A Lightweight Neural Network for Accurate Rice Panicle Detection and Counting in Field Conditions

Wenchao Xu[1], Yangxu Wang[2]
School of Electrical and Computer Engineering, Nanfang College Guangzhou, Conghua 510970, China[1]
Department of Network Technology, Software Engineering Institute of Guangzhou, Conghua 510990, China[2]

***Abstract*—Monitoring rice spikelet yield is crucial for ensuring food security, but manual observations are tedious and subjective. Deep learning approaches for automated counting often require high device resources, limiting their applicability on low-cost edge devices. This paper presents the Rice Lightweight Feature Detection Network (RLFDNet). RLFDNet designed for the field of computer vision, features a lightweight encoder and decoder, effectively decoding shallow and deep information within its neural network architecture. Innovative designs including dense feature pyramid network, reinforcement learning guidance, attention mechanisms, dynamic receptive field adjustment, and shape feature fusion enable outstanding performance in object detection and counting, even with low-resolution images. Across different elevations, ranging from 7m to 20m, RLFDNet demonstrates significantly superior accuracy and inference efficiency compared to other advanced object detection methods. With a parameter count of only 4.40 million, it achieves an impressive frame rate of 80.43 FPS on a GTX1080Ti GPU, meeting real-time application requirements on inexpensive devices. RLFDNet's exceptional performance is further highlighted by an MAE of 1.86 and an R^2 of 0.9461, along with an average precision of mAP@0.5 reaching 0.91. These results underscore RLFDNet's capability as a potent and reliable visual tool for agricultural practitioners, offering promising prospects for future research endeavors.**

***Keywords*—Computer vision; deep learning; lightweight; neural network architecture; remote sensing**

I. Introduction

Rice is a pivotal global crop, essential for food security, particularly for almost half of the world's population. Metrics like spikes per square meter and grain size profoundly influence cereal crop yield [1] [2]. However, accurately counting rice spikes encounters challenges due to outdoor environment complexities, including size variations, lighting conditions, and occlusion. Traditional monitoring methods hinge on manual observation [3], which is time-consuming and subjective, impacting rice quality and yield.

In recent decades, the rapid development of computer vision has made deep learning (DL) a key research field in artificial intelligence [4]. Similarly, deep learning has also been widely applied in agriculture, particularly in various agricultural information management practices [5] [6]. Many studies utilize machine learning for crop yield prediction by estimating the quantity of fruits, such as cotton [7], citrus fruits [8], sugar beets [9], and rice. In the realm of rice, various studies have been conducted: Xiong et al. [10] proposed a rice spike segmentation algorithm based on superpixel region generation, CNN, and superpixel optimization. This method effectively segments and recognizes complex rice spikes, but may involve unreasonable assumptions, such as simplification of rice spike shapes. Misra et al. [11] introduced SpikeSegNet for rice spike detection and counting, achieving an average accuracy of 95%. However, it overly relies on lighting conditions. Wang et al. [12] presented an algorithm utilizing three-dimensional point clouds for crop size estimation, particularly suited for spike counting in high-density scenarios, yet highly relies on high-quality sensor data. Shu et al. [13] proposed a rice spike detection method based on the SSD algorithm, with an average precision mAP of 38.1%. The model's accuracy still needs improvement.

Computer vision applications in agriculture, particularly in rice spike detection, have demonstrated significant potential. However, these models encounter critical issues such as low detection accuracy or lack of lightweight design, resulting in suboptimal user experiences and high entry barriers. Furthermore, due to variations in terrestrial environments, these methods are susceptible to significant errors.

Recognizing these challenges, researchers have turned to micro unmanned aerial vehicles (UAVs). Micro UAVs offer several advantages, including convenient platform setup, low operating and maintenance costs, small size, light weight, simple operation, high flexibility, and short operation cycles, making them an ideal choice for agricultural applications. Tri et al. [14] combined drones with deep learning to predict paddy field yields, marking the first use of drones for image collection and deep learning-based rice spike classification. Hayat et al. [15] proposed an unsupervised Bayesian learning-based segmentation algorithm for rice spike segments, achieving an average F1-score of 82.10%. However, they require significant computational resources and may not be suitable for real-time applications in resource-constrained environments. Reza et al. [16] introduced a method for rice yield estimation based on K-means clustering and segmentation of low-altitude UAV images. However, their method exhibits relatively low accuracy, with a relative error ranging from 6% to 33%, making it challenging to meet the requirements for automated detection of rice spike yield. In summary, further improving accuracy and efficiency is a natural and important research direction.

To address this challenge, the focus of this study is on achieving high accuracy and lightweight design in the model architecture, taking into account the deployment requirements of edge devices in the field of plant science. The proposed method for rice spike localization and counting is a deep

learning-based approach named the Rice Lightweight Feature Detection Network (RLFDNet). RLFDNet utilizes the lightweight backbone CSPDarknet [17] and further incorporates a concise and efficient encoder-decoder module to decode features from both shallow and deep layers. Unlike existing methods, RLFDNet primarily aims to overcome the recognition challenges posed by small and dense targets. It offers several advantages: Firstly, it emphasizes higher spatial resolution to retain detailed information at each pixel position. Secondly, it focuses on extracting more discriminative high-level semantic information. Specifically, the model decoder maximizes the utilization of depth-encoded feature layers generated by the encoder to capture abstract information. By employing an adaptive strategy, RLFDNet effectively restores spatial resolution and merges feature layers from non-adjacent levels. Additionally, a channel attention mechanism is introduced to suppress irrelevant pixel information at critical positions, thus alleviating the difficulty of feature extraction in dense scenes. RLFDNet achieves a balance between accuracy and computational efficiency, making it suitable for real-world implementation on resource-constrained devices, unlike existing methods that often prioritize accuracy over computational efficiency.

