High Accuracy Classification Algorithm of Wheat Grains Based on Improved YOLOv8

Guoxi Cheng[*1], Zhaohui Zhang[*1,*2,†] and Xiaoyan Zhao[*1,*2]

[*1] Shunde Innovation School, University of Science and Technology Beijing
2 Zhihui Road, Daliang, Shunde District, Fo Shan, Guangdong 528399, China
chengguoxi0712@163.com
[*2] School of Automation and Electrical Engineering, University of Science and Technology Beijing
30 Xueyuan Road, Haidian District, Beijing 100083, China
zhangzhaohui@ustb.edu.cn
†Corresponding author

Abstract. Wheat plays a vital role as one of China's major grain crops, and quality identification through wheat sampling tests is crucial in various aspects of wheat production, circulation, and consumption. Nevertheless, the accuracy of the current wheat detection algorithms are rather low, and the issues of missed detections and false positives are prone to occur. To address and improve these problems, this study proposed an enhanced wheat grain detection algorithm called YOLOv8n-wheat, based on the YOLOv8 target detection algorithm. Firstly, the performance of the model in learning complex features was enhanced by incorporating deformable convolution within the backbone network. Secondly, the attention mechanism CBAM was integrated, combining the channel attention module and the spatial attention module to enhance the network's representation capability. Lastly, the small target detection head was modified to fuse shallow feature information with deeper feature information, thereby enhancing the model's sensitivity to small targets. Experimental tests were conducted to evaluate the effectiveness of the proposed YOLOv8n-wheat grain detection algorithm. The results demonstrated an accuracy of 90.5%, a recall of 89.4%, mAP@0.5 of 0.708, and mAP@0.5:0.95 of 0.560. The enhanced accuracy of the proposed algorithm establishes its practicality and its potential application in wheat grain detection.

Keywords: YOLOv8, Wheat Kernel Detection, Deformable Convolutional Networks, Attention Mechanism, Feature Extraction.

1 Introduction

According to statistics, in 2023, China's wheat cultivation covered approximately 23,627.2 thousand hectares, with a total wheat output of 136,590,000 tonnes, cementing its position as the third largest grain crop in the country. Wheat finds extensive applications in food, feed, and various industries. It is a nutritious crop abundant in carbohydrates, vitamins, dietary fiber, and protein. Wheat is commonly used in the

production of flour, bread, pasta, and numerous other food products. As one of China's major food crops, the sampling and testing of wheat for quality identification play a crucial role in wheat production, distribution, and consumption. The national standard for wheat, GB1351-2023, released on May 23, 2023, classifies wheat grains into categories such as sound kernel, broken kernel, sprouted kernel, diseased kernel, insect-bored kernel and useless material. Ensuring comprehensive quality testing for unsound wheat kernel holds immense practical and strategic significance for the smooth development of China's economy, while also safeguarding wheat quality and food safety in the country.

The rapid advancement of deep learning in recent years has propelled progress across various fields, leading to the emergence of machine vision combined with machine learning and deep learning as a promising direction for wheat grain detection. Several scholars have utilized deep learning techniques to detect wheat grains. For instance, C. Hao et al. employed a convolutional neural network to construct a feature pyramid, enabling fusion of imperfect grain features and enhancing recognition accuracy [1]. A. Singh et al. utilized models like VGG16, AlexNet, and ResNet to extract features and classify wheat grains for recognition [2]. K. Laabassi et al. identified wheat grains using convolutional neural networks in conjunction with transfer learning [3]. Y. Dai et al. applied the MobileNet V2 network model to detect wheat ruderalis grains [4]. Q. Hong et al. combined YOLOv4 and MobileNet to develop a model for detecting wheat ruderal grains, achieving promising results [5]. Z. Zhang et al. employed the YOLOv5 algorithm as a base model and integrated three attention mechanisms to enhance network representation and improve wheat detection accuracy [6]. K. Han et al. combined the YOLOv5 algorithm with their own EfficientNet-b2-W classification network cascade to create a multi-seed classification model for wheat, accompanied by system software for convenient user implementation [7]. A. Yasar et al. proposed a combination of CNN and transfer learning to enhance wheat detection accuracy [8].

