agriculture

Article

A Lightweight Pest Detection Model for Drones Based on Transformer and Super-Resolution Sampling Techniques

Yuzhe Bai [1,†], Fengjun Hou [1,†], Xinyuan Fan [1], Weifan Lin [1], Jinghan Lu [1], Junyu Zhou [1], Dongchen Fan [2] and Lin Li [1,*]

[1] China Agricultural University, Beijing 100083, China; byz0871@cau.edu.cn (Y.B.); fxy07dr@cau.edu.cn (X.F.); lujinghan2019@cau.edu.cn (J.L.); cau_zhoujy@163.com (J.Z.)
[2] School of Computer Science and Engineering, Beihang University, Beijing 100191, China; 213352411@buaa.edu.cn
* Correspondence: lilinlsl0726@cau.edu.cn
† These authors contributed equally to this work.

Abstract: With the widespread application of drone technology, the demand for pest detection and identification from low-resolution and noisy images captured with drones has been steadily increasing. In this study, a lightweight pest identification model based on Transformer and super-resolution sampling techniques is introduced, aiming to enhance identification accuracy under challenging conditions. The Transformer model was found to effectively capture spatial dependencies in images, while the super-resolution sampling technique was employed to restore image details for subsequent identification processes. The experimental results demonstrated that this approach exhibited significant advantages across various pest image datasets, achieving Precision, Recall, mAP, and FPS scores of 0.97, 0.95, 0.95, and 57, respectively. Especially in the presence of low resolution and noise, this method was capable of performing pest identification with high accuracy. Furthermore, an adaptive optimizer was incorporated to enhance model convergence and performance. Overall, this study offers an efficient and accurate method for pest detection and identification in practical applications, holding significant practical value.

Keywords: smart agriculture; pest detection; Transformer; super resolution

Citation: Bai, Y.; Hou, F.; Fan, X.; Lin, W.; Lu, J.; Zhou, J.; Fan, D.; Li, L. A Lightweight Pest Detection Model for Drones Based on Transformer and Super-Resolution Sampling Techniques. *Agriculture* **2023**, *13*, 1812. https://doi.org/10.3390/agriculture13091812

Academic Editors: Maciej Zaborowicz and Jakub Frankowski

Received: 20 August 2023
Revised: 5 September 2023
Accepted: 11 September 2023
Published: 14 September 2023

1. Introduction

With the continuous advancement of agricultural technology, drones have been progressively adopted as efficient automation tools in various agricultural operations [1], including sowing, fertilization, and monitoring. In particular, for crop health monitoring, drones have demonstrated immense potential and value. Pests, as one of the primary threats in agricultural production, pose serious risks to crop health. While pesticides can address some pest issues [2], timely and effective pest detection remains paramount for pest prevention and control.

Traditional pest detection methods often rely on manual inspections [3] and solar tracking [4]. Not only these methods exhibit low efficiency, but also their accuracy is constrained by human experience and the intensity of manual labor, leading to potential oversights. Furthermore, the frequency and scope of manual inspections are limited, preventing extensive, high-frequency pest monitoring, especially given the small size of pests [5]. This limitation can result in missing optimal opportunities for prevention and control during the initial stages of pest outbreaks. While pheromone-based pest detection methods exist [6], they are specific to particular pests [7], offering limited versatility.

The rapid advancement of computer vision technology in recent years has introduced new avenues for smart agriculture [8–11]. Through image recognition and deep learning techniques, high-efficiency and accurate identification of pests can be achieved [12]. In this realm, researchers from various countries have embarked on several investigations. Liang,

Quanjia, developed a rice pest recognition model based on an improved YOLOv7 algorithm. By employing the lightweight MobileNetV3 network for feature extraction, the accuracy of 92.3% was achieved on a dataset containing 3773 images of rice pests [13]. Yang, Zijia, and colleagues compiled an image dataset of eight tea tree pests and designed a pest detection and recognition model for tea gardens using the Yolov7-tiny network. By integrating deformable convolutions, the Biformer dynamic attention mechanism, the non-maximum suppression algorithm module, and a new implicit decoupling header, the average accuracy of 93.23% was achieved [14]. Jia, Xinyu, and team established a dataset consisting of 5182 pest images across 14 categories. Using transfer learning, visual geometric group (VGG), residual neural network (ResNet), and a mobile network, citrus pest recognition models were created. Following this, appropriate attention mechanisms were introduced based on model characteristics. Ultimately, average recognition accuracy, Precision, Recall, and F1 score were 93.65%, 93.82%, 93.65%, and 93.62%, respectively [15]. Irjak, Dana, and others developed a DNN-based automatic monitoring system for apple codling moths, comprising a smart trap and an analysis model. Evaluation using a confusion matrix revealed an accuracy exceeding 99% in detecting apple codling moths [16].

Building on previous research, enhancements have been made. Kumar, Nithin, and associates utilized YOLOv5 and incorporated channel and spatial attention modules, enhancing network recognition capabilities. Experimental results showed that with learning on a custom pest dataset, the F1 score approached 0.90, and the mAP value reached 93%. In comparison to other YOLOv5 models, the F1 score increased by 0.02 [17]. Ullah, Zahid, and collaborators proposed the fusion of two pre-trained models, EfficientNetB3 and MobileNet. They also applied techniques such as regularization, dropout, and batch normalization to address model overfitting. The hybrid model achieved a success rate of 99.92% in accurately detecting tomato leaf diseases, proving its capability to extract features effectively [18]. Butera, Luca, and colleagues investigated the capabilities of state-of-the-art (SoA) object detection models based on convolutional neural networks (CNNs) to detect coleopteran pests on heterogeneous outdoor images from various sources, presenting a benchmark model. Results indicated that this combination delivered the Average Precision of 92.66% [19]. Kumar Yadav, P., and co-researchers employed drone-acquired RGB images to detect VC plants in maize fields. Findings showed that YOLOv3 could identify VC plants in maize fields with average detection accuracy above 80%, F1 score of 78.5%, and mAP of 80.38%. Regarding image sizes, no significant differences were observed in mAP across three scales. However, significant differences were found in AP between S1 and S3 ($p = 0.04$), and S2 and S3 ($p = 0.02$). Significant differences in F1 score were also seen between S2 and S3 ($p = 0.02$) [20]. Rong, Minxi, and group enhanced the FPN structure in the feature extraction network and introduced weight coefficients when merging features of different scales. Experimental analysis on 1000 sample images indicated that the improved Mask R-CNN model achieved recognition and detection accuracy of 99.4%, which is 2.7% higher than the unimproved Mask R-CNN model [21].