To validate the universality of the model design, this study utilized the Diverse Rice Panicle Detection (DRPD) dataset [18], which was publicly released by Teng et al. [19]. It is noteworthy that this dataset comprises field rice spikes captured by micro UAVs at different altitudes (7m, 12m, and 20m) and subsequently cropped. Undoubtedly, varying the altitude during capture results in different target sizes and densities of rice spikes in the images, posing a significant challenge for object detection models. Fortunately, extensive experimental results demonstrate that RLFDNet's accuracy and inference efficiency are significantly superior to other advanced object detection methods, showcasing better robustness and adaptability. RLFDNet's parameter count is only 4.40 million, and it reports an outstanding frame rate of 80.43 FPS on the affordable GTX1080Ti GPU, making it sufficient for real-time applications when deployed on inexpensive devices. The efficiency comparison is illustrated in Fig. 1.

In summary, this study makes three main contributions:

- Innovatively introduces a more precise encoder-decoder module and cleverly designs an efficient neural network structure, significantly enhancing the integration capability of image features and effectively improving feature extraction performance.
- Proposes the lightweight RLFDNet model specifically designed for the localization and counting of field rice spikes. Its lightweight architecture allows for flexible deployment on low-end edge devices, providing a novel solution for automated monitoring of rice spikes.
- Through comprehensive comparisons with mainstream object detection models, demonstrates the outstanding performance of the RLFDNet model on rice spike datasets of different scales compared to state-of-the-art methods, highlighting its significant advantages in object detection tasks.

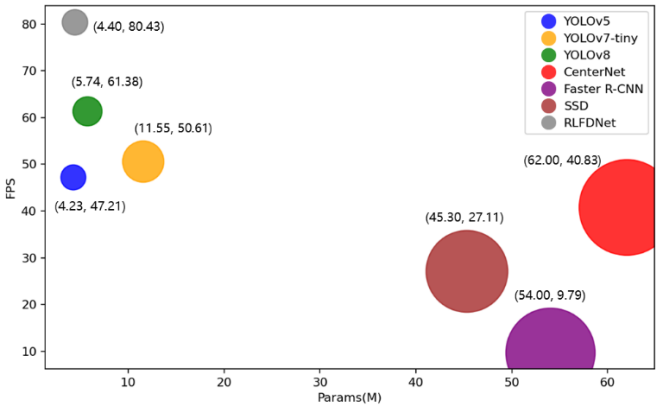

Fig. 1. Efficiency comparison of different models. Performed on a device with NVIDIA GTX1080Ti GPU (8G).

The layout of this paper is as follows:

In Section I (this section), the research background is introduced, and the problem statement is emphasized. Section II provides a detailed introduction and description of the proposed RLFDNet model. Section III conducts experiments and performs comprehensive comparative analyses with other models across various dimensions. Section IV delves into the factors influencing RLFDNet's performance and summarizes the model's innovative aspects. Section V concludes the study and outlines future research directions.

II. Materials and Methods

A. Datasets

This study is based on the publicly available dataset Diverse Rice Panicle Detection (DRPD) [19]. Aerial images of rice fields were captured at three different altitudes: GSD7m, GSD12m, and GSD20m. The images were cropped from the original aerial images, with each image having a size of 512×512 pixels. In total, 5,372 RGB sub-images were collected, annotated with 259,498 rice spikes exhibiting various morphological features. Details of the dataset are presented in Table I, where "Panicles per sub-image" indicates the number of spikes in each sub-image. The dataset includes four key growth stages: heading (1,903 sub-images), flowering (1,676 sub-images), early grain filling (1,235 sub-images), and middle grain filling (558 sub-images). It is noteworthy that, due to cropping by researchers and the high density of the aerial images, the difficulty varies across different altitudes. Rice spikes are larger and less dense at an altitude of 7m, presenting the lowest difficulty. In contrast, at an altitude of 20m, rice spikes are smaller, more densely distributed, and pose the greatest challenge. This requires the model to overcome challenges associated with low-resolution images and dense predictions. Additionally, factors such as different sizes, shapes, postures, occlusions, lighting conditions, and water reflections severely impact detection results. It is precisely because of these challenges that various methods were employed in the model design, carefully addressing these limitations to ensure the model's robustness and good overcome generalization performance. In this study, these challenges were successfully, leading to satisfactory experimental results, as demonstrated in the following sections.

TABLE I. DATASET DETAILS

GSD	Images	Labels	Panicles per sub-image
GSD_{7m}	3,810	106,878	27-30
GSD_{12m}	1,004	71,404	65-70
GSD_{20m}	558	81,216	140-150

B. Model Architecture

Taking into account the deployment requirements of edge devices in the field of plant science, the model architecture was designed with a focus on lightweight design. Effective design modifications were applied to the detection network structure, making it more comprehensive and detailed, particularly suitable for detecting rice spikes of varying sizes in the images. As shown in Fig. 2, the global architecture of the model consists of three main components: the Encoder for generating feature maps, the Decoder for feature parsing, and the Detector for visual output. The following sections will provide a detailed explanation of the design details and the rationale behind them.

