, Liu et al. [10] introduced an enhanced approach, YOLOv5-tassel, by fusing shallow information into BiFPN to enhance perception of small objects. Even so, its structure is complex and comes with a substantial number of parameters. During recent research, Yu et al. [11] proposed an innovative deep convolutional network, TasselLFANet, which features a concise and efficient global architecture. It includes a 16x down-sampling layer in the encoder and utilizes two feature layers in the decoder to accomplish the task of fast detection and counting in high-resolution images of densely populated tassels in the natural canopy layer. It is worth noting that using a 16x down-sampling layer in the encoder for high-altitude UAV image detection offers notable advantages due to higher spatial resolution, preserving more detailed information at each pixel position.

Detecting tassels in RGB images acquired by UAVs is a rather challenging task, especially since the objects of interest are

Manuscript received 26 October 2023; revised 1 December 2023; accepted 16 January 2024. Date of publication 22 January 2024; date of current version 6 February 2024. This work was supported in part by the 2022 key Scientific Research Project of ordinary Universities in Guangdong Province under Grant 2022ZDZX4075, in part by the 2022 Guangdong province ordinary universities characteristic innovation project under Grant 2022KTSCX251, in part by the Collaborative Intelligent Robot Production & Education Integrates Innovative Application Platform Based on the Industrial Internet under Grant 2020CJPT004, in part by the 2020 Guangdong Rural Science and Technology Mission Project under Grant KTP20200153, in part by the Engineering Research Centre for Intelligent equipment manufacturing under Grant 2021GCZX018, and in part by the GDPST&DOBOT Collaborative Innovation Center under Grant K01057060. *(Jianxiong Ye and Zhenghong Yu contributed equally to this work.) (Corresponding authors: Zhenghong Yu; Jianxiong Ye.)*

Jianxiong Ye and Zhenghong Yu are with the College of Robotics, Guangdong Polytechnic of Science and Technology, Zhuhai 519090, China, also with the School of Electronics and Information Engineering, Wuyi University, Jiangmen 529020, China, and also with the College of Engineering, South China Agricultural University, Guangzhou 510642, China (e-mail: jxye59720@gmail.com; honger1983@gmail.com).

The UAV-based dataset can be accessed at https://github.com/Ye-Sk/MTDC-UAV.

Digital Object Identifier 10.1109/JSTARS.2024.3356520

typically small in size. In the context of small object detection, low-level convolutional neural network features often exhibit higher effectiveness [12]. Nevertheless, existing UAV object detection methods often employ feature extraction networks with large down-sampling factors to obtain higher-level features [13], [14]. Inevitably, such large strides tend to compress the feature information of small objects into small points or even make them disappear in low-resolution feature maps due to pixel limitations. A common solution to this challenge is to introduce additional branches that fuse multiscale information with the output feature set [15], [16], [17]. Nonetheless, this approach is not always efficient because there are distinct semantic differences between different layers. For instance, shallow responses typically include more detailed spatial features, such as edges, colors, and textures, which are crucial for precise object localization. In contrast, high-level responses emphasize semantic information, including object categories, shapes, and abstract features. Hence, the effective detection of objects profoundly depends on the integrated use of multiscale and deep-level information [18]. While TasselLFANet may have achieved success, the importance of high-level features should not be overlooked, as they can capture more representative semantic information, aiding in understanding the context and context of the object. To fully harness the advantages of both, we introduce an innovative network architecture called FGLNet. This network focuses on modeling the backbone with a 16x down-sampling layer to preserve richer small object information. It generates a global feature set by fusing multiscale feature layers. Subsequently, this global information is fused with two critical local feature layers, facilitating the organic integration of information. Finally, the two branches fuse their outputs through the learning of nonlinear weight combinations.

On the other hand, datasets hold an extremely important position in this field. Unfortunately, there has not been a dataset available for the UAV maize tassel detection research so far. We noticed that Lu et al. [19] had annotated a maize tassels counting UAV (MTC-UAV) dataset with point annotations. Building upon their annotated points, we took the initiative to add bounding box annotations to this dataset, and we named it maize tassels detection and counting UAV (MTDC-UAV). Simultaneously, we decided to make this dataset publicly available to promote further research in the field of agricultural remote sensing.

Our main contributions include the following.

1) FGLNet: A novel convolutional network that effectively fuses modeled global information with local information to enhance performance.
2) MTDC-UAV: A dataset for the UAV-based maize tassel detection and counting with bounding box annotations.
3) In the evaluation on the MTDC-UAV dataset, we demonstrate state-of-the-art performance compared to various advanced methods.