However, most existing computer vision models demand significant computational resources and exhibit considerable size [22,23], making them unsuitable for direct deployment on drone platforms with limited computational capabilities. Moreover, images captured with drones during flight are often affected by factors such as lighting, distance, and angle, potentially compromising image clarity and recognition accuracy. Hence, the challenge and focal point of current research lie in achieving efficient and accurate pest recognition within constrained computational resources [10].

In response to these challenges, this study introduces a novel pest recognition model based on the Transformer architecture combined with super-resolution sampling techniques, aiming to enhance the recognition accuracy and speed on drone platforms. Initially, through the super-resolution sampling module, high-resolution images with improved clarity can be reconstructed from low-resolution original images, thus enhancing recognition accuracy. Simultaneously, by employing model lightweighting techniques, computational demands and model size are significantly reduced, enabling real-time operation on drone

platforms. Additionally, adaptive optimizers are integrated to further improve model training efficiency and stability. Overall, this study offers a pioneering, drone-compatible pest recognition approach, holding substantial practical significance for pest prevention and control in agriculture and paving the way for potential applications of drones in the agricultural domain.

2. Related Work

In recent years, significant progress has been achieved in pest detection technologies. Notably, techniques related to deep learning have shown outstanding performance in image processing and model optimization [22–27]. This section primarily discusses three technologies: the Transformer architecture, super-resolution sampling modules, and model lightweighting techniques.

2.1. Transformer

The Transformer architecture [22] was initially designed for natural language processing tasks, addressing sequence-to-sequence tasks with its self-attention mechanism. The core idea behind the self-attention mechanism is that during processing, different attention weights can be given to different parts of the input data. This method allows the model to adaptively adjust its structure based on data content, capturing intrinsic features more effectively. Mathematically, self-attention can be expressed as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V represent the query, key, and value, respectively. They are typically linear transformations of the input data, while d_k denotes the dimension of the key.

Although the origins of the Transformer model lie in text data processing, it was quickly discovered that it could be applied to computer vision tasks. For instance, to adapt it for image data, one approach involves dividing an image into a series of fixed-size patches, then flattening these patch pixel values into vectors. Each patch can then be considered an element in a sequence. Based on this, Vision Transformer (ViT) [28] was introduced. This model divides the image into fixed-size patches, linearly embeds each patch into a fixed-size vector, and adds positional encoding to retain spatial information. When exploring how to apply the Transformer model to object detection tasks, a basic strategy involves segmenting the image into patches, assigning category labels and bounding boxes to each patch, and then processing these patches using a Transformer model and learning inter-patch relationships with the self-attention mechanism. During the decoding phase, another Transformer network receives the outputs from the encoding phase, generating category labels and bounding boxes for each patch. This can be represented as

$$O = \text{Transformer-Decoder}(\text{Transformer-Encoder}(I)) \tag{2}$$

where I represents the input image and O represents the output categories and locations. This application of the Transformer model to object detection offers advantages. Its global self-attention mechanism can capture long-range dependencies in images. Objects in images often have complex relationships with their surroundings, such as occlusions and interactions. The Transformer model can understand these relationships better, improving detection accuracy.

2.2. Super-Resolution Sampling

The aim of super-resolution sampling is to recover high-resolution details from low-resolution images. This is a popular research direction in the computer vision field, since it enhances image quality without the need for additional hardware. In particular, deep learning models have demonstrated remarkable performance in super-resolution tasks. SRGAN (Super-Resolution Generative Adversarial Network) [27] is a represen-

tative super-resolution model that uses a Generative Adversarial Network (GAN) [26] for super-resolution image restoration. Specifically, SRGAN comprises a generator and a discriminator. The generator is responsible for upsampling low-resolution images to high-resolution ones, while the discriminator attempts to distinguish between generated high-resolution images and real high-resolution images. Model training aims to minimize the difference between them and optimizes the following loss function:

$$L = L_{\text{content}} + \lambda L_{\text{adversarial}} \tag{3}$$

where L_{content} represents the content loss, usually computed using Mean Squared Error (MSE), and $L_{\text{adversarial}}$ represents the adversarial loss, which measures the difference between generated and real high-resolution images. The weight parameter, λ, balances their importance.

In computer vision tasks, the primary application of super-resolution technology is in image restoration and enhancement. Since collecting high-resolution images might be restricted by hardware or cost, super-resolution provides an effective solution for researchers and industries, extracting high-quality details from existing low-resolution images. When applied to object detection tasks, its main value lies in increasing image resolution, enabling more accurate detection of small or distant objects in images. Specifically, object detection typically involves feature extraction and bounding box regression. High-resolution images can provide richer information, making features more distinct in the feature extraction phase. In the bounding box regression phase, high-resolution images offer more accurate positional information, improving detection accuracy. To apply super-resolution in object detection, a super-resolution model can first upsample the input image, which is then fed into the object detection network. This method can be mathematically represented as

$$O_{\text{detection}} = \text{DetectionNetwork}(SR(I_{\text{low-res}})) \tag{4}$$

where $I_{\text{low-res}}$ denotes the input low-resolution image, $SR(\cdot)$ represents the super-resolution model, and $O_{\text{detection}}$ indicates the object detection output. The advantage of this method is that it not only enhances object detection accuracy but also allows detection models to achieve similar performance on low-resolution images as on high-resolution images. Additionally, since super-resolution models typically have fewer parameters, this method can effectively reduce the overall model size and computational cost.

2.3. Model Lightweighting

The technique of model lightweighting has garnered significant attention in the deep learning domain, as it facilitates the deployment of intricate models onto resource-constrained devices, such as mobile devices or edge computing equipment. The essence of model lightweighting is to not only retain the model's accuracy but also substantially reduce the model's size and computational load. Renowned model lightweighting techniques encompass knowledge distillation, network pruning, and quantization.

