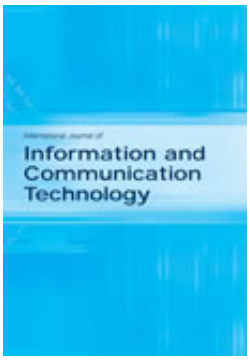
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research




**International Journal of Information and Communication Technology**

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

# Lightweight improvement algorithm for target detection of Pu'er tea harvesting robotic arm based on YOLOv8

Jing Xu, Wei Li

**Article History:**

| | |
|---|---|
| Received: | 10 September 2024 |
| Last revised: | 05 February 2025 |
| Accepted: | 05 February 2025 |
| Published online: | 16 April 2025 |

# Lightweight improvement algorithm for target detection of Pu'er tea harvesting robotic arm based on YOLOv8

Jing Xu and Wei Li*

School of Machinery and Transportation,
Southwest Forestry University,
Panlong District, Kunming City, Yunnan Province, China
Email: xuj@swfu.edu.cn
Email: liwei@swfu.edu.cn
*Corresponding author

**Abstract:** To tackle the challenges of recognition difficulties and constrained computational resources in Pu'er tea intelligent harvesting, this research develops an optimised, resource-efficient object detection algorithm built upon the YOLOv8n architecture for detecting tender shoots of Pu'er tea. The methodology incorporates three primary enhancements: first, replacing the standard Conv module with the Adown down-sampling component enhances detection precision, significantly boosts processing speed, and minimises model complexity; second, modifying the detection head to the LADH configuration cuts down parameter volume, further streamlining the model; third, integrating the AFGC attention mechanism refines detection accuracy. Experimental outcomes reveal that the optimised model achieves a 0.7% increase in mean average precision (mAP), accelerates detection speed by 482.9 FPS, and reduces model size by 1.9 MB compared to the baseline YOLOv8n. This work provides a technical foundation for advancing intelligent harvesting systems tailored for Pu'er tea cultivation.

**Keywords:** the intelligent harvesting of Pu'er tea; target detection; YOLOv8; lightweight.

**Reference** to this paper should be made as follows: Xu, J. and Li, W. (2025) 'Lightweight improvement algorithm for target detection of Pu'er tea harvesting robotic arm based on YOLOv8', *Int. J. Information and Communication Technology*, Vol. 26, No. 8, pp.1–18.

**Biographical notes:** Jing Xu is a Lecturer and PhD student. Her main research areas are intelligent manufacturing, digital twin, and industrial robotics. She has led two scientific research projects, two teaching research projects, published several academic papers and co-edited a textbook.

Wei Li is a Professor, Doctor and Doctoral Supervisor. At present, she mainly engaged in the research of mechatronic control, complex system modelling and simulation, computer control and application, electrical automation, industrial design and other related aspects, participated in two national natural funds, presided over and completed a number of horizontal research projects. She has published more than 70 papers in high-level international journals and conferences, of which 15 are retrieved by SCI/EI, and 20 are published in Chinese core journals.

## 1 Introduction

Pu'er tea is a traditional historic tea from Yunnan Province and a distinctive pillar industry of Yunnan (Yunnan Provincial Department of Agriculture and Rural Affairs, 2021; Xu, 2021). Because of the tall stature of Pu'er tea trees, the harvesting method is also quite unique, often requiring the use of ladders or scaffolding to climb up and pick the leaves. Currently, Pu'er tea is harvested manually (Xu, 2013), but manual harvesting is inefficient, costly, unsustainable, poses safety risks, and is detrimental to the protection of ancient tea trees. Therefore, researching how to utilise mechanised and intelligent technologies and equipment to achieve automatic harvesting of Pu'er tea, thereby protecting ancient tea trees, improving harvesting efficiency, and reducing costs, is an urgent matter that requires immediate attention.

**Figure 1** Pu'er tea harvesting scenario (see online version for colours)

(a) (b)

The leaves on tea trees are similar in colour, the picking targets are small, and the leaves frequently overlap with each other, making the identification of tender tea shoots challenging. Nonetheless, due to the swift evolution of computer vision and artificial intelligence technologies, considerable progress has been achieved in research within this field. Current mainstream research primarily focuses on image processing and machine learning methods for recognising tender tea shoots. Image-based detection technology is crucial for enabling automated tea picking, as it allows for the precise identification of buds on the tea tree. Traditional machine learning methods in image recognition often depend on manually designed features like edges and corners. However, these features may fall short in capturing the complex, high-dimensional, and unstructured information present in images. In contrast, deep learning (Mylonas et al., 2019), particularly through convolutional neural networks (CNNs), automatically learns and identifies key features in images, such as the objects' shapes, textures, and colour arrangements, significantly enhancing recognition accuracy and generalisation capabilities. Consequently, deep learning has become the preferred approach for modern image recognition tasks. Widely used deep learning methods include the series of R-CNN algorithms that utilise region proposals as a basis, regression-based algorithms like YOLO and SSD, and emerging algorithms that utilise transformer models.