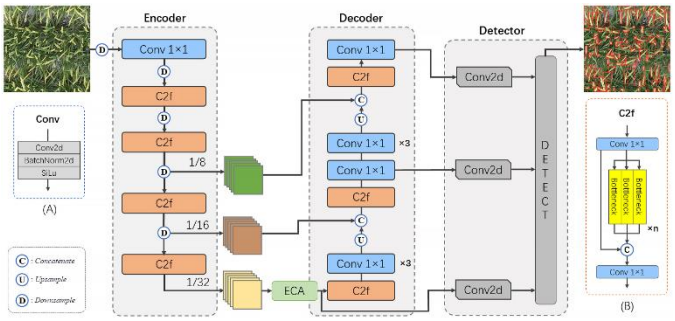

Fig. 2. The global architecture of the RLFDNet model.

1) Encoder: The role of the encoder is to map the input RGB image to feature maps. Given an input image $I \in R^{H\times W\times 3}$, RLFDNet employs the lightweight network CSPDarkNet [17] as the backbone for feature extraction. Down-sampling operations are performed using 2D convolution layers with a 3×3 kernel size and a stride of 2. CSPLayer [20] is inserted at different stages for feature extraction, combined with the C2f module [21] to generate feature maps at different stages. Through these operations, the input image undergoes 32 times downsampling, resulting in feature maps with channel numbers of 64, 128, and 256, representing 1/8, 1/16, and 1/32 of the original image, respectively. These feature maps carry richer gradient information and are utilized in the decoder.

In the final stage of the encoder, the Efficient Channel Attention (ECA) mechanism is applied. It compresses spatial information through global average pooling, learns channel attention information through a 1×1 convolution layer, and combines the channel attention information with the original input feature map. This approach avoids dimension reduction, effectively captures cross-channel interactions, and requires only a small number of parameters for excellent results. In summary, this encoder contributes to improving object detection performance, particularly in the extraction of features when dealing with targets of different scales.

2) Decoder: The role of the decoder is to combine and decode the features from the encoder, mapping them to the final output of object detection. In RLFDNet, after obtaining the feature layer output from the ECA attention mechanism, the C2f module is introduced to reduce redundant representations of convolutional kernels, significantly reducing the number of convolutions and parameters. At this point, this layer's features are passed as a branch to the Detector because it can maintain the detection of smaller objects. To better obtain high-level features and increase semantic information while considering model lightweighting, nearest neighbor upsampling is applied to the upper-level features, doubling the size of the feature map. The Conv layer receives features from the previous layer and concatenates three 1×1 convolutional layers, increasing the model's receptive field to cover a larger area of the image. Then, the corresponding scale-sized feature map extracted from the Encoder is concatenated, followed by another C2f and a 1×1 standard convolutional layer to reduce the number of parameters and computations. This portion of features is then split into two branches: one continues to concatenate the decoder for the same operations, and the other is output to the Detector for detection preparation. This design ultimately accumulates three sets of features at different scales, utilizing feature mappings of different scales for predictions, enhancing RLFDNet's perception of objects of different sizes.

3) Detector: The different-stage feature maps output from the Decoder are passed to the Detector. The main task of the Detector is to merge these feature maps and fuse the encoded information into the original feature map. It predicts the distances between each anchor point and the four edges of the target bounding box through the regression branch, determining the target's position. The Non-Maximum Suppression (NMS) is then applied to filter the generated prediction boxes. The Intersection over Union (IoU) evaluation metric is used to measure the overlap between two prediction boxes. By comparing the IoU values between prediction boxes, the model determines whether they belong to the same object, ultimately eliminating redundant detection results.

Overall, the RLFDNet model has a concise overall architecture design. Through the implementation of multi-scale feature fusion, context information aggregation, and the introduction of channel attention mechanisms, the model's perception and expressive capabilities are enhanced. This enables the model to better adapt to the detection of objects of different sizes and complexities. With minimal parameter settings, the model maintains its lightweight nature, reducing memory requirements, making it easy to deploy on low-end edge devices, and ensuring good real-time performance.

C. Loss Function

The loss function, serving as a guide for adjusting weights during backpropagation, measures the error between the forward propagation results of a neural network and the ground truth values in each iteration. In the implementation of

RLFDNet, various commonly used loss functions were explored. For the Complete Intersection over Union (CIoU) loss function [22], which functions as the bounding box loss, the calculation method is as described in Eq. (1) and Eq. (2):

$$CIoU = IoU - \frac{\rho^2\left(b,b^{g^t}\right)}{c^2} - \alpha v \quad (1)$$

$$L_{CIoU} = 1 - CIoU \quad (2)$$

IoU represents the intersection ratio of the real bounding box and the bounding box. c denotes the minimum diagonal length of the bounding box enclosing the predicted box and the ground truth box, and $\rho^2\left(b,b^{g^t}\right)$ represents the Euclidean distance between the center points of the ground truth box and the predicted box. The calculation method of α and v is shown in Eq. (3) and Eq. (4):

$$v = \frac{4}{\pi^2}(arctan\frac{w^{g^t}}{h^{g^t}} - arctan\frac{w}{h})^2 \quad (3)$$