Knowledge distillation [29,30] serves as a technique to train a smaller model, utilizing the output of a larger model to guide the training of the smaller counterpart. Specifically, given a larger model (often termed the teacher model) and a smaller model (typically referred to as the student model), the aim of knowledge distillation is to approximate the student model's output to that of the teacher model as closely as possible. This can be mathematically expressed using the following loss function:

$$L_{\text{distill}} = \alpha L_{\text{original}} + (1 - \alpha) L_{\text{soft}} \tag{5}$$

where L_{original} represents the original loss function, such as cross-entropy loss, while L_{soft} denotes the loss between the outputs of the student and teacher models. Parameter α serves to balance these two losses.

Network pruning [31] is a technique aimed at reducing model size and computational load by eliminating certain portions of the neural network. The most prevalent method in

this context is weight pruning, which involves deleting certain weights from the model. This is typically conducted based on the magnitude or significance of the weights. For instance, given a threshold θ, weights with an absolute value less than θ can be deleted:

$$w_i' = \begin{cases} w_i & \text{if } |w_i| > \theta \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Quantization [32] is an approach to diminish the precision of model weights. As an example, 32-bit floating-point weight values can be quantized into 8-bit integers. This not only reduces the model's size but also accelerates its computations.

In computer vision tasks, the primary application of model lightweighting is to enhance model deployment efficiency. For object detection tasks, lightweighting the model can yield a higher frame rate for real-time applications or satisfactory performance on resource-limited devices. Specifically, for object detection models, a smaller student model can initially be trained using knowledge distillation, followed by further reduction in model size and computational load through network pruning. Finally, quantization can be employed to reduce the model's storage requirements and computational duration. Such a model lightweighting strategy offers an effective solution for object detection, ensuring efficient and accurate object detection even on resource-limited devices such as mobile devices or drones.

3. Materials and Method

3.1. Dataset Collection

In studies related to pest detection associated with crop health, the construction of datasets plays a pivotal role. The dataset collected encompasses various pests closely related to corn and rice, including *Spodoptera litura*, *Ostrinia furnacalis*, *Spodoptera frugiperda*, *Nilaparvata lugens*, *Cnaphalocrocis medinalis*, and *Leptocorisa chinensis*. The reasons for selecting these pests as the subjects of study are based on the severe threats they pose during the growth of corn and rice. For example, *Spodoptera litura* may damage the corn stalk, causing it to lodge; *Ostrinia furnacalis* and Spodoptera frugiperda directly harm corn leaves and ears, affecting the yield. As for rice, the emergence of pests like *Nilaparvata lugens* and *Cnaphalocrocis medinalis* often indicates a significant decline in yield [33].

The primary data collection site is located in West Science Park of China Agricultural University. Considering the actual crop growth environment, morning and evening were chosen as the primary collection times, as pest activity tends to be frequent during these periods. A 4K resolution camera (3840 × 2160) was employed as collection equipment to ensure the clarity and detail of the images obtained [11]. Moreover, a large number of pest images were scraped from the Internet [33]. By writing a crawler program, a vast amount of images related to these pests were gathered from various agriculture-related websites and communities. This approach allows for the rapid acquisition of substantial data, enriching the diversity of the dataset. The combination of both data collection methods ensures authenticity, reliability, diversity, and richness of the data. The dataset mirrors the various states of pests in real environments, laying a solid foundation for subsequent model training. The distribution of the dataset is shown in Table 1 and Figure 1.

Table 1. Distribution of the dataset used in this paper after preprocessing, discussed in Section 3.2.

Pest Type	Number of Images
Spodoptera litura	1200
Ostrinia furnacalis	1150
Spodoptera frugiperda	1100
Nilaparvata lugens	1250
Cnaphalocrocis medinalis	1000
Leptocorisa chinensis	1300

Figure 1. Samples of dataset used in this paper.

The construction of this dataset provides ample data support for subsequent model training and validation, ensuring the reliability and effectiveness of this research.

3.2. Dataset Preprocessing

In pest detection tasks, acquiring a substantial amount of high-quality training data is essential. However, data collection in real-world scenarios often encounters limitations due to factors like seasons, weather, and equipment, potentially leading to inadequate size and diversity of the initial dataset. Therefore, data preprocessing and augmentation techniques hold significance in such tasks. They not only enhance the model's adaptability to different environments and angles but also effectively mitigate the risk of overfitting, improving the model's generalization capabilities. Initially, image data augmentation, achieved by applying various transformations on the original images, exposes the model to a wider range of scenarios during training, thus enhancing its generalization ability. Various augmentation methods include rotation, flipping, cropping, brightness and contrast adjustment, and noise addition, as depicted in Figure 2.

Figure 2. Illustration of dataset preprocessing methods used in this paper, including flipping and mirroring.

Taking image rotation as an example, by rotating an image by a specific angle, a new image is obtained. The mathematical representation of this transformation can be expressed as

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{7}$$

where x and y represent the coordinates of the original pixel point, and x' and y' are the coordinates after rotation, with θ being the angle of rotation. Image flipping is another prevalent data augmentation method, flipping the image along a specific axis. Horizontal flipping can be represented as

$$x' = W - 1 - x, \quad y' = y \tag{8}$$

where W is the width of the image, x and y are the original pixel coordinates, and x' and y' are the new coordinates post-flipping. Image cropping involves selecting a region from the

original image to create a new one, aiding the model in focusing on various parts of the image. Random cropping can be expressed as

$$x' = x - \Delta x, \quad y' = y - \Delta y \tag{9}$$

where Δx and Δy represent the cropping offsets in the horizontal and vertical directions, respectively. Additionally, adjusting the brightness and contrast of images serves as an effective data augmentation method, which can be implemented using

$$I' = \alpha \cdot I + \beta \tag{10}$$

where I is the original image, I' is the enhanced image, α is the contrast adjustment factor, and β is the brightness adjustment factor. To combat noise and minor image variations, random noise can also be introduced into the images. Common noise models include Gaussian noise and salt-and-pepper noise. Using these augmentation methods, the diversity of the training set can be significantly increased, effectively preventing the model from overly relying on specific data distribution characteristics and enhancing its performance on unseen data. Furthermore, these methods simulate variations likely encountered in real-world applications, bolstering the model's robustness in actual scenarios.