You only look once (YOLO) (Zhang et al., 2024; Endo et al., 2024; Lv et al., 2024; Zhu et al., 2024) is a real-time system for detecting objects by reformulating the detection task as a regression problem, predicting the locations of bounding boxes and class probabilities directly in the output layer. This approach improves training speed and reduces model complexity, making it more advantageous for implementation on mobile devices. It finds extensive application in numerous agricultural and forestry target detection areas. Yang et al. (2019) from Qingdao University of Science and Technology proposed an improved YOLO-v3 algorithm to recognise tea shoot picking points. They used image pyramids to obtain multi-scale features, optimised down-sampling with a residual network, substituted the fully connected layer with 1×1 convolution, and optimised the target box using K-means clustering to achieve efficient and accurate tea shoot detection and posture recognition. He (2023) from Sichuan Agricultural University used an improved YOLOv5 model to build a complex background dataset, enhancing model localisation accuracy and generalisation ability through Adam optimisation, label smoothing, the CBAM attention module, and the Wise IOU loss function, achieving high-precision tender shoot detection. Xu et al. (2022) from Qingdao Agricultural University introduced a two-level fusion network combining fast detection with YOLOv3 and high-precision classification with DenseNet201, achieving precise tea shoot detection and constructing a dataset of famous tea shoots. Yin et al. (2023) from Hangzhou Dianzi University proposed a green tea quality detection algorithm based on YOLOv5s, introducing dilated convolution to enhance the extraction of small features, optimising feature fusion, incorporating the CBAM attention mechanism to reduce interference, and using Swin transformer for cross-dimensional fusion of small-scale features, combined with SimOTA for dynamic sample assignment to improve the ability to recognise various tea qualities. Fang et al. (2022) from Zhejiang Sci-Tech University enhanced attention to small tender shoots by adding a 52x52 shallow layer at the neck of YOLOv4-tiny, introducing CBAM to suppress noise and enhance feature saliency, and using BiFPN to fuse multi-scale features, constructing a high-performance lightweight YOLOv4-tiny-Tea tender shoot detection model. Xu (2023) from Zhejiang Sci-Tech University addressed the challenge of tea tender shoot detection with Longjing 108 as the target, proposing a YOLO-Tea algorithm based on deep learning. They built a natural tender shoot dataset, optimised YOLOv3 to YOLO-Tea, and further lightened it to YOLO-Ghost. The model replaced DarkNet53 with DarkNet19, introduced MCBAM and Ghost modules, reduced parameters, improved accuracy, and achieved efficient real-time detection. Huang et al. (2023) from Nanjing Institute of Technology proposed the Compact-YOLO v4 optimisation version for mobile tea tender shoot recognition, replacing the Backbone with GhostNet and convolution in the neck with ghost convolution, and combining transfer learning to improve accuracy, aiding the application of tea picking robots. Chen (2019) from Qingdao University of Science and Technology proposed an improved PSO-SVM algorithm to segment famous tea tender shoots, removing noise to preserve tender shoots, and using YOLO deep networks to determine picking points.

The aforementioned studies have made significant breakthroughs in recognition algorithms applied to intelligent tea picking. However, the aforementioned studies still face issues in field applications, such as slow detection speed, large model computation requirements, and challenges for deployment on mobile devices. This study aims to optimise and improve existing detection models to address these problems, focusing on ancient tree tea tender shoots. The study designs a lightweight tea tender shoot recognition algorithm based on an improved YOLOv8n. The improved YOLOv8n

replaces the traditional Conv module with Adown down-sampling, reducing model complexity and improving detection accuracy. It uses an LADH asymmetric multi-stage compression detection head to reduce parameter count and computational resource consumption, and adopts adaptive fine-grained channel attention (AFGC) to enhance detection accuracy. The objective of this study is to furnish technical assistance for the advancement of intelligent harvesting machinery for Pu'er tea tender shoots.

## 2 Materials and methods

### *2.1 Experimental data acquisition*

This study centres on the tender shoots of ancient Pu'er tea trees found in the historic tea forests located in Mengsong Town, Menghai County, within the Xishuangbanna Dai Autonomous Prefecture of Yunnan Province. In late March 2024, tea shoot images were captured using a camera positioned perpendicularly to the tree canopy. The camera was positioned at a distance ranging from 40 to 50 cm from the tea tree's harvesting surface. Photos were taken on both sunny and cloudy days, during the hours of 8:00–11:00 AM and 2:00–5:00 PM, resulting in a collection of 1,632 images, each having a resolution of 2,304×4,096 pixels. Considering YOLOv8's standard input image size of 640×640 pixels, which strikes a balance between precision and computational efficiency, the images were resized accordingly to ensure optimal practical application performance.

### *2.2 Data annotation*

The ideal tea shoots for Pu'er tea production consist of one bud and three leaves, as illustrated in Figure 2. The images were manually annotated using LabelImg software, with samples that had insufficient pixel area or unclear images being excluded. Additionally, tender shoots with over 80% occlusion or less than 15% visibility were not annotated. The annotations were initially saved in XML format. However, as YOLOv8 necessitates annotations in TXT format, specifying 'object-class' for the category label, 'x' and 'y' for the centre coordinates, and 'w' and 'h' for the object's width and height, a Python script was utilised to transform the XML files into the appropriate TXT format.

### *2.3 Data augmentation*

Data augmentation techniques were applied to diversify the training data, bolster the model's generalisation capabilities, and enhance its robustness, thereby expanding the dataset. Common data augmentation methods in image datasets include brightness enhancement, contrast enhancement, image rotation, flipping, affine transformations, shearing, HSV augmentation, translation, and the addition of random noise. In this study, affine transformations were applied to simulate changes in camera position and angle, brightness adjustments were made to mimic variations in sunlight, Gaussian noise was randomly added, and random cropping was performed. Consequently, the dataset was enlarged to comprise 4,655 images, which were subsequently split into training, validation, and test sets in the proportion of 7:2:1.

**Figure 2** Pu'er tea tender shoot (one bud and three leaves) (see online version for colours)

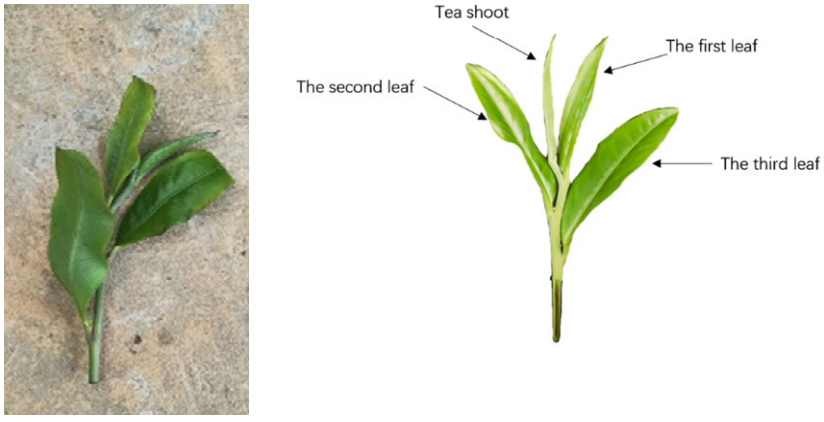
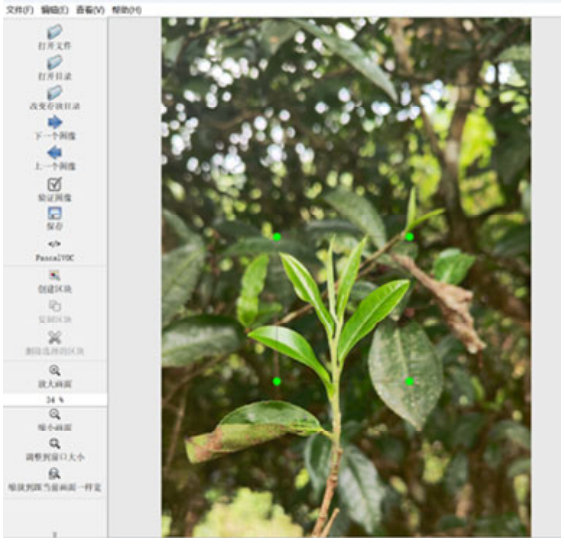