$$\alpha = \frac{v}{(1-IoU)+v} \quad (4)$$

In Eq. (3), h^{g^t} and w^{g^t} represent the height and width of the ground truth box; h and w represent the height and width of the prediction box. CIoU Loss function considers the coverage area, aspect ratio, and center distance, comprehensively, which can measure its relative position well, and solve the problem of optimizing the horizontal and vertical directions of the prediction box, but this method does not consider the direction matching between the target box and the prediction box, which leads to a slow convergence speed. Thus, this paper used the Smooth Intersection over Union (SIoU) loss function [23]. SIoU introduces the optimization of the vector angle between the target box and the predicted box and plays a significant role in the strawberry detection network through a linear combination of four components: angle cost, distance cost, shape cost, and IoU cost. Its calculation method is as described in Eq. (5) and Eq. (6):

$$L_{SIoU} = 1 - IoU + \frac{\Delta + \Omega}{2} \quad (5)$$

$$IoU = \frac{|B \cap B^{GT}|}{|B \cup B^{GT}|} \quad (6)$$

where B and B^{GT} represent a prediction box and a ground truth box, Ω represents the shape cost, Δ represents the angle cost, and the distance cost is redefined. Ω and Δ are defined in Eq. (7) and Eq. (8):

$$\Omega = \sum_{t=w,h}(1 - e^{-w_t})^{\theta} \quad (7)$$

$$\Delta = \sum_{t=x,y}(1 - e^{-\gamma\rho_t}) \quad (8)$$

In Eq. (7), $w_w = \frac{|w-w^{g^t}|}{max\left(w,w^{g^t}\right)}$, $w_h = \frac{|h-h^{g^t}|}{max\left(h,h^{g^t}\right)}$, θ indicates the degree of concern for Ω.

In Eq. (8), $\rho_x = \left(\frac{|b_{c_x}^{g^t} - b_{c_x}|}{c_w}\right)^2$, $\rho_y = \left(\frac{|b_{c_y}^{g^t} - b_{c_y}|}{c_h}\right)^2$, γ is defined in Eq. (9):

$$\gamma = 1 + 2\sin^2(arcsin\frac{max(b_{c_y}^{g^t}, b_{c_y}) - min(b_{c_y}^{g^t}, b_{c_y})}{\sqrt{(b_{c_x}^{g^t} - b_{c_x})^2 + (b_{c_y}^{g^t} - b_{c_y})^2}} - \frac{\pi}{4}) \quad (9)$$

In Eq. (9), $b_{c_x}^{g^t}$ and $b_{c_y}^{g^t}$ represent the coordinates of the ground truth bounding box's center. b_{c_x} and b_{c_y} represent the coordinates of the predicted bounding box's center.

The SIoU loss function redefines distance loss by considering vector angles between required regressions, reducing regression freedom, accelerating network convergence, and enhancing accuracy. For instance, in densely packed rice panicles, SIoU effectively distinguishes boundaries, improving detection accuracy and stability. Therefore, SIoU is advantageous for detecting dense or small targets.

III. Experiments

A. Experimental Details

In this study, rice spike images captured at different altitudes, including 7m, 12m, and 20m, were used to evaluate the RLFDNet model. The detailed dataset information is shown in Table I, and images at each altitude were randomly divided into training, validation, and test sets in a ratio of 6:1:3. The experiments were implemented using the PyTorch deep learning framework [24] and accelerated with CUDA. Since each image's size was 512×512 pixels, inputting the model with the original image size maintained a low resolution, aligning more with the requirements of edge devices. To ensure the objectivity of results, all methods were trained and tested under the same configuration. During training, the batch size was set to 16, the learning rate was initialized to 0.01, Stochastic Gradient Descent (SGD) optimizer was used with a momentum factor of 0.937, and weight decay was set to 5×10^{-4}. To prevent overfitting and enhance model robustness, data augmentation techniques were applied, including color distortion, random translation, random flipping, random scaling, and random cropping. After configuring the relevant parameters, the RLFDNet model was optimized for 300 epochs based on convergence speed considerations.

B. Evaluation Metrics

When establishing a detection model, both precision and recall need to be considered. Therefore, this study used metrics such as Precision, Recall, F1-score, mAP@0.5, and mAP@0.5:0.95 to assess the model's performance and evaluate the detection results. The calculation methods for Precision, Recall, and F1-score are given by Eq. (10) to Eq. (12):

$$P = \frac{TP}{TP+FP} \quad (10)$$

$$R = \frac{TP}{TP+FN} \quad (11)$$

$$F1 = 2 \times \frac{P \times R}{P + R} \quad (12)$$

where, P represents precision, R represents recall, and F1 represents F1-score. TP (True Positive) denotes the number of positive samples correctly classified, TN (True Negative) represents the number of negative samples correctly classified. FP (False Positive) indicates the number of negative samples

incorrectly classified as positive, while FN (False Negative) represents the number of positive samples incorrectly classified as negative.