II. Materials and Methods

A. Dataset Processing

Before proceeding with the introduction, let us review the MTC-UAV dataset. The image capture experiment was conducted at China Agricultural University during the spring sowing season of 2019, where a diverse array of over 400 maize varieties was meticulously planted. Different varieties were randomly distributed across various small plots, each replicated six times. Each microplot measured 5 m in length and 0.6 m in width. The imaging process took place under favorable conditions, capturing scenes on clear sunny, overcast, and cloudy days. The resulting dataset comprises 306 images acquired by a UAV flying at an altitude of 12.5 m, with a resolution of 5472×3648 pixels. Among these, 200 images are allocated for training, while the remaining 106 are designated for testing. Each image encompasses a range of 36–550 tassels, covering approximately one hectare of experimental farmland. The camera has a focal length of 28 mm, resulting in a ground sampling resolution of roughly 0.3 cm/pixel. The annotation process for the MTDC-UAV dataset involved the use of the LabelImg annotation tool [20]. In Fig. 1, we present an annotation example. It should be emphasized that annotating such small instances is highly challenging. Therefore, we annotated only the 200 images in the training set, while the remaining 106 images were used for the counting task. Although the annotation process was time-consuming (taking approximately one month), it proved to be meaningful and valuable. What is more, we encountered difficulties when attempting to directly train on these images. Largely, it is because the down-sampling factor of the neural network has led to the loss of detailed information regarding the tassels in the UAV imagery. A straightforward solution would be to enhance the input resolution, but this undoubtedly incurs a significant increase in computational expenditure. So for the annotated images, we employed a beneficial technique by evenly splitting each image into four parts, as illustrated in Fig. 2. This resulted in a total of 800 images. It should be noted that the annotation file for the corresponding image will also be split into four correspondingly, and the bounding box that exceeds the split size will be set back to the split boundary value. This is similar to annotating a partially obscured/visible tassel. By employing the splitting approach, the feature map preserves a greater amount of spatial information, thereby assisting in the accurate localization and capturing of intricate tassels details. Furthermore, it also proves beneficial in alleviating the demands placed on computational resources. In summary, the training set of the MTDC-UAV dataset comprises 500 segmented images, with the remaining 300 images used for testing detection performance. As for the 106 unsegmented images with point annotations, we employed them for evaluating the model's counting performance.

B. FGLNet Architecture

We noticed that Wang et al. [21] proposed a Gold-YOLO based on a gather-and-distribute mechanism. It achieves more efficient information interaction and fusion by uniformly aggregating and distributing features from different levels on a global scale. Moreover, Yu et al. [11] recently achieved advanced results in maize tassel detection and counting. Their research suggests that in agricultural scenarios, modeling the backbone network with a 16x down-sampling layer is sufficient and contributes to building a concise and efficient architecture. Inspired by these findings, we adopt an efficient strategy to fuse global

Fig. 1. Example of bounding box annotation in the MTC-UAV dataset, with reference to existing point annotations during the labeling process.

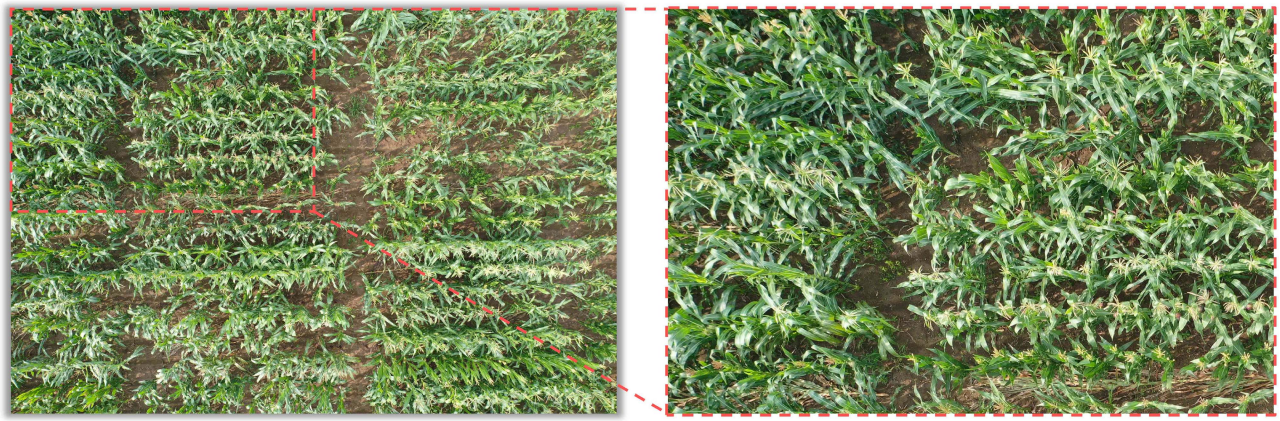

Fig. 2. Data processing, the left is the original image of the MTC-UAV, and the right is the first image split into 4 equal parts.

and local information. Our constructed FGLNet is illustrated in Fig. 3, and we will now explain its modeling ideas.

Given a 3-D tensor $I \in R^{H\times W\times 3}$ of an RGB image, we start by performing feature extraction using the CSPDarknet [22] backbone with a 16x down-sampling. This can be viewed as defining a transformation $R^{H\times W\times 3} \rightarrow R^{(H/16)\times(W/16)\times C}$ that maps the input I to a new tensor space. As shown in Fig. 3, we model only the generated C2, C3, and C4 feature layers, with down-sampling rates of 1/4, 1/8, and 1/16, respectively. Next, we build a Global-Fusion (G-Fusion) module to obtain global information. Given three different spatial-dimension feature maps x_1, x_2, x_3, and a weight coefficient W, we align C2 and C4 with intermediate layer C3 using average pooling and bilinear interpolation, mapping two feature maps to a new tensor space $R^{(H/8)\times(W/8)\times C}$. This preserves feature map sizes while avoiding excessive computational overhead. Considering that these three features from different layers contribute unequally to the output features, we assign an additional weight coefficient for each feature map to balance the semantic differences between the layers. Specifically, the weight coefficient W is defined as

$$W = \frac{w}{\varepsilon + \sum_{i=1}^{3} w_i} \tag{1}$$

w represents a learnable three-element vector, where W corresponds to the weights associated with the input data, and ε is a small constant value to prevent the denominator from being