**Figure 3** Data annotation (see online version for colours)



## 3 Object detection model and improvements

YOLO is a target detection algorithm whose core concept is that 'you only need to look at the image once' to recognise all targets in the image. By segmenting the image into grids and predicting directly the location, class, and confidence level of targets within each grid, this algorithm significantly enhances detection speed and real-time performance. Currently, the most commonly used YOLO models are YOLOV5 and YOLOV8. YOLOV5 is renowned for its lightweight model, high speed, and active community support, making it well-suited for real-time object detection on devices with limited resources. Building upon this foundation, YOLOV8 has achieved significant

improvements in accuracy and speed, while also enhancing robustness and flexibility, particularly in its ability to adapt to complex backgrounds and diverse scenarios. Therefore, it is highly suitable for object detection in tea picking applications.

## *3.1 YOLOv8 model*

YOLOv8, developed by Ultralytics, is the iteration of the YOLO object detection and image segmentation model, designed to support various tasks such as detection, classification, and segmentation. The YOLOv8 architecture comprises three key components: the backbone, neck, and head. The backbone is responsible for feature extraction, utilising a blend of convolutional and deconvolutional layers, alongside residual connections and bottleneck designs to minimise the network's size while enhancing its performance. Unlike YOLOv5, in the backbone, YOLOv8 utilises the C2f module as its fundamental unit, which provides fewer parameters and superior feature extraction capabilities in comparison to the C3 module. The neck is responsible for fusing multi-scale features, which improves feature representation by integrating feature maps from various stages of the backbone. In YOLOv8, the neck incorporates the spatial pyramid pooling fast (SPPF) module, which enhances detection capabilities for objects of varying sizes by executing multi-scale pooling operations and concatenating feature maps of different scales. The Head is responsible for the final tasks of object detection and classification, comprising a detection head and a classification head. The detection head employs a sequence of convolutional and deconvolutional layers to produce detection outputs, predicting bounding box regression values and object confidence scores for each anchor box. The classification head applies global average pooling (GAP) to classify the feature maps, reducing their dimensionality and producing a probability distribution for each class. Additionally, YOLOv8 has shifted from an anchor-based to an anchor-free detection approach.

YOLOv8 offers models of different scales, such as nano, small, medium, large, and extra large, to suit various hardware platforms and application scenarios. By adjusting parameters such as network depth and width, these models attain a favourable equilibrium between detection accuracy and computational complexity.

## *3.2 Model lightweight improvements*

### *3.2.1 Tea shoot detection model improvements*

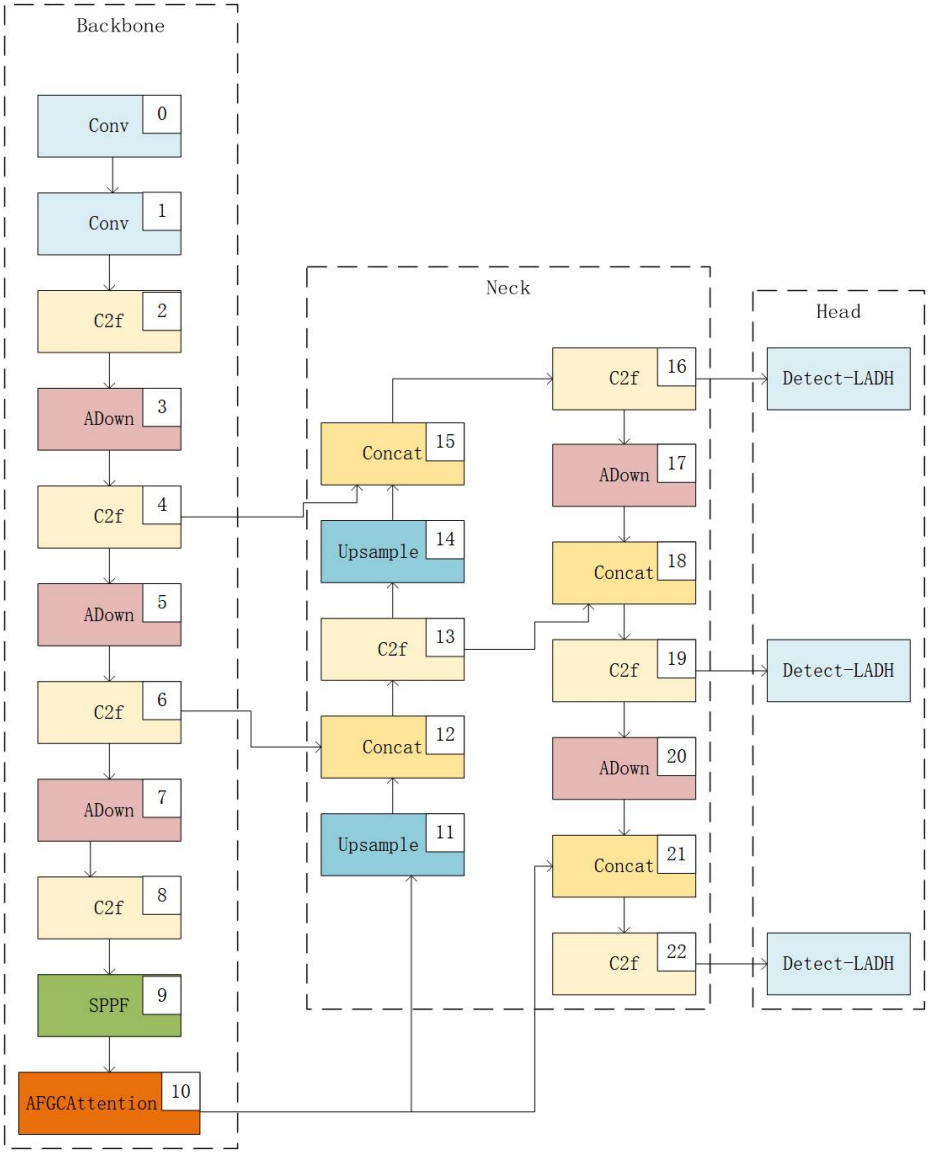
Adown (Wang et al., 2024) is an innovative downsampling module introduced in YOLOv9, optimised for target detection accuracy and efficiency through lightweight design and learning capabilities. In this study, the Adown downsampling module was used to replace certain Conv modules. Additionally, the AFGC Attention mechanism was incorporated for further lightweight design, and the lightweight asymmetric multi-level compression LADH detection head replaced the original detection head.