While more scholars are adopting deep learning methodologies for wheat grain detection, existing algorithms for wheat grain recognition still suffer from issues such as low accuracy and susceptibility to missed detections and false positives. To address and improve these challenges, this study focused on optimizing and enhancing the novel YOLOv8n target detection algorithm. The network model was improved by incorporating deformable convolution, embedding the attention mechanism CBAM, and modifying the target detection head. The objective was to enhance the accuracy of wheat detection and recognition.

2 Related Work

2.1 YOLOv8n Target Detection Algorithm

The YOLO family represents a prominent class of one-stage target detection algorithms. J. Redmon et al. initially proposed the YOLOv1 algorithm in 2016 [9]. Through continuous development and iteration, subsequent models such as

YOLO9000, YOLOv3, and YOLOv4 were introduced, incorporating the strengths of their predecessors. In 2023, Ultralytics presented the YOLOv8 algorithm, which garnered significant attention due to its exceptional speed and accuracy [10-12]. The YOLOv8 algorithm demonstrates remarkable performance in target detection tasks and has achieved impressive results on various benchmark datasets. Furthermore, the YOLOv8 algorithm introduces novel ideas and methodologies that provide valuable insights for future research in target detection algorithms. The YOLOv8 framework encompasses five network model specifications: n, s, m, l, and x, each offering a different trade-off between complexity, detection accuracy, training difficulty, and detection speed. The YOLOv8n model, being the smallest and fastest, imposes lower hardware arithmetic requirements. The YOLOv8n network comprises three key components: the Backbone network, the Neck network, and the detection Head network, as depicted in **Fig. 1** below.

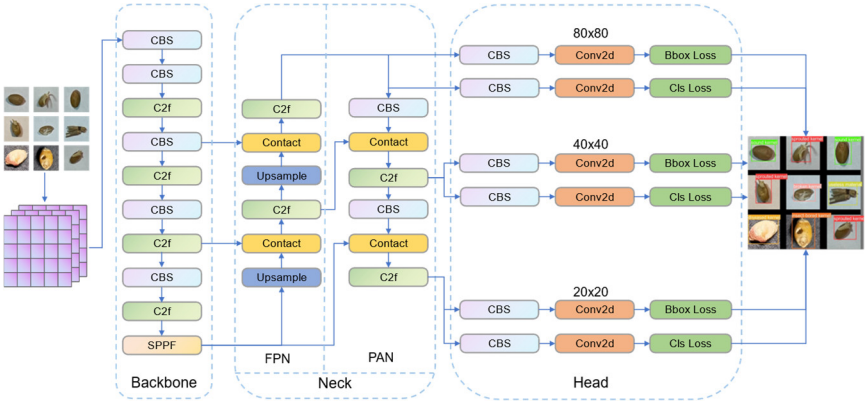

Fig. 1. YOLOv8n model network architecture diagram.

In this study, we proposed improvements and optimizations to YOLOv8n to enhance the model detection accuracy. We employed three strategies: backbone network using deformable convolutional DCN, inclusion of the attention mechanism CBAM, and modifying the target detection head.

2.2 Backbone Network using Deformable Convolution

DCN (Deformable convolutional networks) is an improved convolution operation proposed by J. Dai et al. to enhance the perception of convolutional neural networks when dealing with images containing deformable targets or complex scenes [13]. Unlike traditional convolutional operations that are applied to fixed sample points, deformable convolution enables the network to dynamically adapt and perceive deformation information in images by learning deformable sample point locations. The main concept behind deformable convolution is the introduction of learnable offsets to adjust the sampling position of the convolution kernel on the input feature map. These offsets are learned during training and adaptively adjusted based on the content

and contextual information of the input image. Deformable convolution allows each convolution kernel's sampling position to be dynamically adjusted to better accommodate the deformation of the target and the complexity of the background. The key operations involved in deformable convolution are offset computation and feature interpolation. In offset computation, deformable convolution predicts the offset for each pixel point based on the content of the input feature map by utilizing an offset regression network. During feature interpolation, the sampled points on the input feature map are interpolated based on the learned offsets to obtain new sampled point locations, and then the convolution operation is performed. **Fig. 2** illustrates a comparison between conventional convolution and deformable convolution.