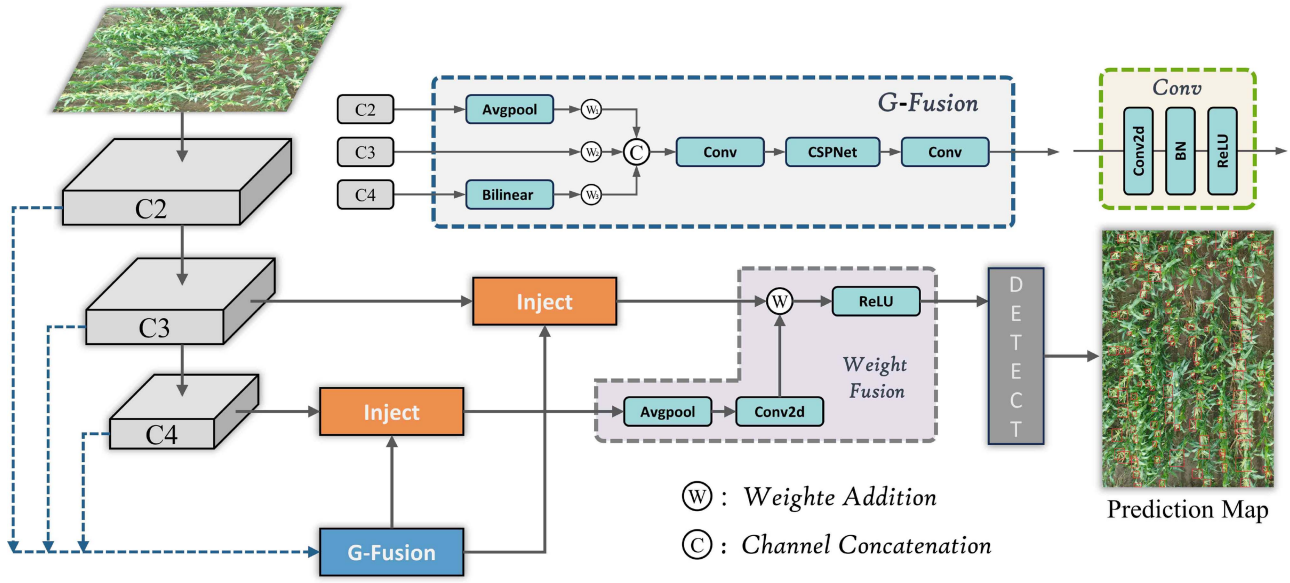

Fig. 3. Architecture of FGLNet. C2, C3, and C4 refer to feature maps sampled at 4, 8, and 16 times, respectively, with output channels of 32, 64, and 128.

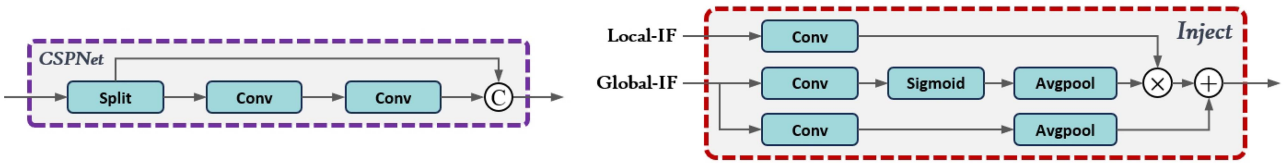

Fig. 4. Structure of CSPNet and Inject in FGLNet.

zero. Thereafter, the calculated weights are mapped back to the three feature maps, resulting in a weighted coefficient structure

$$x_1, x_2, x_3 = W_1 \cdot x_1, W_2 \cdot x_2, W_3 \cdot x_3. \tag{2}$$

Among them $W_1 + W_2 + W_3 = W$, and $W = 1$. Then, we concatenate the three feature maps along the channels and further process them to generate more advanced representations. In particular, we employ a dimension-reduction Conv followed by CSPNet [23] to enhance the feature representation. Finally, we attach another Conv to restore the channel number, with Conv consisting of 2-D convolution, batch normalization (BN) [24], and rectified linear unit (ReLU) [25]. The structure of CSPNet is illustrated in Fig. 4, which acquires richer gradient information representations by channel splitting.

After obtaining the global information through G-Fusion, the next step is to effectively fuse it with local information. Here, we employ the inject' module proposed by Wang et al. [21], as depicted in Fig. 4. Its essence lies in using attention operations to fuse information. The input to the Inject module comprises local information and global information, with the global information extracted from global features. First, the local information is processed through a local embedding layer to obtain local features. Simultaneously, the global information also undergoes embedding, including global embedding and global activation layers. Furthermore, average pooling is employed to ensure proper alignment between global and local information. Subsequently, the local features are fused with the activated global information through elementwise multiplication, followed by elementwise addition with the global embedding layer. The activation information plays a crucial role in determining the extent of global information's influence on the local information. Through the inject module, we achieve the fusion of global information with two critical local feature layers.