### *3.2.2 ADown downsampling module*

Downsampling is a prevalent technique utilised in deep learning models to decrease the spatial dimensions of feature maps, aiding the model in capturing higher-level image features while mitigating computational complexity. The Adown downsampling module

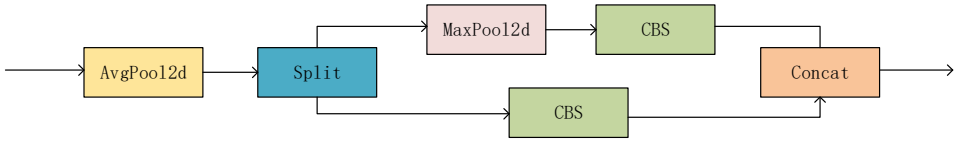
uses the AvgPool2d pooling function to downsample feature maps by setting the pooling window size, halving the input data's dimensions. The output values are then split into new subsets, with the first subset processed by convolutional layer cv1 (with an input channel count halved from the original input and an output channel count halved from the final output, using a 3×3 kernel, stride of 2, and padding of 1). The second subset undergoes MaxPool2d, followed by processing through convolutional layer cv2 (with the same input and output channel counts as cv1 but with a 1×1 kernel, stride, and padding of 0). The two data subsets are ultimately merged along the channel dimension, yielding the final result. The main structure is shown in Figure 5.

**Figure 4** Improved YOLOv8n model (see online version for colours)



The ADown module achieves lightweight and real-time detection by decreasing the parameter count, thereby reducing the model's complexity. It emphasises preserving as much image information as feasible, enabling the model to detect targets with greater accuracy and enhancing detection precision. In the Backbone, ADown can be used for downsampling between different layers of the feature map, while in the neck; it helps further refine the feature map's resolution for more precise target detection. By replacing the traditional downsampling operations in YOLOv8n with the ADown module, the model's parameters are reduced, and detection accuracy and speed are improved.

**Figure 5** ADown downsampling structure (see online version for colours)



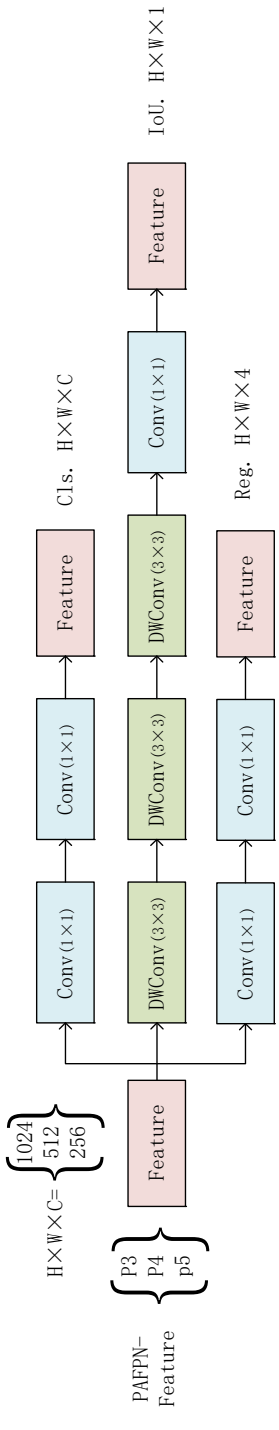

### *3.2.3 LADH detection head*

The YOLOv8 detection head has two branches, each extracting information through two 3×3 convolutions and one 1×1 convolution, ultimately calculating Bbox.loss and CLs.loss. To reduce the model's computational burden, the detection head of YOLOv8 was improved using the LADH detection head, which employs an asymmetric multi-level compression method. This approach applies different compression ratios to detection heads for different classes, better accommodating the feature representation and target size distribution of various classes, and enhancing the detector's generalisation ability.

The LADH detection head (Zhang et al., 2023; Huang et al., 2023) is a decoupled head that separates the network according to the specific task, using three distinct channels to handle bounding box regression and classification tasks. To enlarge the receptive field of the IoU branch and increase task-specific parameters, three convolutional layers are utilised. These layers primarily focus on compressing features along the channel dimension, which boosts the model's capability to gather wider spatial context information. This improves the accuracy of intersection over union (IoU) predictions for bounding boxes, permitting the network to more adeptly interpret and handle object locations and sizes within the input images, thereby boosting the overall performance in object detection or segmentation tasks.

In the LADH network, standard 3×3 convolutions are replaced with depth-wise separable convolutions (DWConv). DWConv (Patil et al., 2023) splits standard convolution into two steps: first, a depthwise convolution (convolution within each channel), followed by a pointwise convolution (1x1 convolution applied to all channels). This approach reduces computation while maintaining a certain degree of model complexity. The reduced parameter count in depthwise separable convolutions leads to higher computational efficiency, rendering it appropriate for application in environments with limited resources. Furthermore, by utilising 3×3 depthwise separable convolutions to separate the classification and bounding box tasks, we avoid excessive task inflation, given that positive samples generally exhibit smaller matching losses for both tasks. This approach further minimises the model's parameters, shrinks its size, and accelerates the detection speed.