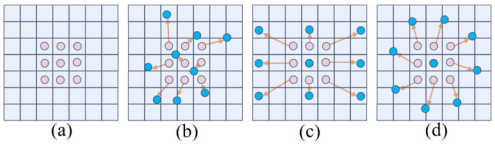

Fig. 2. Deformable convolutional networks. The pink dots represent conventional convolution, while the blue dots represent deformable convolution. (a) depicts a regular 3x3 convolution block, (b), (c), and (d) are deformable convolutions formed by adding offsets to each convolution kernel based on (a). (c) performs size scaling, and (d) performs a rotation operation.

In the following equations, p_0 denotes any point on the input feature map, p_n represents each offset within the range of the convolution kernel, in **Fig. 3**, R denotes the range of values for p_n, $w(p_n)$ denotes the weights, $x(p_0 + p_n)$ represents the value of the element at the corresponding position in the feature map, $y(p_0)$ denotes the result of the convolution operation, and Δp_n denotes the offset of deformable convolution. Equation **(1)** represents the formula for regular convolution, while equations **(2)** and **(3)** represent the formulas for deformable convolution. Deformable convolution introduces an offset for each point, and the operation process may result in fractional values that do not correspond to exact points on the input feature map. Typically, this problem is resolved through bilinear interpolation.

(-1,-1)	(-1,0)	(-1,1)
(0,-1)	(0,0)	(0,1)
(1,-1)	(1,0)	(1,1)

Fig. 3. Illustrates the range of values for R.

$$y(p_0) = \sum_{p_n \in R} w(p_n) * x(p_0 + p_n) \tag{1}$$

$$R = \{(-1,-1), (-1,0), \dots, (0,0), \dots, (1,0), (1,1)\} \tag{2}$$

$$y(p_0) = \sum_{p_n \in R} w(p_n) * x(p_0 + p_n + \Delta p_n) \quad (3)$$

Deformable convolution, by introducing learnable offsets to adjust the sampling positions of the convolution kernel, enables the network to better adapt to deformed targets in images, thereby enhancing target perception and discrimination. It has demonstrated significant improvements in various computer vision tasks, including target detection, semantic segmentation, and pose estimation. Particularly, deformable convolution provides more accurate prediction results when dealing with challenging scenes involving deformed targets, occlusion, and pose changes. In this study, we replaced the conventional convolution in the four C2f modules of the YOLOv8n backbone network with deformable convolution to improve the model's performance in learning complex features.

2.3 Attentional Mechanism CBAM

The attention mechanism is a technique employed to enhance image processing tasks by enabling models to dynamically select and focus on task-relevant regions or features during image processing. This approach allows models to prioritize task-relevant image regions and features, thereby improving the performance of image processing tasks such as target detection, image classification, image segmentation, and image generation. Attention mechanisms are widely used in computer vision, particularly for processing large-sized images or complex scenes, as they provide enhanced representational capabilities and accuracy.

CBAM (Convolutional Block Attention Module) is a lightweight attention mechanism proposed by S. Woo et al. [14]. It is primarily used to enhance the performance of convolutional neural networks in image processing tasks. The authors integrated CBAM into classical structures like ResNet and MobileNet and achieved promising results. CBAM focuses on extracting the most important features in an image and weighting these features across different spatial and channel dimensions.

CBAM consists of two main components: the Channel Attention Module and the Spatial Attention Module. The Channel Attention Module, depicted in **Fig. 4**, adaptively weights the feature maps of each channel. It captures the global correlation between channels by computing global average pooling and global maximum pooling for each channel. These pooled features are then passed through two fully connected layers, one generating the maximum response for the channel and the other generating the average response. Finally, these two responses are summed and normalized using a sigmoid function to obtain a channel attention map. This map assigns weights to features in the channel dimension, allowing the network to pay more attention to important channels. Similarly, the Spatial Attention Module, illustrated in **Fig. 5**, adaptively weights the feature map at each spatial location. It captures the global correlation in the spatial dimension by computing the mean and maximum of the feature map at each location. The output is then processed by a fully connected layer and sigmoid function normalization to obtain a spatial attention map, which directs the network's attention to important spatial locations.

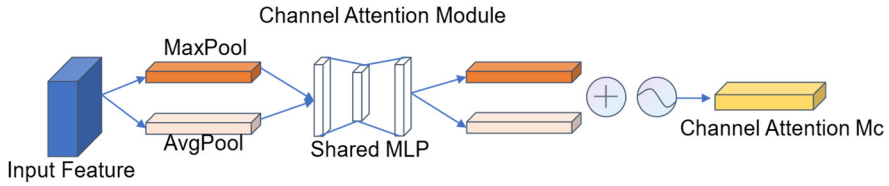

Fig. 4. CAM schematic diagram.