3.3. Proposed Method

3.3.1. Overall

A novel pest identification model is proposed, designed for efficient and accurate pest detection for drones. The overall method framework consists of three main components: a Transformer-based object detection network, a super-resolution sampling module, and lightweight techniques. Each of these components is elaborated upon below, with an explanation of their integration into a cohesive workflow, as shown in Figure 3.

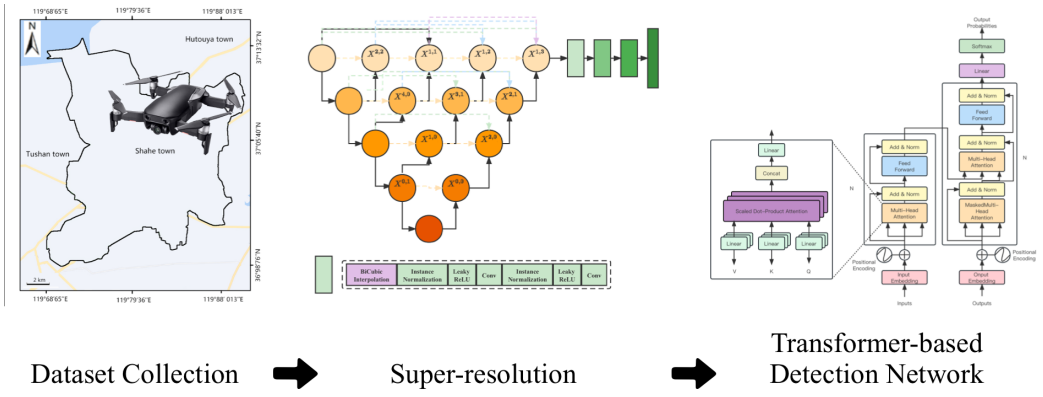

Figure 3. Illustration of the whole method proposed in this paper.

Initially, the Transformer-based object detection network serves as the backbone of the model, responsible for identifying pests in images [25]. The strength of the Transformer model lies in its self-attention mechanism, which captures long-range dependencies within images. In object detection tasks, the Transformer model can effectively differentiate between background and foreground, as well as identify relationships among multiple targets, which is pivotal in pest detection. However, images captured with drones can become blurred due to various factors, such as distance, lighting, and motion blur. To address this, a super-resolution sampling module was incorporated. Its primary role is to enhance image resolution, bringing out clearer details. By employing advanced deep learning methods, this module is capable of restoring low-resolution images to high-resolution ones while preserving intrinsic details. Prior to object detection, the super-resolution sampling module serves as a preprocessing step, supplying the Transformer network with crisper inputs, consequently improving detection accuracy. However, such a model may become extensive and computationally intensive. To mitigate this concern, lightweight techniques

were employed. These techniques encompass knowledge distillation, network pruning, and quantization and are capable of substantially reducing model size and computational demands without significantly compromising performance. With the incorporation of these lightweight techniques, the proposed model can operate in real time on drones, facilitating instantaneous pest detection. Integrating these three components, a comprehensive pest detection procedure emerges, as shown in Figure 3. Firstly, images captured with drones undergo preprocessing via the super-resolution sampling module, resulting in high-resolution outputs. Subsequently, these images are fed into the Transformer-based object detection network, yielding pest location and category information. Finally, lightweight techniques ensure efficient operation of the model on drones.

To achieve real-time pest detection on this drone, the model was chosen to run on NVIDIA's Jetson Nano platform [11]. Jetson Nano, a compact and energy-efficient computing platform, is particularly apt for edge computing. Possessing formidable graphics processing capabilities, it effortlessly manages the inferencing tasks of deep learning models. Crucially, its small size and low power consumption render it ideal for integration into mobile devices like drones. Additionally, to capture rich image details and ensure the model's precise pest detection capabilities, the drone was equipped with a 4K resolution camera. Such high-resolution cameras not only provide clear images but also capture minute details of pests, playing a pivotal role in enhancing detection accuracy. Once processed by the super-resolution sampling module, these 4K images can be further augmented, optimizing the Transformer network's performance. With the aforementioned hardware configuration, the overall method framework can efficiently and accurately detect pests on drones. Drones, using their 4K cameras, first capture images, which are then preprocessed on Jetson Nano by the super-resolution sampling module, resulting in high-resolution outputs. These images are subsequently fed into the Transformer-based object detection network for real-time inferencing on Jetson Nano, providing pest location and category details. Lightweight techniques guarantee the fluidity and efficiency of the entire procedure. In summary, the proposed method framework, integrating Transformer, super-resolution sampling, and lightweight techniques, forms a complete pest detection procedure. This approach, apart from efficient and accurate pest detection, also offers real-time operation on resource-constrained drones. It presents agriculture with a potent tool, aiding farmers in superior pest management, thereby enhancing crop yield and quality.

3.3.2. Super-Resolution Module

Super-resolution techniques aim to recover high-resolution images from low-resolution counterparts, thus revealing more details and improved clarity. This step proves crucial for pest detection, as adequate details must be captured to accurately identify and locate pests. The core of the super-resolution sampling module is grounded in convolutional neural networks. While conventional super-resolution methods, such as bicubic interpolation and Lanczos resampling, can somewhat augment image resolution, they fail to recover lost high-frequency details. However, convolutional neural networks are capable of learning methods to restore these nuances. For every low-resolution image input I_{LR}, the network is designed to produce a high-resolution output I_{HR}. Mathematically, the objective is to minimize the difference between the output image and the actual high-resolution image, represented as Mean Squared Error (MSE):

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n} \| I_{HR}^{(i)} - F(I_{LR}^{(i)}) \|_2^2 \tag{11}$$

where n stands for batch size and F represents the super-resolution model. A structure based on convolutional neural networks (CNNs) was developed, eschewing the Generative Adversarial Network (GAN) framework. Although GANs can produce visually satisfactory results, their demands for training stability and computational resources render them less suitable for real-time processing on mobile devices. The super-resolution model employed is founded on the classical ResNet [34] structure. To cater to super-resolution

tasks, adaptations and refinements were made. Specifically, a 20-layer deep network structure was employed, as shown in Figure 4.