Afterwards, we employ weight fusion to merge the two branches and obtain the final output layer. First, the C4 branch undergoes dimension alignment through average pooling and 2-D convolution. We initialize a weight coefficient w with two elements, which is used to perform a weighted average of the feature maps y_1 and y_2 from both branches, ensuring that their sum equals 1. This process can be described as

$$y = ReLU(wt_1 \cdot y_1 + wt_2 \cdot y_2) \tag{3}$$

wt is defined as

$$wt = \frac{w}{\varepsilon + \sum_{i=1}^{2} w_i}. \tag{4}$$

Here, the weights w are learned through backpropagation and gradient descent optimization. Next, the ReLU activation function is used to enhance the nonlinear expressiveness of the output. After entering the detection layer, a separate feedforward neural network is used to perform both position regression of the object bounding boxes and category information prediction. The detector provides dense pixel-level predictions as output. Finally, nonmaximum suppression (NMS) is performed to filter

the generated prediction boxes and eliminate redundant detections.

C. Loss Function

The loss functions measure the difference between the model's predictions and the actual objects, quantifying the model's performance on the given task. In this section, we introduce the bounding box regression process of FGLNet, supervised by two loss functions.

Classification loss is used to assess the model's ability to match its category predictions with the actual category labels, aiding in accurate classification tasks. We employ binary cross-entropy loss (BCE) as a guiding metric, which is a common binary classification loss and easy to optimize. It is defined as follows:

$$L_{cls} = -\frac{1}{n}\sum_{i=1}^{n}(y_i log(p_i) + (1-y_i) log(1-p_i)) \quad (5)$$

n represents the batch size, y is the ground truth labels, and p is the model's predictions.

Localization loss is used to measure the disparity between the object detection model's position regression predictions for the object bounding boxes and the actual bounding box positions. We employ the complete intersection over union (CIoU) loss for supervision, which takes into account the overlapping area, center point distance, and aspect ratio. It is described as

$$L_{loc} = IoU - \frac{d^2}{c^2} - \alpha v \quad (6)$$

where IoU represents the intersection over union (IoU) between the predicted bounding box and the true bounding box, d represents the Euclidean distance between the center points of the object bounding box, c represents the sum of the diagonal lengths of the object bounding box and the true label bounding box, v is used to measure the similarity of aspect ratios, and α is the influence factor of v.

III. Experiments and Analysis

A. Implementation Details

To ensure the objectivity and authenticity of the results, all compared methods in our experiments are trained and tested with the same configurations. Our implementation is based on the PyTorch deep learning framework and accelerated using CUDA. During training, to reduce experimental costs, FGLNet scales down high-resolution images to 1216 pixels. The backbone uses CSPDarknet pretrained on the COCO dataset [26] for weight initialization. We use AdamW [27] as the optimizer with an initial learning rate of 0.002 and a momentum factor of 0.9, and the batch size is set to 4. 150 epochs of iterative optimization based on convergence were performed. To avoid overfitting, mosaic, random scaling, and color distortion methods were used to enhance the images. Importantly, when pretrained weights were available, we configured the weight initialization for all comparative methods to ensure they could achieve optimal performance.

TABLE I
Quantitative Results of Different Comparative Methods

Method	P	R	AP_{50}	$AP_{50\text{-}95}$
Faster R-CNN	0.327	0.273	0.161	0.043
FCOS	0.578	0.796	0.670	0.274
Yolov8	0.815	0.717	0.763	0.323
WheatLFANet	0.782	0.674	0.704	0.253
TasselLFANet	0.823	0.723	0.746	0.294
FGLNet	**0.852**	**0.797**	**0.837**	**0.403**

The best performance is in boldface.

B. Evaluation Metrics

We employ the following evaluation metrics to quantify the model's detection performance: precision (P), recall (R), as well as average precision at 50% IoU (AP_{50}) and average precision from 50% to 95% IoU (AP_{50-95}). They are expressed as follows:

$$P = \frac{TP}{TP+FP} \quad (7)$$

$$R = \frac{TP}{TP+FN} \quad (8)$$

$$AP = \int_0^1 PRd(R). \quad (9)$$

Here, TP, FP, and FN represent the numbers of true positives, false positives, and false negatives, respectively. P denotes the proportion of correctly predicted objects among all objects predicted by the model, and R represents the proportion of correctly predicted objects among all true objects. AP_{50} and AP_{50-95} provide a more precise measure of the model's localization performance.

C. Comparison With State of the Art

We compared our proposed FGLNet with five advanced computer vision methods, including Faster R-CNN [28], FCOS [29], Yolov8 [30], WheatLFANet [5], and TasselLFANet [11], and the quantitative results are presented in Table I. It is evident that our FGLNet outperforms all other methods on all metrics. Notably, in this UAV scenario, Faster R-CNN's detection performance is very suboptimal, mainly due to its output of a single and relatively small feature layer, which leads to the loss of pixel information for many small objects. FCOS tries to achieve a recall (R) close to FGLNet by regressing detection boundaries for tassels on a per-pixel basis using five dense prediction layers, but it has lower precision (P) compared to other methods. One possible reason is that, due to the more complex imaging views of maize tassels captured by the UAV, the detection results from different layers introduce noise, leading to suboptimal final accuracy upon merging, as evidenced in Yan et al. [31] study. Yolov8, as the current state-of-the-art detector in general scenes, performs well but still lags behind FGLNet, particularly in the context of tassel detection in UAV scenarios. To some

TABLE II
COUNTING RESULTS OF DIFFERENT COMPARATIVE METHODS

Method	Params	MAE	RMSE	R^2
Faster R-CNN	54.00M	103.78	134.03	0.3012
FCOS	61.00M	29.44	43.24	0.9237
Yolov8	3.00M	26.99	39.63	0.9004
WheatLFANet	**0.72M**	38.81	53.12	0.8196
TasselLFANet	3.00M	33.50	46.51	0.8761
TasselNetV2	1.01M	24.74	35.63	0.9082
FGLNet	0.77M	**17.86**	**28.34**	**0.9409**