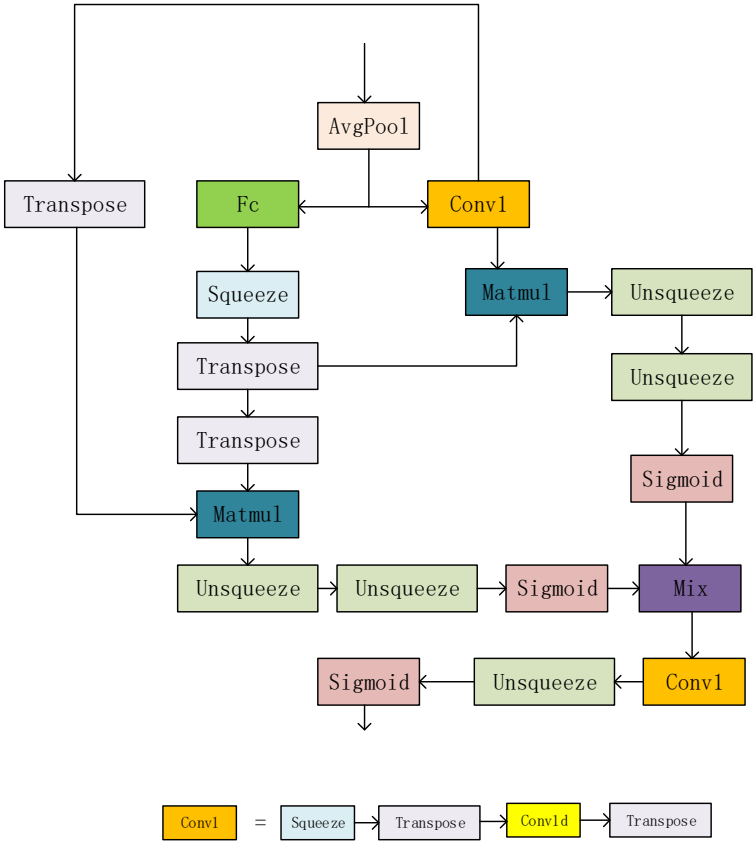
**Figure 6** LADH detection head structure (see online version for colours)



### 3.2.4 Incorporating the attention mechanism

The attention mechanism marks a substantial progress in deep learning, especially in the domains of image recognition and natural language processing (NLP), showcasing immense potential and worth. The core concept behind this mechanism is to emulate the human tendency to focus on important information while disregarding irrelevant details. In deep learning, the vast number of parameters accumulated by a model can enhance its ability to express features but often result in high computational demands, making it challenging to capture effective features. The attention mechanism tackles this issue by allowing the model to focus more intently on pivotal regions during image processing, like objects in detection assignments or prominent characteristics in classification tasks. By reducing the processing of irrelevant information, the attention mechanism greatly enhances the model's computational efficiency, enabling it to precisely recognise and capture vital patterns and features within the data, thus elevating the accuracy and effectiveness of task performance.

**Figure 7** AFGC structure (see online version for colours)




To improve the model's ability to learn features, this study integrates the AFGC mechanism (Sermanet et al., 2014) at the end of the backbone network. AFGC utilises GAP to condense each channel's feature map into a singular value, effectively achieving

a comprehensive receptive field spanning the entire input. These pooled channel features are then processed through a one-dimensional convolution to generate initial attention weights. The kernel size for this convolution is adaptively calculated based on the number of input channels and two custom parameters, allowing the model to dynamically adjust the attention mechanism's sensitivity in response to the complexity of the input features. The correlation between different channels is determined using a dot product operation, which produces the attention weights. These weights are scaled to a range between 0 and 1 through the application of the Sigmoid function, indicating the significance of each channel. The mix module is utilised to combine attention weights from various sources, resulting in a more comprehensive channel attention representation. Finally, the computed attention weights are applied as multipliers to the initial input feature map, enabling the amalgamation of weighted features. The strength of this attention mechanism lies in its adaptability and fine-grained channel adjustment capabilities, which enable the model to more effectively capture and emphasise relevant features when dealing with complex channel interactions, ultimately improving detection accuracy.

## 4 Results and analysis

### *4.1 Training environment*

The models in this study were trained on an Ubuntu 22.04 operating system with 64 GB of memory, using an NVIDIA GeForce RTX 4080 SUPER GPU, and powered by an Intel Core i9-14900KF CPU. The software used for compilation was PyCharm, and the framework employed was PyTorch 2.4.0, with Python version 3.12. GPU acceleration was achieved using Compute Unified Device Architecture (CUDA) toolkit 12.4 to enhance computational graphics processing capabilities. Based on preliminary experiments, we determined that 300 training epochs, a batch size of 16, and the SGD optimiser with a learning rate of 0.01, momentum of 0.937, and a weight decay coefficient of 0.0005 were appropriate. The dataset comprises 4,655 sheets, allocated to training, validation, and testing sets in a 7:2:1 ratio.

### *4.2 Evaluation metrics*

In this research, precision (P), recall (R), mean average precision (mAP), F1-score (Ultralytics, 2023), frames per second (FPS), and model size (Ms) are utilised as critical evaluation metrics to analyse the performance of the detection models. These metrics collectively offer a thorough evaluation of the models' efficiency and practical applicability from multiple perspectives.

Precision (P) quantifies the ratio of correctly predicted positive samples to the total number of samples classified as positive by the model. Recall (R), also known as the true positive rate, evaluates the proportion of actual positive samples that the model successfully detects as positive.

The F1-score represents the harmonic average of precision and recall, offering a holistic assessment of both precision and recall capabilities.

mAP serves as a metric to evaluate the performance of classification or detection tasks across various categories. It starts with computing the average precision (AP) for

each category individually, often achieved by plotting the precision-recall curve and calculating the area underneath. The ultimate mAP score is derived by averaging the AP values across all categories, offering a comprehensive assessment of the model's overall effectiveness.

After the training runs, the validation phase infers on the test set and compares the IoU (intersection and union ratio) between each prediction frame and the real labels, determining which are TPs, FPs or FNs based on a set threshold. The values of precision and recall are then derived from the summary statistics based on the definitions above. For example, if TP = 887, FP = 113, FN = 192 then P = TP/(TP + FP) = 0.887, R = TP/(TP + FN) = 0.822. Next, by applying different confidence thresholds, a precision-recall (PR) curve can be plotted, where each point corresponds to the precision and recall obtained at a specific confidence threshold. The AP value for each class is determined by computing the area underneath the PR curve. Finally, the mean of the AP values across all classes is computed to obtain the mAP value.

FPS is a vital measure for assessing a model's capability in real-time video processing or analysis tasks. It measures how many frames the model can process per second, which directly affects the model's responsiveness and the overall user experience in practical applications. The higher frame rate (FPS) signifies that the model is able to process video data more swiftly, making it ideal for situations where real-time processing is essential.