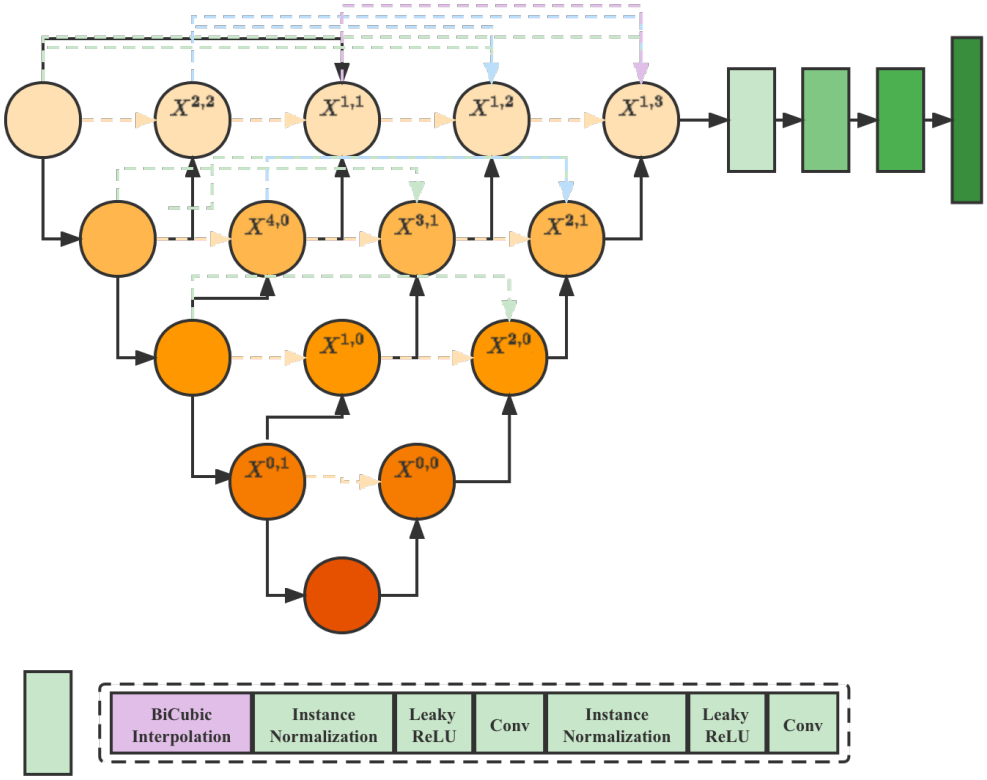

Figure 4. Structure of super-resolution module used in this paper.

This design, compared with deeper networks, has fewer parameters, which reduces computational and memory demands yet still achieves satisfactory super-resolution results. The model takes a low-resolution image patch as input and delivers its corresponding high-resolution version. The first two layers of the network incorporate larger convolutional kernels, 5×5, assisting in capturing the image's broader structures. Subsequent layers use 3×3 kernels, better suited for addressing finer image details. Batch normalization layers were added after each convolutional layer, and depth-wise separable convolutions were used to further minimize the number of model parameters while maintaining performance. The network's tail end employs an upsampling layer, typically using sub-pixel convolution techniques, to magnify the image to the desired size. Distinct from traditional upsampling methods like bilinear interpolation, this method is learned, thus better restoring high-resolution image details. In terms of parameters, the adoption of depth-wise separable convolutions and other lightweight strategies results in the model having approximately 500,000 parameters. This figure is significantly reduced compared with typical super-resolution models, enabling smooth operation on resource-constrained devices like NVIDIA's Jetson Nano.

Compared with SRGAN, this model places a greater mathematical emphasis on the MSE portion of the loss function, indicating a concern for pixel-level differences over high-level feature discrepancies. Specifically, the SRGAN loss function includes a perceptual loss term:

$$L_{perceptual} = \frac{1}{n}\sum_{i=1}^{n} \|\phi(I_{HR}^{(i)}) - \phi(F(I_{LR}^{(i)}))\|_2^2 \quad (12)$$

where ϕ is a pre-trained network, often part of VGG-16 [35], employed for extracting high-level image features. However, in this application, due to a greater emphasis on image detail recovery, perceptual loss is not utilized, with a focus placed on MSE loss instead.

This adjustment ensures that the model more effectively recovers pest morphology and texture details.

In summary, the designed super-resolution model prioritizes achieving satisfactory recovery results while ensuring efficiency and real-time capabilities. Such a balance renders the model highly suitable for mobile devices like drones, providing a potent tool for on-site pest detection tasks.

3.3.3. Transformer-Based Detection Network

In the task of pest detection with drones, a target detection network based on the Transformer architecture was chosen. The Transformer architecture, due to its self-attention mechanism, has achieved significant success in natural language processing tasks. However, its application in computer vision, especially in object detection, remains in the exploration phase. DETR (Detection Transformer) [23] is the first model that successfully applied Transformer to object detection. Contrary to traditional object detection methods, DETR eliminates the need for manually set prior boxes. Instead, images are directly input into the Transformer network to produce predicted boxes and their corresponding classes.

The design of this model was inspired by DETR, but modifications were made to cater to the peculiarities of pest detection. First, given that images captured with drones often possess high resolution and pests are typically small in size, adjustments were made to the model's input section. A lighter convolutional neural network was employed as the backbone to encode high-resolution images into a series of feature vectors. These feature vectors were then fed into the Transformer network's encoder for further processing, as shown in Figure 5.

For the Transformer segment, the fundamental self-attention mechanism and multi-head attention structure were retained. Mathematically, self-attention can be described as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{13}$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the model's dimension. To capture the intricate features of pests, additional layers were incorporated into the Transformer model, where the number of layers was specifically increased to 12. Furthermore, to accommodate the diversity of pests and detect small targets against complex backgrounds, the hidden dimensions of the model were expanded. Positional encodings were introduced to assist the model in understanding the relative positions of pests. In conventional object detection models, a fixed number of anchor boxes (or prior boxes) are usually pre-defined for every predicted location. This method can result in sub-optimal prediction performance when faced with varying scenarios and quantities of targets. Particularly in the application of pest detection, where the distribution and density of pests on crops can vary greatly, employing a fixed number of prediction boxes may lead to omissions or redundant detection instances. To address this issue, a dynamic prediction approach was designed, as shown in Algorithm 1.