The best performance is in boldface.

extent, this is because Yolov8 employs a larger stride to achieve a broader receptive field and higher-level features. Such a large stride can directly reduce small objects to dots or even make them disappear in deeper layers. This renders subsequent upsampling operations ineffective in recovering the lost feature information. WheatLFANet is an optimization for TasselLFANet regarding speed and exhibits good generalization in wheat heads detection but struggles with small-sized tassels. TasselLFANet is the current SOTA model for maize tassel detection and achieves performance second only to FGLNet, largely due to its ability to preserve richer detail information with 16x down-sampling layers. FGLNet's advantage lies in modeling larger feature maps and fusing global and local information, significantly enhancing accurate recognition and positioning of small tassels.

D. Counting Experiment and Visualization

In this section, we conducted a counting performance evaluation on the 106 MTDC-UAV images with point annotations. We also compared it with the specialized maize tassel counting method, TasselNetV2 [32], an extension of TasselNet [6]. The motivation of TasselNetV2 is to observe the importance of weak context to nonrigid plants and use this insight to markedly improve counting performance. We summarize the counting results for all the methods mentioned earlier in Table II, using three counting performance metrics: mean absolute error (MAE), root mean squared error (RMSE), and the coefficient of determination R^2. Their formal expression is

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|G_n - P_n| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(G_n - P_n)^2} \tag{11}$$

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(G_n - P_n)^2}{\sum_{n=1}^{N}(\bar{G}_n - P_n)^2}. \tag{12}$$

N represents the total number of images, while G_n and P_n denote the ground truth count and predicted count, respectively, for the nth image. Among these, MAE measures counting accuracy, RMSE assesses counting robustness, and R^2 reflects the goodness of fit to the data. Additionally, we also report the parameter size for each model, an important consideration for deployment. From the data in the table, it is evident that FGLNet outperforms all other methods to a great extent, with a parameter size of only 0.77 M, second only to WheatLFANet. This is vital because WheatLFANet is designed for deployment on edge devices. An additional note is that there is a significant performance gap between Faster R-CNN and other methods. In fact, there is evidence of this difference, as demonstrated in the detection tests of Table I, where Faster R-CNN has shown a noticeable inability to effectively fit the level of tassels in UAV images. This is primarily attributed to the substantial information loss in the output feature layer of Faster R-CNN. Although one possible improvement is to increase the input image resolution, the associated gains in performance come at an almost disproportionate computational cost, as the performance bottleneck is mainly constrained by the Faster R-CNN network structure. Another potentially more effective approach is to utilize a feature pyramid network (FPN) [33]. To provide a more intuitive representation of the results, we offer inference examples for the top-performing four methods in Fig. 5. The density map is generated using a Gaussian function $G(\mu, \sigma)$ with a mean of μ and a variance of σ.

From these visual examples, we can conclude that FCOS still exhibits significant disparities from ground truth results. Yolov8 often underestimates the number of objects in complex scenes, which may be attributed to its semantic ambiguity in distinguishing between object and background areas. In comparison to other methods, TasselNetV2 can only produce coarse responses, making it difficult to further diagnose exceptional cases, and this lack of interpretability can become a bottleneck for counting performance. Unlike other methods, FGLNet demonstrates strong fitting capabilities in the vast majority of scenarios, even in cases with deceptive backgrounds. It is important to note that these counting results are obtained from inference on high-altitude UAV images, each containing a large number of instances. Even for professionals, achieving precise counts in such situations remains extremely challenging.

E. Linear Regression Results

As a supplement to the aforementioned counting results, we present the linear regression results for the two methods with the best counting performance, FGLNet and TasselNetV2, in Fig. 6 to enhance the interpretability of our experiments. These curves correspond to the predicted results obtained by both models based on regression analysis, and we have also calculated the prediction bias and slope. Careful examination of the regression results reveals interesting differences. FGLNet's regression results exhibit robustness with a smaller bias and slopes that are close to the 1:1 reference line. In contrast, TasselNetV2 shows relatively poorer stability. This indicates that FGLNet provides a more consistent fit across most images. However, it is worth noting that while FGLNet appears to perform well in most cases, we must consider subtleties and outliers that may not be immediately evident in regression analysis. Further research into which specific scenarios each model excels in and where they