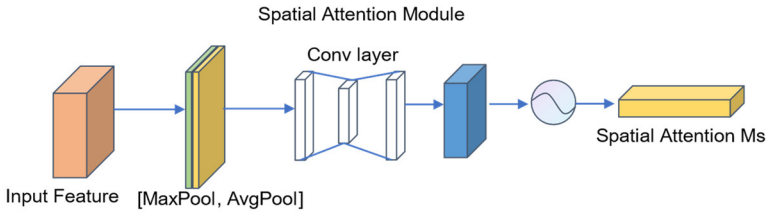

Fig. 5. SAM schematic diagram.

By combining channel attention and spatial attention, CBAM weights features in both the channel and spatial dimensions, extracting the most representative and important features. **Fig. 6** presents the schematic diagram of CBAM. This adaptive weighting mechanism helps the network better adapt to the characteristics of different images and improves the performance of image processing tasks, including image classification, target detection, and image segmentation. Additionally, the CBAM module is plug-and-play, easily integrating into existing base model networks for end-to-end training. In this study, the CBAM attention module was added to both the FPN and PAN in the YOLOv8n Neck Network. This integration combined the channel attention module and the spatial attention module to enhance the model network representation capability, enabling the model to pay more attention to important feature information.

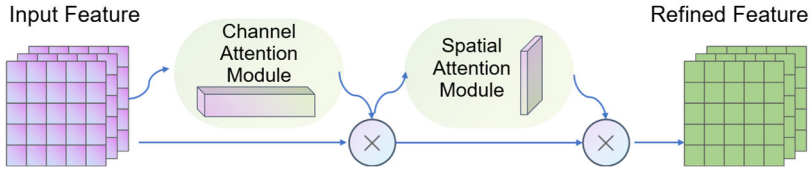

Fig. 6. CBAM schematic diagram.

2.4 Modifying the Target Detection Head

The Head network of YOLOv8n utilizes three feature maps for prediction, with an input image size of 3*640*640. The smallest feature map is obtained by down sampling the original image by 32 times, resulting in a size of 144*20*20. The middle feature map is downscaled by 16 times, resulting in a size of 144*40*40. The largest feature map is obtained by downsampling the original image by 8 times, resulting in a

size of 144*80*80. However, as the down sampled feature maps become deeper, it becomes challenging to effectively capture feature information for small targets. To address this limitation, a modification to the target detection head in the Head network of YOLOv8n was proposed.

In this study, the number of down sampling operations was reduced from the original 32 times to 4 times, resulting in a feature map size of 144*160*160. By reducing the down sampling factor, we retained more detailed information, enabling the model to focus better on small targets. This modification enhanced the model's capacity to pay attention to the fine-grained features of small targets, thereby increasing sensitivity and improving detection accuracy. Importantly, this change didn't significantly impact the model's complexity, allowing it to remain efficient and accurate even with limited computational resources.

The YOLOv8n-wheat model architecture is presented, as depicted in **Fig. 7** below. Compared to YOLOv8n, it can be seen that 3 modifications have been made. Sequence number 1 indicates that a deformable convolution is used, sequence number 2 indicates that an attentional mechanism CBAM is embedded, and sequence number 3 indicates that the target detection header has been modified.

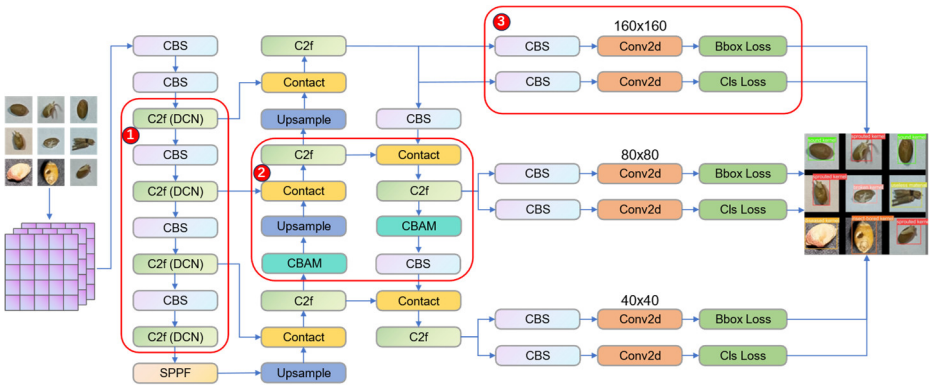

Fig. 7. YOLOv8n-wheat model network architecture diagram.