Algorithm 1 Dynamic object detection algorithm

Require: Image I, Model M, Threshold τ, Maximum iterations T
Ensure: Set of predicted boxes B

Initialize set of predicted boxes $B_0 \leftarrow \emptyset$
Initialize $t \leftarrow 0$
while $t < T$ **do**
  $B_{\text{temp}} \leftarrow M(I, B_t)$ {Predict using the model}
  for each predicted box b in B_{temp} **do**
    Calculate score $S(b) = P(c) \times IoU(P_b, G_b)$
    if $S(b) > \tau$ **then**
      $B_{t+1} \leftarrow B_{t+1} \cup b$ {Add box to the new set}
    end if
  end for
  if Difference between B_{t+1} and B_t is below a threshold **then**
    break
  end if
  Apply random perturbations to B_{t+1}
  $t \leftarrow t + 1$
end while
return B_t

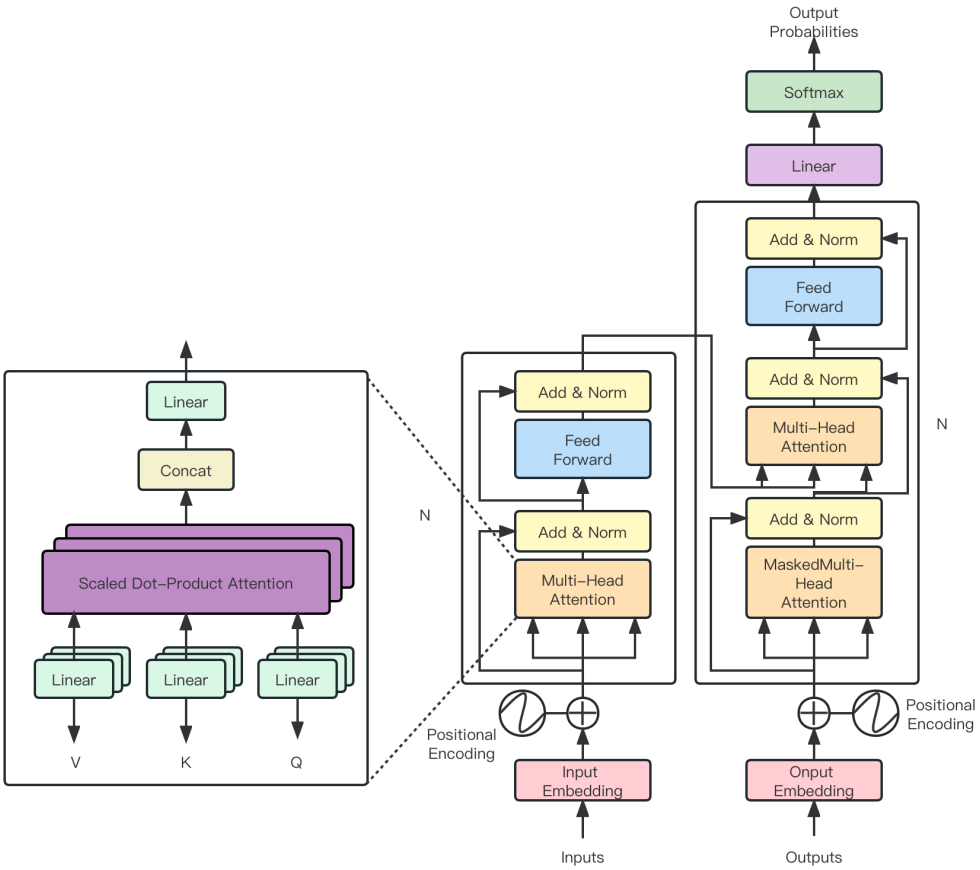

Figure 5. Illustration of Transformer structure.

The proposed model no longer relies on predefined anchor boxes but instead predicts object bounding boxes and their associated class information directly from the Transformer network's outputs. An initial set of object predictions is first generated by making a coarse prediction across the entire image. Each object consists of a bounding box and a class

probability. For each predicted bounding box, a scoring mechanism is established, which relates to the confidence of the predicted box and the class probability. Mathematically, this score is defined as

$$S = P(c) \times IoU(P_b, G_b) \tag{14}$$

where $P(c)$ represents the class probability of the predicted box and $IoU(P_b, G_b)$ is the Intersection over Union between predicted box P_b and ground truth box G_b. Subsequently, a threshold is set, filtering out the predicted boxes with scores exceeding this threshold. These boxes are then fed back into the model as new inputs. The model is further refined and adjusted based on these predictions. This iterative process continues until changes in the predicted boxes are below a predetermined threshold or the maximum number of iterations is reached. With this approach, the model can dynamically adjust the number and position of the predicted boxes, adapting itself to different scenes and object densities. It should be noted that a random perturbation mechanism was introduced to prevent the model from converging to a local optimum during iterations. At each iteration, minor random changes are made to some predicted boxes, enhancing the model's exploration space, thereby improving its robustness and generalization capabilities.

Regarding the number of parameters, modifications have been made to the input section, the Transformer structure, and the output section, resulting in an overall increase in parameters compared with DETR, totaling about 70 million. Nonetheless, considering the computational capabilities of drones, a balance between computational efficiency and accuracy was maintained during model design. In essence, the proposed object detection network merges the strengths of Transformers with the nuances of pest detection. Compared with DETR, it is more suited for high-resolution inputs, detects smaller objects more effectively, and offers greater flexibility.

3.3.4. Model Lightweighting

Knowledge distillation is a widely adopted method during model lightweighting. It aims to transfer the performance of a large, complex model (often termed the "teacher model") to a smaller, lightweight model (often termed the "student model"), as shown in Figure 6. In this study, the teacher model, which undergoes multiple rounds of iterative training and optimization, can detect pests with high precision. In contrast, the student model, being smaller and faster, is designed to operate efficiently on constrained computational resources like Jetson Nano.