The mean Average Precision (mAP) represents the overall performance at different IoU thresholds, including mAP@0.5 and mAP@0.5:0.95. Here, mAP@0.5 denotes the average mAP at an IoU threshold of 0.5, with a higher value indicating higher detection accuracy for that category. mAP@0.5:0.95 represents the average mAP across different IoU thresholds (ranging from 0.5 to 0.95 with a step size of 0.05), providing a more stringent evaluation of the model's performance. The calculation method for mAP is given by Eq. (13):

$$mAP = \frac{1}{n}\sum_{1}^{n} P(R)d(R) \tag{13}$$

where, n is the number of classes, in this experiment, there is only one class, which is rice spikes, so n=1. In addition, since the distribution of rice spikes is dense, evaluating counting performance is also meaningful. Here, three metrics are used to assess the consistency between predicted and true values, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2). Specifically, they are defined by the following Eq. (14) to Eq. (16):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{14}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{15}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y}_i - y_i)^2} \tag{16}$$

where, n is the number of samples, y_i is the true count, $\hat{y}_i$ is the predicted count, and $\bar{y}_i$ is the mean of the true counts. These metrics provide a quantitative assessment of the model's ability to accurately predict the count of rice spikes. The result of R^2 falls within the range [0, 1], indicating the proportion of the variance in the predicted values to the variance in the actual values near the mean. This metric can be interpreted as the goodness of fit of the model, where 1 represents a perfect fit, and 0 indicates no linear relationship between the actual counts and the predicted values.

C. Analysis of Counting Performance of RLFDNet

To comprehensively evaluate the performance of the RLFDNet model across different altitudes (7m, 12m, and 20m), the model was trained and tested on each dataset, and the experimental results are presented in Table II.

Furthermore, Fig. 3 illustrates a linear regression plot, which is an indispensable tool in the analysis of counting task experiments. It visually presents the counting performance differences between RLFDNet model inference and manual counting. A closer alignment of points to the perfect prediction line indicates better model fitting. The results show that the model performs exceptionally well at 7m altitude, demonstrating more accurate predictions and higher model fitting (R^2=0.9461). Conversely, at altitudes of 12m and 20m, the model's performance slightly decreases, showing larger MAE, RMSE, and slightly lower R^2, indicating potential challenges in predicting at these two altitudes.

A deeper investigation into the performance differences at different shooting heights and discussion of the possible reasons for these differences were conducted. The variations in this regard are mainly influenced by two key factors: shooting height and environmental conditions. Firstly, changes in shooting height directly affect the size and resolution of panicles in the images. At lower altitudes, panicles are relatively larger and easier for the model to capture details. At higher altitudes, smaller panicles increase the difficulty of detection. Secondly, lighting conditions also vary at different altitudes, leading to varying degrees of light and shadow in the images. Uniform lighting at lower altitudes facilitates the model in capturing the edges and details of panicles. Conversely, lighting conditions at higher altitudes may be more complex, adding to the difficulty of model inference.

At the same time, Fig. 4 illustrates images with the highest prediction errors in each dataset. Ground Truth (GT) is represented by red points in the images, indicating manual counting results, while Predicted (PD) is indicated by red boxes in the inference images, representing the model's inference results. This aids in understanding the potential reasons behind these inaccuracies. Clearly, these images confirm a common notion that they mostly contain significant environmental noise. Factors such as differences in lighting, small target sizes, and high density significantly increase the difficulty of detection. In some cases, even experienced human experts may find identifying spikes challenging. A comprehensive evaluation of counting performance across the entire dataset will be further discussed in the next section.

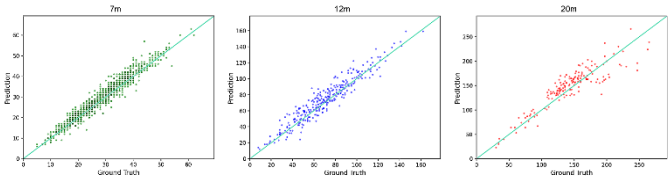

Fig. 3. The linear regression graph, illustrating the variance between counting results of RLFDNet model and human counting.

TABLE II. RLFDNET'S COMPREHENSIVE PERFORMANCE ACROSS ALTITUDES

GSD	P	R	mAP@0.5	mAP@0.5:0.95	MAE	RMSE	R^2
GSD_{7m}	0.915	0.923	0.961	0.691	1.86	2.49	0.946
GSD_{12m}	0.830	0.813	0.861	0.423	6.97	9.07	0.906
GSD_{20m}	0.871	0.839	0.907	0.614	15.26	19.29	0.816

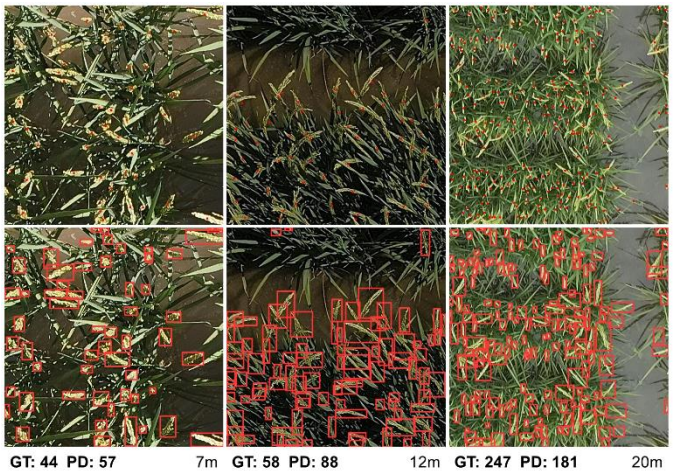

Fig. 4. Images with the maximum errors at each altitude. (GT: Red points - manual counting results; PD: Red boxes - RLFDNet's inference results).