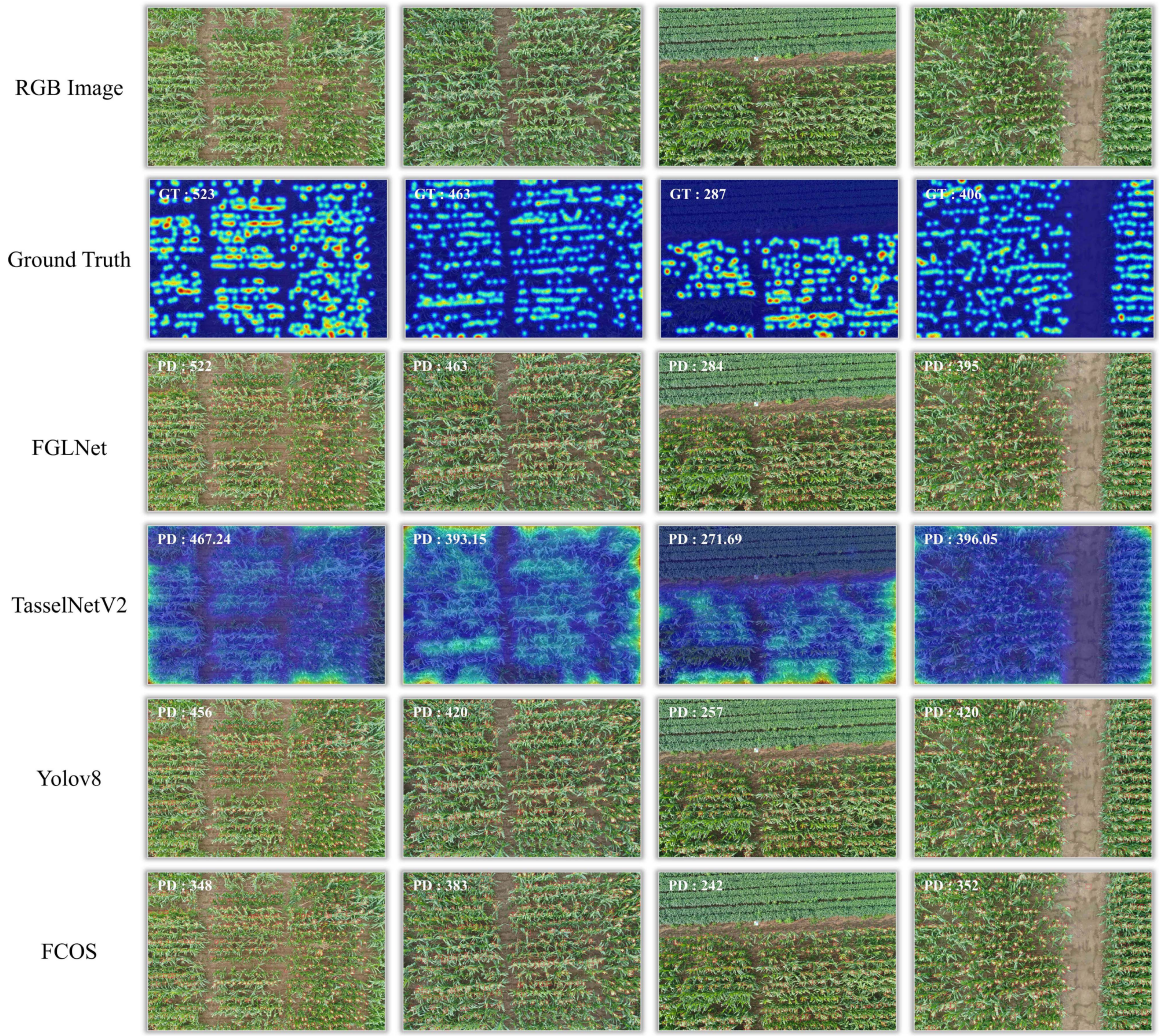

Fig. 5. Visualized results. The first row is the input RGB image, the second row is the density map, the third row is the inference result of FGLNet, the fourth row is the inference result of TasselNetV2, the fifth row is the inference result of Yolov8, and the sixth row is the inference result of FCOS. 'GT' stands for the ground truth number of tassels, 'PD' represents the predicted number of tassels. Zoom in for a better view.

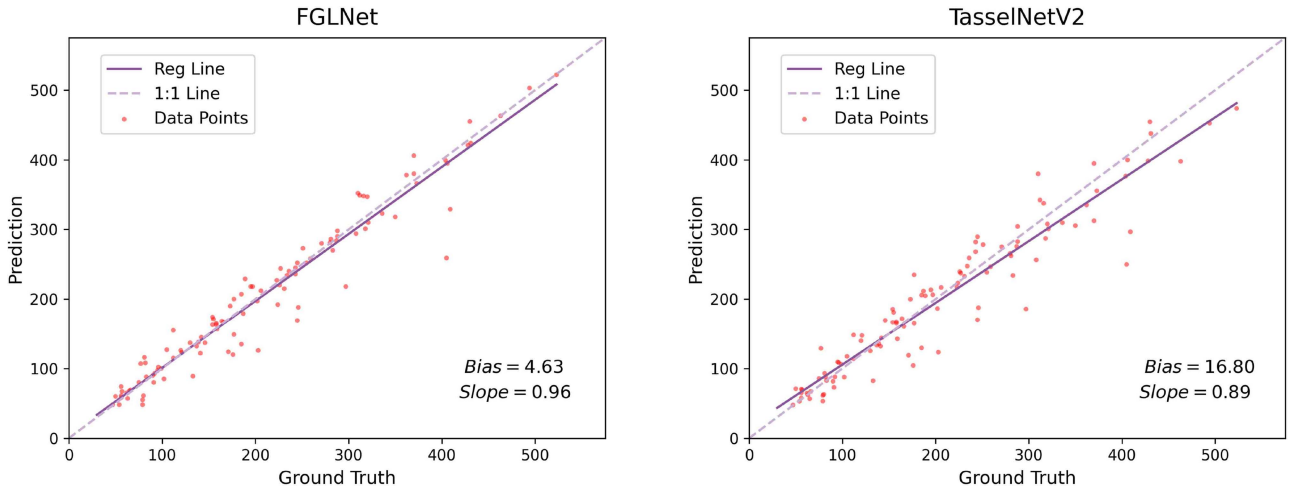

Fig. 6. Scatterplot shows the results of the two best-performing models in the counting task, with the calculation of linear regression's bias and slope in the bottom right corner.

may perform less optimally will provide a more comprehensive understanding of their counting performance.