3 Experiment and Analysis

3.1 Data Sets and Experimental Environments

Using our self-constructed dataset, 2901 images of wheat grains were collected in total, as illustrated in **Fig. 8**. The dataset consisted of the following categories: 553 images of sound kernel, 498 images of broken kernel, 409 images of insect-bored kernel, 519 images of sprouted kernel, 477 images of diseased kernel, and 445 images of useless material. Subsequently, the wheat categories in the dataset were randomly divided into training, validation, and testing sets in an 8:1:1 ratio.

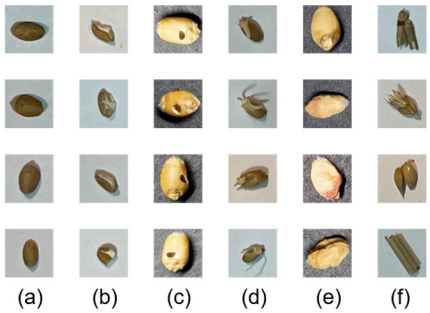

Fig. 8. Schematic diagram of wheat kernels by category. In this figure, (a) shows sound kernel, (b) shows broken kernel, (c) shows insect-bored kernel, (d) shows sprouted kernel, (e) shows diseased kernel, and (f) shows useless material.

The dataset underwent several preprocessing steps, including the Mosaic enhancement, random cropping, flipping, scaling, color perturbation, and hybrid enhancement techniques. During training, the hyperparameters was set as follows: the input image size was set to 640*640, the initial learning rate was 0.001, and the learning rate was updated using the cosine annealing algorithm. The batch size was set to 8, the training period consisted of 150 epochs, and the optimizer employed was SGD. The experiments were conducted with hardware and software parameter settings as outlined in **Table 1** below.

Table 1. Experimental environment settings.

Options	Parametric
Operating system	Ubuntu 18.04.6 LTS (Bionic Beaver)
CPU	Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz
GPU	NVIDIA GeForce GTX 1080 Ti
RAM	64GB
Memory(GPU)	12GB
Deep learning frameworks	PyTorch 1.10.0

3.2 Evaluation Indicators

In this study, the performance of the model was evaluated by using accuracy, recall, and mAP (mean Average Precision) as evaluation metrics. The specific formulas for these metrics are as follows:

Accuracy: Accuracy measures the overall correctness of the model's predictions. It is calculated by dividing the sum of true positives (TP) and true negatives (TN) by the total number of samples (N):

$$P = \frac{TP}{TP + FP} \tag{4}$$

Recall: Recall, also known as the true positive rate, measures the proportion of positive samples that are correctly identified by the model. It is calculated as the ratio of true positives (TP) to the sum of true positives (TP) and false negatives (FN):

$$R = \frac{TP}{TP + FN} \tag{5}$$

To assess precision and recall across different confidence thresholds, we established a right-angled coordinate system, where the x-axis represents the recall rate (R) and the y-axis represents the accuracy (P). The area enclosed by the Precision-Recall (PR) curve and the coordinate axis is known as the average precision (AP), which represents the average precision across all confidence thresholds. The mean of the average precision values for all samples in the dataset is referred to as mAP.

$$AP = \int_0^1 P(r)dr \tag{6}$$

$$mAP = \frac{\sum_{i=0}^{n} AP_{(i)}}{n} \tag{7}$$

We calculated mAP at an IoU (Intersection over Union) threshold of 0.5, denoted as mAP@0.5. This threshold determines the level of overlap required between predicted bounding boxes and ground truth boxes for a detection to be considered correct. Additionally, mAP@0.5:0.95 represents the average mAP across IoU thresholds ranging from 0.5 to 0.95 in steps of 0.05.