D. Comparison with different Object Detection Method

To compare the superiority of the RLFDNet model, six advanced and commonly used object detection models, including YOLOv5 [25], YOLOv7-tiny [26], YOLOv8 [21], CenterNet [27], Faster R-CNN [28], and SSD [29], were selected for a comprehensive analysis of evaluation performance, counting performance, and model lightweighting. To ensure fair and objective results, they were trained and tested on identical training, validation, and test sets for each altitude dataset. Although this effort was substantial, it was meaningful, providing insights into the differences between different models and presenting results more objectively.

1) Performance comparison: The results of the evaluation performance for different models are presented in Table III. Upon examination of the test results, RLFDNet demonstrated satisfactory counting performance. While YOLOv8's results were very close, even surpassing RLFDNet in one metric at altitudes 12m and 20m, the difference was marginal. Overall, RLFDNet outperformed its counterparts. Conversely, other models exhibited slightly inferior performance in detecting rice panicles at each altitude, particularly Faster R-CNN and SSD. This suggests that these models may lack robustness when dealing with smaller targets or more complex conditions, hindering their ability to achieve highly accurate object detection.

2) Counting performance comparison: Furthermore, a detailed analysis of counting performance was conducted for each altitude dataset, and the experimental results are presented in Table IV. Observations revealed that YOLOv8, CenterNet, and RLFDNet consistently demonstrated stable prediction performance at each altitude, with RLFDNet maintaining the optimal performance. On the other hand, Faster R-CNN and SSD exhibited higher errors and lower fitting accuracy at higher altitudes, corroborating the results of the performance evaluation. These models faced challenges in object detection when dealing with smaller targets or more complex environments. In contrast, RLFDNet maintained relatively good and stable performance even at higher altitudes with smaller targets and higher density. The experiments indicate that RLFDNet exhibits strong generalization and robustness in counting performance.

TABLE III. COMPARISON OF EVALUATION PERFORMANCE ACROSS DIFFERENT MODELS

GSD	Model	F1	mAP@0.5	mAP@0.5:0.95
GSD_{7m}	YOLOv5	0.835	0.884	0.528
	YOLOv7-tiny	0.870	0.923	0.609
	YOLOv8	0.919	0.959	0.675
	CenterNet	0.600	0.958	0.756
	Faster R-CNN	0.647	0.654	0.635
	SSD	0.478	0.513	0.463
	RLFDNet	0.942	0.961	0.691
GSD_{12m}	YOLOv5	0.818	0.956	0.409
	YOLOv7-tiny	0.554	0.674	0.397
	YOLOv8	0.819	0.860	0.428
	CenterNet	0.638	0.793	0.432
	Faster R-CNN	0.493	0.392	0.364
	SSD	0.317	0.352	0.269
	RLFDNet	0.821	0.861	0.423
GSD_{20m}	YOLOv5	0.835	0.883	0.525
	YOLOv7-tiny	0.554	0.678	0.466
	YOLOv8	0.862	0.909	0.608
	CenterNet	0.656	0.877	0.566
	Faster R-CNN	0.369	0.237	0.192
	SSD	0.340	0.361	0.267
	RLFDNet	0.872	0.907	0.614

TABLE IV. COMPARISON OF COUNTING PERFORMANCE ACROSS DIFFERENT MODELS AT VARIOUS ALTITUDES

GSD	Model	MAE	RMSE	R^2
GSD_{7m}	YOLOv5	12.38	18.88	0.829
	YOLOv7-tiny	4.04	5.24	0.868
	YOLOv8	1.70	2.98	0.943
	CenterNet	1.70	2.93	0.942
	Faster R-CNN	10.74	11.67	0.843
	SSD	5.84	7.65	0.677
	RLFDNet	1.86	2.49	0.946
GSD_{12m}	YOLOv5	6.59	9.65	0.900
	YOLOv7-tiny	8.18	10.59	0.846
	YOLOv8	10.00	12.80	0.873
	CenterNet	9.61	12.12	0.883
	Faster R-CNN	18.51	21.55	0.653
	SSD	28.66	32.87	0.660
	RLFDNet	6.97	9.07	0.906
GSD_{20m}	YOLOv5	15.76	23.60	0.816
	YOLOv7-tiny	23.31	29.65	0.549
	YOLOv8	15.34	21.85	0.812
	CenterNet	19.72	24.91	0.696
	Faster R-CNN	27.34	35.57	0.302
	SSD	33.81	42.16	0.195
	RLFDNet	15.26	19.29	0.816

3) Model lightweight comparison: After evaluating model performance metrics and counting performance, it is crucial to consider another key factor in the design of the RLFDNet model – achieving lightweightness. To assess different models, the number of parameters (Params) is used to reflect the total trainable parameters in the network, indicating model complexity and its capacity to learn and represent features. The calculation is defined by Eq. (17):

$$Params = [i \cdot (k \cdot k) \cdot o] + o \quad (17)$$

where, i is the input size, k is the convolution kernel size, and o is the output size. Regarding inference efficiency, the evaluation is conducted using the Frames per Second (FPS) metric to reflect the model's inference speed. A higher FPS indicates a faster generation of inference results. The calculation is defined by Eq. (18):

$$FPS = \frac{1000}{pre\text{-}process + inference + NMS} \quad (18)$$

Here, pre-process, inference, NMS is pre-processing, inference, and Non-Maximum Suppression time, respectively for each image.