Model size (Ms) denotes the storage space required by the model, generally measured in megabytes (MB) or gigabytes (GB). A smaller model size simplifies deployment and transmission, particularly on devices that have constrained resources, like mobile phones or embedded systems. Consequently, balancing high performance with model compactness is crucial for efficient use and deployment.

In summary, these metrics collectively form a comprehensive framework for evaluating the detection performance of models, considering accuracy, efficiency, practicality, and scalability in real-world applications. To provide a more holistic evaluation of the models, this study introduces the min-max normalisation method (Qiu et al., 2023) for scoring. This method maps each metric to a range of 0 to 1 and assigns different weights to the metrics, using weighted scores to evaluate the models comprehensively. For this study, the mobile application requires the model to achieve detection tasks with a smaller model weight while maintaining high accuracy and achieving faster detection speeds. Based on this requirement, weights of 0.1 are assigned to P, 0.15 to both mAP and R, 0.3 to FPS, and –0.3 to model size (Ms) (as smaller models are advantageous for mobile deployment). The weighted score for each model is calculated as follows:

$$S_w = 0.1\frac{P - P_{\min}}{P_{\max} - P_{\min}} + 0.15\frac{mAP - mAP_{\min}}{mAP_{\max} - mAP_{\min}} + 0.15\frac{R - R_{\min}}{R_{\max} - R_{\min}} + 0.3\frac{FPS - FPS_{\min}}{FPS_{\max} - FPS_{\min}} - 0.3\frac{Ms - Ms_{\min}}{Ms_{\max} - Ms_{\min}}$$

where $S_w$ is the weighted score for each model, $mAP_{\min}$, $P_{\min}$, $R_{\min}$, $FPS_{\min}$, and $Ms_{\min}$ are the minimum values of the corresponding metrics among the models being compared, and $mAP_{\max}$, $P_{\max}$, $R_{\max}$, $FPS_{\max}$, and $Ms_{\max}$ are the maximum values of the corresponding metrics.

### *4.3 Ablation experiment analysis of network structure*

To ascertain the efficacy of the enhancements made to this algorithm for detecting tender shoots of Pu'er tea leaves, an ablation experiment was designed. Under consistent training methods and environments, the ADown downsampling module, LADH detection head, and AFGC Attention mechanism were sequentially added to the original YOLOv8n algorithm. The outcomes of the ablation studies are presented in Table 1.

**Table 1** Ablation experiment results and weighted scores for each model

| *Model* | *Adown* | *LADH* | *AFGCAttention* | *P(%)* | *R(%)* |
|---|---|---|---|---|---|
| YOLOv8n | × | × | × | 88.70 | 82.20 |
| YOLOv8n-A | √ | × | × | 86.70 | 84.45 |
| YOLOv8n-AL | √ | √ | × | 86.80 | 83.30 |
| YOLOv8n-ALAt | √ | √ | √ | 89.10 | 81.70 |
| | *MAP(%)* | *FPS(F/s)* | *Ms(MB)* | *Sw* | |
| YOLOv8n | 89.90 | 909.3 | 6.3 | –0.06 | |
| YOLOv8n-A | 90.60 | 1,359.65 | 5.4 | 0.33 | |
| YOLOv8n-AL | 89.40 | 1,412.32 | 4.2 | 0.37 | |
| YOLOv8n-ALAt | 90.60 | 1,392.2 | 4.4 | 0.53 | |

Note: '×' indicates that the module is not used, while '√' indicates that the module is used.

Referring to the results of the ablation experiments presented in Table 1, it can be concluded that replacing the original Conv module with the Adown downsampling module in the YOLOv8n model improves the AP in a 0.7% increase, increases detection speed of 450.35F/S, and reduces model size to 5.4MB. On this basis, replacing the original detection head with the LADH detection head slightly decreases the AP by 1.2%, but further increases detection speed to 1412.32, and reduces model size to 4.2MB. Finally, incorporating the attention mechanism at the backbone's termination enhances both precision and AP, with precision increasing by 2.3% and AP by 1.2%. Compared to the YOLOv8n model, the YOLOv8n-ALAt model, which integrates all three enhancements, demonstrates a 0.7% improvement in mAP, a boost in detection speed by 482.9 FPS, and a reduction in model size by 1.9MB. The weighted scores in Table 1 indicate that the YOLOv8n-ALAt model, with all three modifications, achieves the highest score, making it the chosen model for detecting tender shoots of Pu'er tea leaves.

### *4.4 Comparative experiment analysis*

To further evaluate performance, comparative experiments were conducted using the mainstream models from the YOLO series. The outcomes are presented in Table 2, with the trends in mAP, P, and R for each model illustrated in Figure 7.

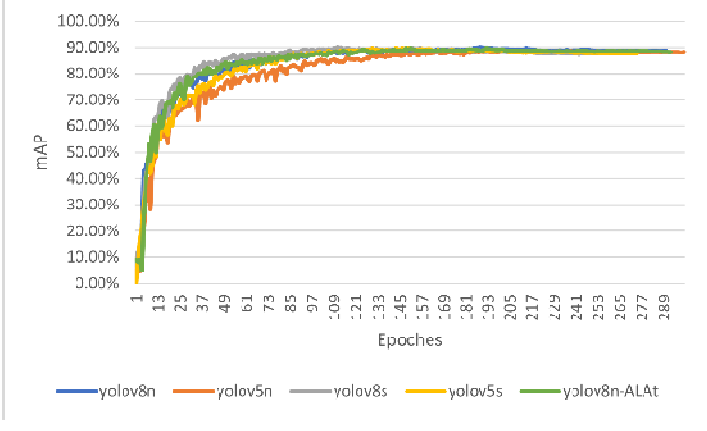
The data outlined in Table 2 reveal that, in terms of model lightweighting, the YOLOv8n-ALAt model achieves a substantial decrease in both model parameters and computational load by optimising the downsampling and detection head of the YOLOv8n model, as well as incorporating an attention mechanism to boost detection accuracy. As a result, the improved model's parameter count is significantly reduced, with the

computational load reduced to only 5.0 GFLOPs, and the model's weight reduced to just 4.4 MB, while inference speed is significantly improved. Compared to YOLOv5s and YOLOv8s, the YOLOv8n-ALAt model reduces the number of parameters by 77.65% and 81.69%, respectively, decreases the computational load by 79.00% and 82.39%, increases the AP by 1.6% and 1.1%, boosts real-time detection speed by 50.99% and 62.47%, and reduces the model weight by 76.22% and 80.44%, respectively, rendering it more appropriate for deployment on mobile devices with lightweight requirements.