3.3 Results

The YOLOv8n target detection algorithm serves as the base model for the YOLOv8n-wheat model, incorporating three improvements such as the use of deformable convolutional DCN, the inclusion of the attentional mechanism CBAM, and the replacement of the target detection head. The dataset and experimental environment described earlier were utilized, and the experimental results are presented in **Table 2**.

Analyzing the table, it is evident that the YOLOv8n-wheat model demonstrates enhanced effectiveness in identifying sprouted kernels and useless materials. This improved performance may be attributed to the distinct shape and color characteristics exhibited by sprouted kernels and useless materials. On the other hand, the detection results for sound kernels and diseased kernels appear to be relatively lower. Further examination of the experimental results revealed instances of misclassification, particularly where sound kernels were misidentified as diseased kernels, and vice versa. This misclassification might be attributed to the similarities in shape, color, and texture characteristics between sound kernels and diseased kernels. **Fig. 9** provides a visual representation of the actual detection results for each category of wheat grains.

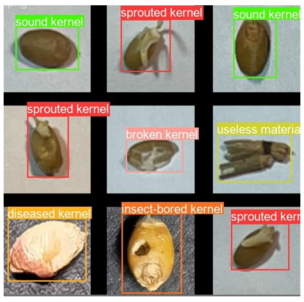

Fig. 9. Schematic diagram of the detection effect of wheat grains in each category.

Table 2. YOLOv8n-wheat experimental results.

Category	Accuracy/%	Recall/%	mAP@0.5	mAP@0.5:0.95
sound kernel	86.4	84.9	0.665	0.525
broken kernel	91.7	90.6	0.710	0.565
insect-bored kernel	92.3	89.2	0.700	0.555
sprouted kernel	91.5	89.5	0.715	0.570
diseased kernel	87.4	87.8	0.680	0.540
useless material	93.7	94.4	0.735	0.585
all	90.5	89.4	0.708	0.560

3.4 Ablation Experiment

The objective of this study is to introduce three improvements to the YOLOv8n algorithm. To validate the effectiveness of these enhancements, ablation experiments were designed and conducted. A total of five experiments were performed, maintaining the same dataset, hyperparameters, and experimental environment for each experiment. The only variation across the experiments was the algorithm model employed. The experimental results are presented in **Table 3**. "YOLOv8n(DCN)" denotes the case where only the regular convolution in the Backbone is replaced with deformable convolution. "YOLOv8n-wheat" signifies the utilization of all three proposed improvements, and so forth for the remaining cases.

Table 3. Ablation experiment results.

Method	DCN	CBAM	Modifying detector head	Accuracy/%	Recall/%	mAP @0.5	mAP @0.5:0.95
YOLOv8n	×	×	×	87.2	86.7	0.683	0.532
YOLOv8n (DCN)	√	×	×	88.5	86.8	0.692	0.541
YOLOv8n (CBAM)	×	√	×	87.9	87.3	0.689	0.552
YOLOV8n (Head)	×	×	√	89.7	88.6	0.691	0.557
YOLOv8n-wheat	√	√	√	90.5	89.4	0.708	0.560

3.5 Comparison Experiment

A comparison experiment was set up in this study, and YOLOv5n, YOLOX-Tiny, YOLOv7-Tiny, and YOLOv8n-wheat were used for performance comparison. The data set, hyperparameters and experimental environment of each experiment remained the same, only the algorithm model was different. The experimental results are shown in **Table 4**.

Table 4. Comparison experiment results.

Method	Accuracy/%	Recall/%	mAP @0.5	mAP @0.5:0.95	Speed (ms/img)	Params (M)	FLOPs (B)
YOLOv5n	81.8	77.9	0.459	0.407	142	1.9	4.5
YOLOX-Tiny	84.7	83.8	0.494	0.477	193	5.06	6.45
YOLOv7-Tiny	79.9	81.2	0.442	0.403	89	6.2	3.46
YOLOv8n	87.2	86.7	0.683	0.532	247	3.2	8.7
YOLOv8n-wheat	90.5	89.4	0.708	0.560	292	4.1	11.4

The YOLOv8n-wheat model demonstrates an accuracy of 90.5%, a recall of 89.4%, an mAP@0.5 of 0.708, and an mAP@0.5:0.95 of 0.560. Although the number of parameters and computation of YOLOv8n-wheat are increased, and the inference time is longer, these are still acceptable. These results indicate that combining these improvements significantly enchances the YOLOv8n model's performance.