In this experiment, RLFDNet's efficiency is compared with different models. FPS measures the number of image frames the model can process per unit time, while Params is a direct measure of model complexity and an important constraint for deployment. The tests were conducted on an NVIDIA GTX1080Ti GPU (8G) device, a lower-end GPU with slower computational speed. The results are shown in Fig. 1. It is evident that RLFDNet achieves an excellent overall performance. Compared to YOLOv5, which is relatively close in performance, RLFDNet has only 4% more Params, while the FPS has increased by 70%, reaching 80.43 frames per second. This improvement is significant, as it maintains a relatively small total parameter count while substantially enhancing inference efficiency. It contributes greatly to deploying the model on low-end edge devices. The size of the model's parameter count directly impacts whether individuals with budget-friendly devices can enjoy the benefits of advanced technology, especially in resource-constrained fields such as agriculture, where edge devices, embedded systems, and mobile robots provide practitioners with more decision support and production management tools.

IV. Discussion

This study introduces the RLFDNet model, offering a lightweight and real-time method for rice panicle localization and counting. The model leverages the lightweight backbone CSPDarknet and introduces an innovative strategy in the design, maintaining a relatively low image resolution in experiments to meet the requirements of edge devices, achieving high accuracy and lightweight characteristics. As shown in Fig. 5, the model accurately localizes and counts rice panicles in four crucial growth stages at different shooting heights of 7m, 12m, and 20m. Moreover, RLFDNet demonstrates good robustness and adaptability when facing common natural factors in the field, such as strong sunlight, overcast conditions, and interferences like mutual occlusion, varied panicle poses, changes in lighting conditions, and water reflections, as depicted in Fig. 6.

Fig. 5. Rice panicle detection results at four different growth stages (Example at 7m altitude).

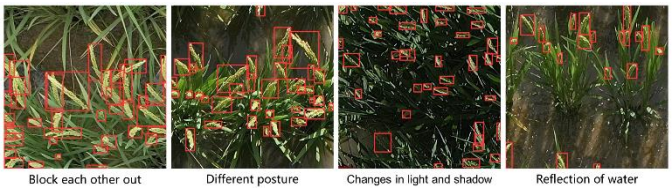

Fig. 6. Detection results of rice panicles in the face of various influencing factors (Example at 7m altitude).

In summary, RLFDNet's design incorporates several key innovations:

1) Multi-Scale receptive fields and feature fusion: RLFDNet employs a pyramid network architecture in the encoder to capture multi-scale information effectively. Different convolutional layers' receptive fields help in understanding spatial object relationships, while a feature fusion mechanism integrates features from various scales to enhance dense target detection.

2) Reinforcement learning-guided object detection: RLFDNet dynamically adjusts its object detection strategy during training using reinforcement learning mechanisms. This adaptive approach helps the model better adapt to changes in altitudes and environmental conditions, enhancing performance in complex scenes.

3) Attention mechanism integration: RLFDNet incorporates attention mechanisms at the connections between the encoder and decoder, allowing the model to adaptively focus on important regions in the image. By introducing attention mechanisms, the model learns the importance of target regions during training, improving the precision of target localization and counting accuracy.

4) Lightweight design: Prioritizing lightweight design for feasible deployment on edge devices, RLFDNet reduces the total number of network parameters, enhancing the model's inference efficiency. This design decision maintains counting performance while strengthening the model's adaptability in resource-constrained environments.

5) Environmental adaptability: Experimental details consider different shooting heights and environmental conditions, with the model employing a self-adaptive adjustment strategy. Learning richer features under varying conditions enhances adaptability to complex scenarios, crucial for practical rice panicle counting applications.

However, while RLFDNet significantly outperforms other advanced object detection methods in both accuracy and inference efficiency, it acknowledges certain limitations. Firstly, as the experiments were conducted at three specific altitudes (7m, 12m, and 20m), real-world applications may involve different shooting heights not covered in this study, potentially affecting inference results due to variations in panicle size and density. Secondly, variations in rice panicle phenotypes due to different rice varieties in different regions could result in suboptimal inference performance, highlighting the need for further research in addressing these limitations.

V. CONCLUSION

In this research, rice panicle localization and counting method named RLFDNet was designed and proposed. A concise and efficient encoder-decoder module was further developed within the model. A series of experiments demonstrated that RLFDNet achieved excellent results in rice panicle detection at different shooting heights, providing real-time and accurate localization and counting of rice panicles. With an MAE of 1.86 and an R^2 of 0.9461, the model showed robust performance. Considering various altitudes, the model achieved an average accuracy of mAP@0.5 at 0.91, with a total parameter count of only 4.40M. The inference efficiency reached 80.43 FPS, meeting the requirements for deployment on low-end edge devices. This provides a valuable tool for farmers and governments in assessing rice yields. In the future, exploration will be conducted to test the model with more shooting heights and different rice varieties to expand its capability to adapt to diverse environmental conditions, such as varying lighting and weather patterns, thereby enhancing its adaptability and reliability in real agricultural settings. The aim is to broaden its applicability across different countries and regions while addressing emerging challenges in agricultural technology.