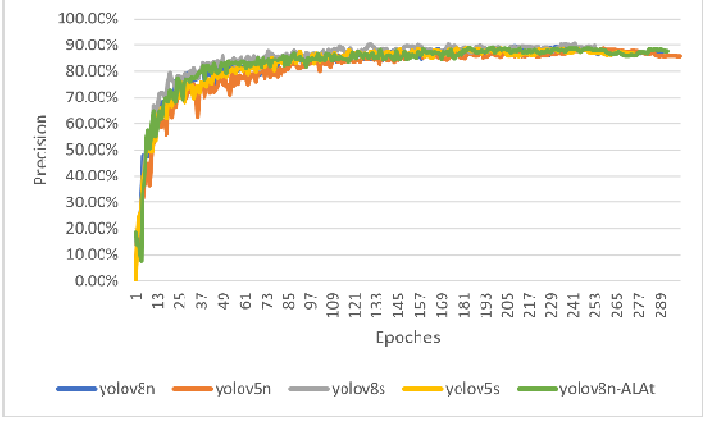
**Table 2** Comparative experiment results

| *Model* | *Param* | *FLOPS* | *P(%)* | *R(%)* | *mAP(%)* | *FPS(F/s)* | *Ms(MB)* |
|---|---|---|---|---|---|---|---|
| YOLOv5s | 9,111,923 | 23.8G | 88.2 | 82.1 | 89 | 922.08 | 18.5 |
| YOLOV8n | 3,005,843 | 8.1G | 88.7 | 82.2 | 89.9 | 909.3 | 6.3 |
| YOLOV8s | 11,125,971 | 28.4G | 89.4 | 82.2 | 89.5 | 856.92 | 22.5 |
| YOLOV8n-ALAt | 2,036,889 | 5.0G | 89.1 | 81.7 | 90.6 | 1,392.2 | 4.4 |

**Figure 8** mAP, P, R and F1-score curves of different models (see online version for colours)

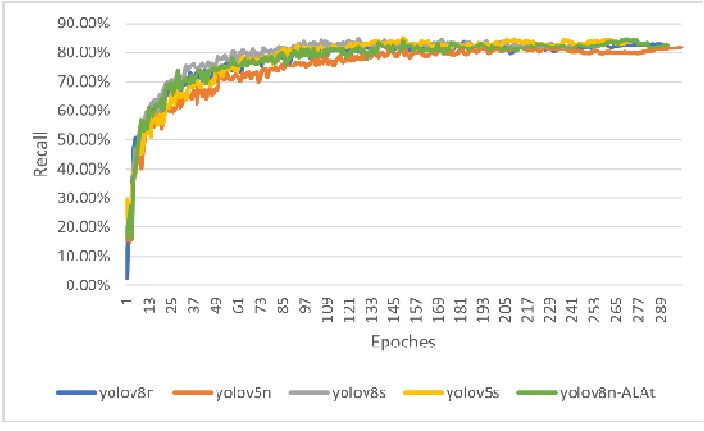



(a)

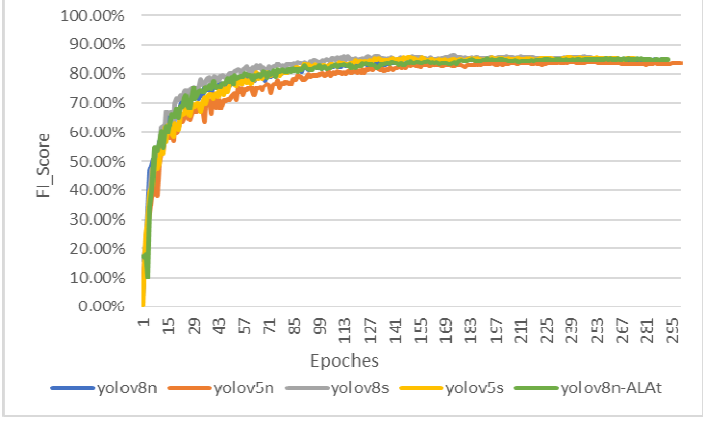



(b)

**Figure 8** mAP, P, R and F1-score curves of different models (continued) (see online version for colours)




(c)




(d)

Figure 8 shows that the mAP, P, R and F1 curves of the improved YOLOv8n-ALAt model exhibit minimal fluctuation and a steady increase, indicating that the model's structure is stable.

## 5 Conclusions

In this study, improvements were made to the YOLOv8n object detection model by replacing the original convolutional and detection head modules and introducing an attention mechanism into the backbone, which decreased the number of parameters and computational burden of the model, lowered its weight, and made it easier to deploy on mobile devices.

Through ablation and comparative experiments, the following conclusions were drawn:

1 Replacing the Conv module with Adown downsampling improves detection accuracy, significantly speeds up detection, and reduces model weight. Switching to the LADH detection head significantly cuts down on the model's parameter count, thereby further reducing its weight. The incorporation of the AFGC attention mechanism helps to counteract the decrease in detection accuracy that often accompanies model lightweighting. In comparison to the YOLOv8n model, the optimised version exhibits a 0.7% improvement in mAP, a boost of 482.9 FPS in detection speed, and a model size reduction of 1.9 MB.

2 Compared to the YOLOv5s and YOLOv8s models, the YOLOv8n-ALAt model decreases the number of parameters by 77.65% and 81.69%, decreases computational load by 79.00% and 82.39%, increases AP by 1.6% and 1.1%, boosts real-time detection speed by 50.99% and 62.47%, and reduces the model weight by 76.22% and 80.44%, respectively. The improved model is more suitable for resource-constrained mobile deployments.

3 The optimised YOLOv8n-ALAt model is stable, with high detection accuracy, fast speed, and a small model weight, enabling precise data processing of Pu'er tea tender shoots based on visual data. This provides accurate data support for subsequent research on Pu'er tea harvesting.