IV. Discussions

Despite reporting promising performance, plant vision applications remain an open and unsolved problem. A pervasive issue is domain shift, where in the MTDC-UAV dataset, due to images being captured in nearly identical external environments, training and testing data adhere to similar distributions, conforming to standard machine learning assumptions. Nevertheless, when the source domain (training data) and the target domain (testing data) have different data distributions, domain shift becomes apparent. It's often observed that performance deteriorates when a well-trained model is tested in fields with different plant varieties, lighting, and weather conditions. Different image capture equipment and viewpoints exacerbate domain discrepancies. Better solutions to overcome these challenges may require customized approaches, such as domain alignment [34], self-supervised learning [35], and meta-learning [36]. At times, this also demands global collaboration among researchers, as seen in initiatives like the global wheat head detection (GWHD) [37] and diverse rice panicle detection (DRPD) [38] datasets.

Another aspect to consider is the challenge posed by high-density cultivation of crops, such as rice panicles [38] and wheat heads [39], FGLNet faces relative difficulty, primarily due to the suppression of many dense bounding boxes by NMS, leading to missed detections. This challenge is prevalent and needs to be addressed by most object detection methods. A notable example in the literature is the detection transformer (DETR) [40], which leverages the advantages of post-processing-free techniques like NMS to achieve end-to-end detection.

What is more, when contemplating practical applications, efficiency remains a critical concern. Due to the original intention of FGLNet's design sacrifices some operational speed, it may be necessary to utilize deep learning acceleration libraries, such as tensor real-time (TensorRT) and open neural network exchange (ONNX). In comparison, Yolov8 has a more straightforward high efficiency.

V. Conclusion

In this article, we have introduced an innovative approach, FGLNet, which combines global and local information to address the challenging task of maize tassel detection and counting in complex agricultural UAV scenarios. Furthermore, we have contributed the MTDC-UAV dataset, including maize tassel images annotated with bounding boxes, which serves practical purposes in the field. In the evaluation of detection and counting using the MTDC-UAV dataset, our method has demonstrated superior performance compared to other state-of-the-art computer vision methods. In conclusion, this study provides beneficial technical insights and plays a crucial role in advancing maize tassel detection and counting for future UAV applications.

Data Availability: The MTDC-UAV dataset and other supporting materials are available online at: https://github.com/Ye-Sk/MTDC-UAV

Acknowledgment

The authors would like to thank Kangqing Pan, Rujun Wu, and Yueming Wu for their help in annotating the MTDC-UAV dataset.

References

[1] M. Ye, Z. Cao, and Z. Yu, "An image-based approach for automatic detecting tasseling stage of maize using spatio-temporal saliency," in *Proc. SPIE - Int. Soc. Opt. Eng.*, 2013, vol. 89210Z, pp. 235–242.
[2] Y. Su et al., "Evaluating maize phenotype dynamics under drought stress using terrestrial LiDAR," *Plant Methods*, vol. 15, no. 11, pp. 11–26 2019.
[3] Z. Yu et al., "Automatic image-based detection technology for two critical growth stages of maize: Emergence and three-leaf stage," *Agricultural Forest Meteorol.*, vol. 174-175, pp. 65–84, 2013.
[4] J. Huang et al., "Assimilation of remote sensing into crop growth models: Current status and perspectives," *Agricultural Forest Meteorol.*, vol. 276-277, 2019, Art. no. 107609.
[5] J. Ye et al., "Wheatlfanet: In-field detection and counting of wheat heads with high-real-time global regression network," *Plant Methods*, vol. 19, no. 1, pp. 103–118, 2023.
[6] H. Lu, Z. Cao, Y. Xiao, B. Zhuang, and C. Shen, "TasselNet: Counting maize tassels in the wild via local counts regression network," *Plant Methods*, vol. 13, no. 1, pp. 79–95, 2017.
[7] L. Quan et al., "Maize seedling detection under different growth stages and complex field environments based on an improved faster r-CNN," *Biosyst. Eng.*, vol. 184, pp. 1–23, 2019.
[8] Y. Liu, C. Cen, Y. Che, R. Ke, Y. Ma, and Y. Ma, "Detection of maize tassels from UAV RGB imagery with faster r-CNN," *Remote Sens.*, vol. 12, no. 2, pp. 338–350, 2020.
[9] C.-Y. Song et al., "Detection of maize tassels for UAV remote sensing image with an improved YOLOX model," *J. Integrative Agriculture*, vol. 22, no. 6, pp. 1671–1683, 2022.
[10] W. Liu, K. Quijano, and M. M. Crawford, "YOLOv5-tassel: Detecting tassels in RGB UAV imagery with improved YOLOv5 based on transfer learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 8085–8094, 2022.
[11] Z. Yu, J. Ye, C. Li, H. Zhou, and X. Li, "TassellfaNet: A novel lightweight multibranch feature aggregation neural network for high-throughput image-based maize tassels detection and counting," *Front. Plant Sci.*, vol. 14, pp. 1–17, 2023.
[12] J. Huang, Y. Shi, and Y. Gao, "Multi-scale faster-RCNN algorithm for small object detection," *J. Comput. Res. Develop.*, vol. 56, pp. 319–327, 2019.
[13] A. Alzadjali et al., "Maize tassel detection from UAV imagery using deep learning," *Front. Robot. AI*, vol. 8, pp. 1–15, 2021.
[14] J. Li et al., "Automatic rape flower cluster counting method based on low-cost labeling and UAV-RGB images," *Plant Methods*, vol. 19, no. 40, 2023.
[15] H. Zhao, K. Chu, J. Zhang, and C. Feng, "Small-size target detection in remotely sensed image using improved multi-scale features and attention mechanism," *IEEE Access*, vol. 11, pp. 56703–56711, 2023.
[16] Y. Liu, F. Yang, and P. Hu, "Small-object detection in UAV-captured images via multi-branch parallel feature pyramid networks," *IEEE Access*, vol. 8, pp. 145740–145750, 2020.
[17] D. Lu, J. Ye, Y. Wang, and Z. Yu, "Plant detection and counting: Enhancing precision agriculture in UAV and general scenes," *IEEE Access*, vol. 11, pp. 116196–116205, 2023.
[18] J. Ye, Z. Yu, Y. Wang, D. Lu, and H. Zhou, "PlantbicNet: A new paradigm in plant science with bi-directional cascade neural network for detection and counting," *Eng. Appl. Artif. Intell.*, vol. 130, 2024, Art. no. 107704.
[19] H. Lu, L. Liu, Y.-N. Li, X.-M. Zhao, X.-Q. Wang, and Z.-G. Cao, "TasselNetV3: Explainable plant counting with guided upsampling and background suppression," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 4700515.
[20] D. Tzutalin, "A graphical image annotation tool to label object bounding boxes in images." 2022. [Online]. Available: https://github.com/tzutalin/labelImg
[21] C. Wang et al., "Gold-YOLO: Efficient object detector via gather-and-distribute mechanism," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–19.