REFERENCES

[1] Liu S, Baret F, Andrieu B, Burger P, Hemmerlé M. "Estimation of Wheat Plant Density at Early Stages Using High Resolution Imagery." Frontiers in Plant Science. 2017, 8:739.

[2] Slafer GA, Savin R, Sadras VO. "Coarse and fine regulation of wheat yield components in response to genotype and environment." Field Crops Research. 2014 Feb, 157:71-83.

[3] Yang J, Sun J, Du L, et al. "Monitoring of Paddy Rice Varieties Based on the Combination of the Laser-Induced Fluorescence and Multivariate Analysis." Food Anal. Methods. 2017, 10:2398–2403.

[4] LeCun Y, Bengio Y, Hinton G. "Deep learning." Nature. 2015, 521:436–444.

[5] Stafford JV. "Implementing precision agriculture in the 21st century." Journal of Agricultural Engineering Research. 2000, 76:267-275.

[6] Hong Son N, Thai-Nghe N. "Deep Learning for Rice Quality Classification." 2019 International Conference on Advanced Computing and Applications (ACOMP). 2019, pp. 92-96.

[7] Singh N, Tewari VK, Biswas PK, Pareek CM, Dhruw LK. "Image processing algorithms for in-field cotton boll detection in natural lighting conditions." Artificial Intelligence in Agriculture. 2021, 5:142-156.

[8] Dorj UO, Lee M, Yun SS. "An yield estimation in citrus orchards via fruit detection and counting using image processing." Computers and electronics in agriculture. 2017, 140:103-112.

[9] Barreto A, Lottes P, Yamati FRI, et al. "Automatic UAV-based counting of seedlings in sugar-beet field and extension to maize and strawberry." Computers and Electronics in Agriculture. 2021, 191:106493.

[10] Xiong X, Duan L, Liu L, et al. "Panicle-SEG: a robust image segmentation method for rice panicles in the field based on deep learning and superpixel optimization." Plant Methods. 2017, 13:104.

[11] Misra T, Arora A, Marwaha S, et al. "SpikeSegNet-a deep learning approach utilizing encoder-decoder network with hourglass for spike segmentation and counting in wheat plant from visual imaging." Plant Methods. 2020, 16:40.

[12] Wang F, Mohan V, Thompson A, Dudley R. "Dimension fitting of wheat spikes in dense 3D point clouds based on the adaptive k-means algorithm with dynamic perspectives." 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor). 2020; pp. 144-148.

[13] Shu BY, Jiong M, Haoyang Y, Hongjie W, Jie Y. "Detection of Ears of Rice in field Based on SSD." Proceedings of the 2020 6th International Conference on Computing and Artificial Intelligence. 2020.

[14] N. C. Tri et al., "A novel approach based on deep learning techniques and UAVs to yield assessment of paddy fields," 2017 9th International Conference on Knowledge and Systems Engineering (KSE), Hue, Vietnam, 2017, pp. 257-262.

[15] Hayat MA, Wu J, Cao Y. "Unsupervised Bayesian learning for rice panicle segmentation with UAV images." Plant Methods. 2020, 16:18.

[16] Reza N, Na IS, Baek SW, Lee K. "Rice yield estimation based on k-means clustering with graph-cut segmentation using low-altitude UAV images." Biosystems Engineering. 2019, 177:109-121.

[17] Bochkovskiy A, Wang CY, Liao HYM. "Yolov4: Optimal speed and accuracy of object detection." [Online]. Available: https://arxiv.org/abs/2004.10934.

[18] Teng Z, Chen J, Wang J, Wu S, Chen R, Lin Y, Shen L, Jackson R, Zhou J, Yang C. "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. 2023, 5:0105.

[19] Teng Z, Chen J, Wang J, Wu S, Chen R, Lin Y, Shen L, Jackson R, Zhou J, Yang C. "Diverse Rice Panicle Detection." [Online]. Available: https://github.com/changcaiyang/Panicle-AI.

[20] Wang CY, Mark Liao HY, Wu YH, Chen PY, Hsieh JW, Yeh IH. "CSPNet: A New Backbone that can Enhance Learning Capability of CNN." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).

[21] Jocher, G., Chaurasia, A., & Qiu, J. (2023). YOLO by Ultralytics (Version 8.0.0) [Computer software]. URL https://github.com/ultralytics/ultralytics.

[22] Zheng Z, Wang P, Ren D, Liu W, Ye R, Hu Q, Zuo W. "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation." IEEE Transactions on Cybernetics. 2020, 52:8574-8586.

[23] Gevorgyan, Z. (2022). SIoU loss: More powerful learning for bounding box regression. arXiv preprint arXiv:2205.12740. doi: 10.48550/arXiv.2205.12740.

[24] Paszke, Adam, et al. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural in-formation processing systems, 32. arXiv preprint arXiv:1912.01703. doi: 10.48550/arXiv.1912.01703.

[25] Jocher G. "YOLOv5 by Ultralytics (Version 7.0) [Computer software]." [Online]. Available: https://doi.org/10.5281/zenodo.3908559.

[26] Wang CY, Bochkovskiy A, Liao HYM. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, pp. 7464-7475.

[27] Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q. "CenterNet: Keypoint Triplets for Object Detection." 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019, pp. 6568-6577.

[28] Ren S, He K, Girshick R, Sun J. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017, 39(6):1137-1149.

[29] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. "SSD: Single Shot MultiBox Detector." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I. 2016, 14:21-37.