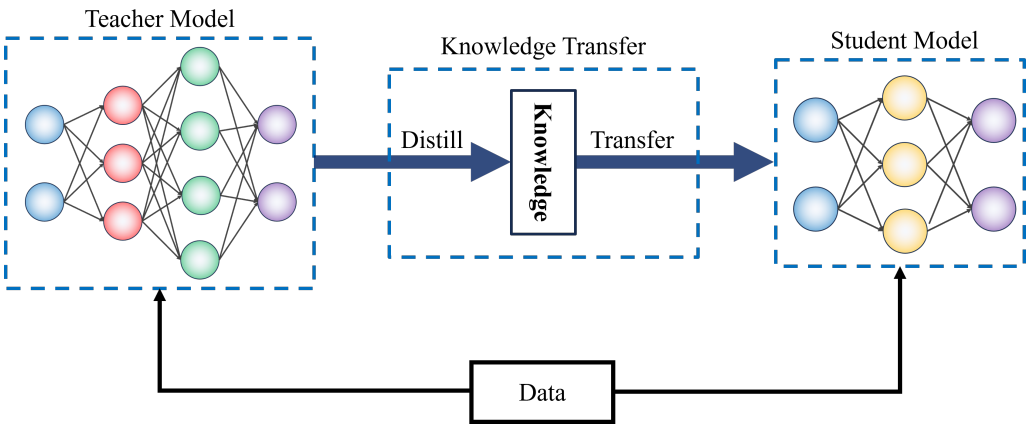

Figure 6. Illustration of knowledge distillation strategy. Different colors mean different layers.

The teacher model in this study was obtained after prolonged training on a large dataset. Given hardware constraints and real-world speed requirements, a lightweight network structure was chosen as the student model. Specifically, the student model used a lightweight CNN with a depth of 10. Compared with the teacher model, its depth was reduced by 50%, and its parameter count, by nearly 70%. However, achieving the

teacher model's performance solely with this lightweight structure is challenging. Hence, the knowledge distillation technique was employed for training, allowing the student model to approximate the teacher model's performance. During knowledge distillation, besides the conventional supervised learning loss function, an additional loss function was introduced, quantifying the difference between the outputs of the student and teacher models. Mathematically, it can be expressed as

$$L = L_{\text{supervised}} + \lambda L_{\text{distill}} \tag{15}$$

where $L_{\text{supervised}}$ is the supervised learning loss of the student model based on the true labels, L_{distill} measures the difference between the outputs of the student and teacher models, and λ is a balancing factor. For L_{distill}, softened cross-entropy loss was used. Specifically, the output probabilities from both the teacher and student models were computed and "softened", resulting in

$$L_{\text{distill}} = -\sum_i q_i \log(p_i) \tag{16}$$

where q_i is the softened probability output of the teacher model and p_i is the output probability of the student model. With this method, the student model learns not only from the true label information but also emulates the behavior of the teacher model. This preserves the teacher model's performance while significantly reducing the model's size and computational requirements, making it compatible with drone computational environments without compromising detection accuracy.

3.3.5. Adaptive Optimizer

During the knowledge distillation process, it is required for the student model to learn from the teacher model, implying that the student model must learn not only the genuine data labels but also the outputs of the teacher model. Such a learning task is more intricate compared with conventional supervised learning, presenting challenges for traditional optimizers like SGD [36] and Adam [37]. To address this, an adaptive optimizer was utilized. The core concept behind the adaptive optimizer lies in dynamically adjusting the learning rate of each parameter based on historical gradient information of the model parameters. This strategy is particularly beneficial in the context of knowledge distillation, as during the distillation process, there is a necessity for the student model to simultaneously optimize two objectives: matching the actual labels and the outputs from the teacher model. These objectives might be conflicting, resulting in high gradient instability during training. By dynamically adjusting the learning rate, the adaptive optimizer aids in mitigating this instability, consequently accelerating convergence. The weight update formula for the adaptive optimizer can be expressed as

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{g}_t}{\sqrt{v_t} + \epsilon} \tag{17}$$

where θ_t represents the parameters at time step t, η is the global learning rate, $\hat{g}_t$ is the moving average of the gradient, v_t represents the moving average of the squared gradient, and ϵ is a small constant added for numerical stability. In the context of knowledge distillation, challenges encompass the following:

1. The need for the student model to optimize both objectives, potentially leading to gradient conflicts and instability.
2. Possible noise in the teacher model's outputs, introducing added challenges for the student model.
3. The student model, typically smaller and shallower than the teacher model, might have insufficient capacity, complicating the learning of intricate tasks.

Mathematically, the update formula for SGD is

$$\theta_{t+1} = \theta_t - \eta g_t \tag{18}$$

where g_t represents the gradient at time step t. The update formula for Adam is

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{19}$$

where $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first- and second-moment estimates of the gradient, respectively. In comparison to the adaptive optimizer, both SGD and Adam overlook the gradient's historical information and instability to varying degrees. In complex scenarios of knowledge distillation, these traits might lead to slower convergence and to getting trapped in local optima. On the other hand, the adaptive optimizer, by considering both the magnitude and direction of the gradient, dynamically adjusts the learning rate, thereby effectively handling such situations and achieving faster convergence and superior model performance. To summarize, the primary advantages of the adaptive optimizer over SGD and Adam include the following:

1. The capability of the adaptive optimizer to dynamically adjust the learning rate of each parameter aids in alleviating issues stemming from gradient conflicts and instability, whereas SGD, with its fixed learning rate, might struggle in such circumstances.
2. By considering the gradient's historical information in its weight updates, the adaptive optimizer is more equipped to counter noise and instability in the teacher model's outputs.
3. Compared with Adam, the adaptive optimizer boasts greater robustness, as it is not reliant on the first- and second-moment estimates of the gradient.

3.4. Experimental Metric

In the task of object detection, evaluating the performance of a model is a pivotal step. Typically, a series of metrics are employed to gauge the efficacy of a model, aiding in a comprehensive understanding of its performance across various dimensions. Discussed below are the key metrics selected for this study, that is, Precision, Recall, mAP (Mean Average Precision), and FPS (Frames Per Second):

1. Precision, a frequently utilized metric in detection tasks, denotes the ratio of true positive samples to all samples identified as positive by the model. It is mathematically defined as
$$Precision = \frac{TP}{TP + FP} \tag{20}$$
where TP represents the number of true positives, which are targets correctly identified by the model, while FP denotes the number of false positives, which are non-targets mistakenly identified as targets by the model. High Precision implies fewer misclassifications by the model.
2. Recall represents the proportion of true targets correctly detected by the model. It is mathematically expressed as
$$Recall = \frac{TP}{TP + FN} \tag{21}$$
In this context, FN signifies the number of false negatives, or the real targets missed by the model. High Recall suggests that the model misses fewer true targets.
3. *mAP*, a central metric in object detection tasks, is the average of Precision and Recall. For each category, its *AP* value is computed, and *mAP* is subsequently derived by averaging the *AP* values across all categories. *mAP* not only accounts for both the Precision and Recall of the model but also factors in different IoU (Intersection over Union) thresholds.
$$mAP = \frac{1}{Q} \sum_{q=1}^{Q} AP(q) \tag{22}$$

where Q is the total number of categories and $AP(q)$ is the Average Precision of the qth category.