In brief, this research aims to enhance the accuracy and real-time performance of tea shoot detection by refining the YOLOv8n model, addressing two key areas of interest in the domain of object detection. Our findings indicate that increasing accuracy often necessitates increasing model complexity, implying more parameters and a more intricate network structure, subsequently leading to a substantial increase in computational load and storage requirements, as well as a corresponding decrease in inference speed. However, to enhance real-time performance, reducing model size and parameter count can significantly boost inference speed, decrease computational burden, and to render the model more suitable for deployment on edge devices with limited resources. Additionally, decreasing model complexity may also impact the model's reliability and generalisation capability. While a judicious reduction in the number of parameters can aid in preventing overfitting, bolster the model's generalisation ability, and diminish redundancy and complexity, thus mitigating the risk of errors, an excessive decrease in parameters may hinder the model's capacity to adequately capture data features, adversely impacting its generalisation performance and potentially causing instability in particular scenarios. Therefore, the optimisation objective should be rationally selected based on the application scenario of the object detection model, with careful performance evaluation to ensure that high accuracy and reliability are maintained while meeting practical application requirements for real-time performance and generalisation capability.

Future research directions include deploying this model on smart devices such as Pu'er tea harvesting robots or drones to further expand the application of intelligent, mechanised harvesting of Pu'er tea. Following this, the object detection model will be integrated with tasks such as keypoint detection, classification, and segmentation to boost the model's overall performance and generalisation capabilities, catering to diverse agricultural application scenarios.

## Acknowledgements

This paper is supported by Yunnan Provincial Department of Education Scientific Research Fund Project (2023J0712).

## Declarations

All authors declare that they have no conflicts of interest.

## References

Chen, M. (2019) *Identification and Localization of Premium Tender Tea Shoots Based on Computer Vision*, Qingdao University of Science and Technology, No. 11.

Endo, K., Hiraguri, T., Kimura, T. et al. (2024) 'Estimation of the amount of pear pollen based on flowering stage detection using deep learning', *Scientific Reports*, Vol. 14, No. 1, p.13163.

Fang, M., Lv, J., Ruan, J., Bian, L., Wu, C. and Yao, Q. (2022) 'Tea bud detection model based on improved YOLOv4-tiny', *Journal of Tea Science*, Vol. 4, No. 4, pp.549–560.

He, H. (2023) *Design and Implementation of a Tea Bud Detection System Based on Deep Learning*, Sichuan Agricultural University.

Huang, J., Tang, A., Chen, G., Zhang, D., Gao, F. and Chen, T. (2023) 'Mobile end recognition method for tea buds based on compact-YOLOv4', *Transactions of the Chinese Society of Agricultural Machinery*, Vol. 3, No. 3, pp.282–290.

Huang, L., Li, W., Shen, L., Fu, H., Xiao, X. and Xiao, S. (2023) *YOLOCS: Object Detection based on Dense Channel Compression for Feature Spatial Solidification*, arXiv, arXiv:abs/ 2305.04170.

Lv, F., Wang, X. and Li, L. (2024) 'Tree detection algorithm based on embedded YOLO lightweight network', *Journal of Shanghai Jiaotong University (Science)*, Vol. 29, No. 3, pp.518–527.

Mylonas, A, Keall, P.J., Booth, J.T. et al. (2019) 'A deep learning framework for automatic detection of arbitrarily shaped fiducial markers in intrafraction fluoroscopic images', *Medical Physics*, Vol. 46, No. 5, pp.2286–2297.

Patil, P.S., Bhosale, P.R., Holambe, R.S. and Waghmare, L.M. (2023) 'SDDSCNet: Siamese-based dilated depthwise separable convolution neural network with wavelet fusion for change detection', in Doriya, R., Soni, B., Shukla, A. and Gao, X.Z. (Eds.): *Machine Learning, Image Processing, Network Security and Data Sciences. Lecture Notes in Electrical Engineering*, Vol. 946, Springer, Singapore, https://doi.org/10.1007/978-981-19-5868-7_13.

Qiu, Y., Gu, J., Pan, S. et al. (2023) 'Deep learning-based target detection and visual navigation for intelligent vehicles', *Light Industry Science and Technology*, Vol. 39, No. 4, pp.107–110.

Sermanet, P., Frome, A. and Real, E. (2014) 'Attention for fine-grained categorization', *Computer Science*, Vol. 10, No. 1, pp.224–30.

Ultralytics (2023) *Ultralytics YOLOv8: GitHub Repository* [2023-10-20] [online] https://github.com/ultralytics/ultralytics (accessed 20 October 2023).

Wang, C-Y., Yeh, I-H. and Liao, H-Y.M. (2024) *YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*, arXiv:2402.13616.

Xu, H. (2021) *Research on Technology Path Selection of Yunnan Pu'er Tea Industry*, Kunming University of Science and Technology, No. 5.

Xu, H. (2023) *Research on Tender Tea Shoots Detection Method Based on Deep Learning*, Zhejiang Sci-Tech University.

Xu, W., Zhao, L., Li, J. et al. (2022) 'Detection and classification of tea buds based on deep learning', *Computers and Electronics in Agriculture*, Vol. 192, p.106547, ISSN 0168-1699.

Xu, Y. (2013) 'A survey of Yunnan Pu'er tea farmers', *Tea*, Vol. 2013, No. 1, pp.24–27.

Yang, H. et al. (2019) 'Tender tea shoots recognition and positioning for picking robot using improved YOLO-V3 model', in *IEEE Access*, Vol. 7, pp.180998–181011, DOI: 10.1109/ACCESS.2019.2958614.

Yin, C., Su, Y., Pan, M. and Duan, J. (2023) 'Quality detection of premium green tea based on improved YOLOv5s', *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 39, No. 8, pp.179–187.

Yunnan Provincial Department of Agriculture and Rural Affairs (2021) *Yunnan Province Tea Industry Development Report* (2022-7-30) [online] https://nync.yn.gov.cn/html/2022/nongyechanyebaogao_0729/389286.html?cid=4978 (accessed 30 July 2022).

Zhang, J., Chen, Z., Yan, G., Wang, Y. and Hu, B. (2023) 'Faster and lightweight: an improved YOLOv5 object detector for remote sensing images', *Remote Sens.*, Vol. 15, No. 20, p.4974.

Zhang, X., Hu, G., Li, P. et al. (2024) 'Recognizing safflower using improved lightweight YOLOv8n', *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 40, No. 13, pp.163–170.

Zhu, H., Zhang, Y., Mu, D. et al. (2024) 'Research on improved YOLOx weed detection based on lightweight attention module', *Crop Protection*, March, Vol. 177, p.106563, ISSN 0261-2194.