[22] A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[23] C. Y. Wang, H. Liao, I. H. Yeh, Y. H. Wu, P. Y. Chen, and J. W. Hsieh, "CspNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 1571–1580.
[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, pp. 448–456, 2015.
[25] B. Xu, N. Wang, and T. E. A. Chen, "Empirical evaluation of rectified activations in convolutional networks," *Comput. Sci.*, 2015, *arXiv:1505.00853*.
[26] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[27] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017, *arXiv:1711.05101*.
[28] S. Ren, K. He, and R. E. A. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[29] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635.
[30] G. Jocher, "Yolov8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
[31] J. Yan, J. Zhao, and Y. E. A. Cai, "Improving multi-scale detection layers in the deep learning network for wheat spike detection based on interpretive analysis," *Plant Methods*, vol. 19, no. 1, pp. 46–58, 2023.
[32] H. Lu and Z. Cao, "Tasselnetv2: A fast implementation for high-throughput plant counting from high-resolution RGB imagery," *Front. Plant Sci.*, vol. 11, pp. 1–15, 2020.
[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.
[34] Y. Ganin et al., "Domain-adversarial training of neural networks," in *Proc. Adv. Comput. Vis. Pattern Recognit.*, 2017, pp. 189–209.
[35] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2070–2079.
[36] J. Casebeer, N. J. Bryan, and P. Smaragdis, "Meta-AF: Meta-learning for adaptive filters," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 355–370, 2023.
[37] E. David et al., "Global wheat head detection 2021: An improved dataset for benchmarking wheat head detection methods," *Plant Phenomics*, vol. 2021, pp. 1–9, 2021.
[38] Z. Teng et al., "Panicle-cloud: An open and AI-powered cloud computing platform for quantifying rice panicles from drone-collected imagery to enable the classification of yield production in rice," *Plant Phenomics*, vol. 5, pp. 1–12, 2023.
[39] Y. Zhu, Z. Cao, H. Lu, Y. Li, and Y. Xiao, "In-field automatic observation of wheat heading stage using computer vision," *Biosyst. Eng.*, vol. 143, pp. 28–41, 2016.
[40] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. Lecture Notes Comput. Sci.*, vol. 12346, pp. 213–229, 2020.

Jianxiong Ye is currently working toward B.S. degree in electronic and information engineering with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, China.

He is a Researcher with the College of Robotics, Guangdong Polytechnic of Science and Technology, Zhuhai, China. His research interests include computer vision, pattern recognition, and intelligent robotics and addressing challenges in few-shot learning, remote sensing image analysis, and various computer vision in agriculture problems.

Mr. Ye was the recipient of the first prize for his most recent agricultural research project at the prestigious Chinese Robotics and Artificial Intelligence Competition (CRAIC).

Zhenghong Yu received the B.S. and M.S. degrees in computer science from the Wuhan Institute of Technology, Wuhan, China, in 2005 and 2008, respectively, and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, in 2014.

He is currently an Associate Professor with the College of Robotics, Guangdong Polytechnic of Science and Technology, Zhuhai, China. He has been invited as a Guest Professor with the Mahanakorn Institute of Innovation, Bangkok, Thailand, Hubei Provincial Laboratory of Intelligent Robot, Hubei, China, and South China Agricultural University, Guangzhou, China, while also a Distinguished Research Fellow with Fujian Agriculture and Forestry University, Fuzhou, China. His research interests include computer vision, intelligent robots, and agriculture automation.

Dr. Yu was the recipient of the Chinese Digital Craftsman of the Year for 2023.