4 Conclusions

In this study, an improved and optimized algorithm, YOLOv8n-wheat, was proposed for wheat grain detection based on the YOLOv8 target detection algorithm. The algorithm incorporates several enhancements to enhance its performance. Firstly, the utilization of deformable convolution in the backbone network improves the model's ability to learn complex features. Secondly, the attention mechanism CBAM is introduced, combining the channel attention module and spatial attention module to enhance the model's network representation capability. Lastly, modifications in the target detection head merge shallow feature information with deeper feature information, enhancing the model's sensitivity to small targets.

Experimental evaluations were conducted to assess the effectiveness of the proposed approach. The results indicate that the YOLOv8n-wheat grain detection algorithm achieves an accuracy of 90.5%, a recall of 89.4%, an mAP@0.5 of 0.708, and an mAP@0.5:0.95 of 0.560. These performance metrics represent improvements of 3.3, 2.7, 2.5, and 2.8 percentage points, respectively, compared to the original YOLOv8n algorithm. The proposed algorithm demonstrates high practicality and applicability in the field of wheat grain detection. Additionally, this study provides valuable insights for enhancing other target detection tasks and serves as a foundation for further research and development of detection models with superior performance.

References

1. C. Hao et al., “Wheat imperfect grain identification based on hyperspectral fusion images,” Modern Computer, Vol. 36, pp. 44–48, 2019. https://doi.org/10.3969/j.issn.1007-14 23.2 019.36.009
2. A. Singh and M. Arora, “CNN Based Detection of Healthy and Unhealthy Wheat Crop,” 2020 International Conference on Smart Electronics and Communication (ICOSEC), pp. 121-125, 2020. https://doi.org/10.1109/ICOSEC49089.2020.9215340.
3. K. Laabassi et al., “Wheat varieties identification based on a deep learning approach,” Journal of the Saudi Society of Agricultural Sciences, Vol. 20, No. 5, pp. 281-289, 2021. https://doi.org/10.1016/j.jssas.2021.02.008.
4. Y. Dai et al., “Recognition of wheat blast disease based on image processing and Deeplabv3+ model,” Chinese Journal of Agricultural Mechanical Chemistry, Vol. 42, No. 9, pp. 209, 2021. https://doi.org/10.13733/j.jcam.issn.2095-5553.2021.09.29
5. Q. Hong et al., “A Lightweight Model for Wheat Ear Fusarium Head Blight Detection Based on RGB Images,” Remote Sensing, Vol. 14, No. 14, pp. 3481, 2022. https://d oi.org/10.3390/rs14143481
6. Z. Zhang et al., “Real-Time Wheat Unsound Kernel Classification Detection Based on Improved YOLOv5,” J. Adv. Comput. Intell. Intell. Inform., Vol. 27, No. 3, pp. 474-480, 2023. https://doi.org/10.20965/jaciii.2023.p0474
7. K. Han et al., "An improved strategy of wheat kernel recognition based on deep learning," Revista DYNA, Vol. 98, No. 1, pp. 91-97, 2023. https://doi.org/10.6036/10686
8. A. Yasar, "Benchmarking analysis of CNN models for bread wheat varieties," European Food Research and Technology, Vol. 249, No. 3, pp. 749-758, 2023. https://doi.or g/1 0.1007/s00217-022-04172-y
9. J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016. https://doi.org/10.1109/ CVPR.2016.91
10. J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” 2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 7263-7271, 2017. https://doi.org/1 0.48550/arXiv.1612.08242
11. J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv:1804.02767, 2018. https://doi.org/10.48550/arXiv.1804.02767
12. A. Bochkovskiy, C.Y. Wang, and H.Y.M. Liao, “Yolov4: Optimal speed and accuracy of object detection,” arXiv:2004.10934, 2020. https://doi.org/10.48550/arXiv.2004.10934
13. J. Dai et al., “Deformable convolutional networks,” 2017 IEEE International Conference on Computer Vision (ICCV), pp. 764-773, 2017. https://doi.org/10.1109/ICCV.2017.89
14. S. Woo et al., “CBAM: Convolutional block attention module,” 2018 European conference on computer vision (ECCV), pp. 3-19, 2018. https://doi.org/10.1007/978-3-030-01234-2_1