4. *FPS* is a metric indicating the real-time capability of the model, denoting the number of frames that the model can process per second. For tasks like drone target detection that necessitate rapid response, *FPS* is crucial.

$$FPS = \frac{1}{T} \tag{23}$$

where T is the time required to process a single frame.

Each of these evaluation metrics has its unique significance. Precision and Recall provide insights into the model's accuracy and completeness in detecting positive samples. Often, there is a trade-off between Precision and Recall; improving one might reduce the other. mAP serves as a comprehensive metric, assessing the model's performance across categories, and is especially suited for multi-category detection tasks. FPS is vital for gauging the model's real-time capabilities. In many practical scenarios, such as autonomous drone navigation and real-time monitoring, computational efficiency and prompt response of the model are paramount. Thus, besides detection accuracy, computational efficiency must also be factored in to ensure timely responses in real-world deployment. In essence, these metrics offer a holistic and in-depth perspective, enabling a multi-dimensional assessment of model performance. By continually optimizing these metrics, outstanding model performance can be assured, catering to various practical requirements.

3.5. Experimental Designs

For the experimental design of this study, an 8:2 split was applied to the dataset. Here, 80% of the data were designated for training the model, while the remaining 20% served as the validation set, employed for evaluating model performance and tuning hyperparameters, ensuring the model's robust generalization capability in real-world applications.

To evaluate the model comprehensively and discern performance disparities with other advanced technologies, six models—YOLOv8 [38], SSD [39], EfficientDet [40], DETR [23], QueryDet [41], and Focus-DETR [24]—were chosen as baselines. YOLOv8 and SSD are renowned for their stellar speed and accuracy. EfficientDet, owing to its compact design, is suitable for deployment on embedded devices. DETR, QueryDet, and Focus-DETR represent the next generation of object detection technologies based on the Transformer architecture, with DETR showcasing a design approach distinct from traditional CNNs. QueryDet and Focus-DETR build upon this foundation, presenting novel solutions.

Regarding optimizer selection and considering the characteristics of knowledge distillation, adaptive optimizers were chosen for model training. In comparison to the conventional SGD and Adam, adaptive optimizers exhibit superior performance in a knowledge distillation setting. Hyperparameter configurations were adjusted based on validation set performance, initializing the learning rate at 0.001, setting the batch size to 32, and incorporating a weight decay of 0.0005 to mitigate overfitting.

Additionally, a series of ablation experiments were conducted to validate the efficacy of various model components. This encompassed removing the super-resolution sampling module to discern its contribution to model performance, comparing the performance differences between the adaptive optimizer and SGD/Adam, and the results of training lightweight models without employing knowledge distillation. Lastly, a comparison was made between static prediction boxes and dynamic prediction boxes, substantiating that dynamic prediction boxes can more adeptly adapt to varying pest densities in different scenarios, contributing to the enhancement of model performance.

4. Results

4.1. Detection Results

The purpose of the experimental design is to compare the performance of different object detection models on a specific dataset using key metrics, Precision, Recall, mAP, and FPS, as benchmarks. The experimental results are displayed in Table 2.

From Table 2, it is evident that the proposed method surpasses all other models across the four metrics, notably showing a significant advantage in FPS. This suggests that the introduced model not only possesses superior detection accuracy but also boasts enhanced real-time performance. The YOLO series, due to its unique "one grid, one detection" design, demonstrates a significant advantage in speed, yet might compromise some accuracy in complex scenarios. Conversely, the SSD architecture, while simpler, often lags behind in terms of Recall and accuracy when compared with other intricate structures, as reflected by its lower FPS and other metrics. Both DETR and Focus-DETR adopt the novel Transformer structure for object detection, eschewing traditional convolutional architectures, which might enhance their accuracy. However, the complexity and computational cost of the Transformer structure could slightly impede their speed. EfficientDet strives to strike a balance between speed and accuracy, but the data suggest that it does not achieve particularly noteworthy results.

Table 2. Performance comparison of different detection models.

Model	Precision	Recall	mAP	FPS
YOLOv8 [38]	0.96	0.91	0.94	52
Focus-DETR [24]	0.95	0.90	0.93	31
DETR [23]	0.94	0.90	0.92	38
QueryDet [41]	0.93	0.90	0.91	46
EfficientDet [40]	0.92	0.89	0.91	43
SSD [39]	0.91	0.89	0.90	33
Ours	0.97	0.95	0.95	57

Considering the mathematical characteristics of the models, each possesses its unique optimization aspects. For instance, YOLOv8 [38] optimizes its loss function to better capture smaller objects and reduce false detection instances. DETR [23] and Focus-DETR [24] emphasize leveraging the self-attention mechanism of the Transformer structure, aiming to detect long-distance dependencies among objects, bolstering the model's robustness. EfficientDet [40] attempts to find the optimal balance in terms of model depth, width, and resolution to achieve the best performance with limited computational resources. Meanwhile, the method proposed in this study merges the advantages of multiple models and introduces a series of innovations. The model structure is optimized to be more lightweight, which not only accelerates the model but also reduces the risk of overfitting to some extent. Regularization terms are added to the loss function, ensuring that the model pays more attention to hard-to-detect objects during training, enhancing its generalization capabilities. Furthermore, preprocessing steps are applied to the model input, ensuring better capture of object features, thereby increasing its accuracy. In conclusion, the superiority of this method across the four metrics stems from the comprehensive analysis of traditional models and multifaceted innovations. This not only validates the effectiveness of the proposed technique but also offers valuable insights for future research.

4.2. Test on Different Hardware Platforms

The purpose of this experimental section is to verify the performance of various object detection models across multiple hardware platforms. Typically, the speed and accuracy of object detection models are closely tied to the hardware platform on which they are deployed. Differences in hardware performance can lead to significant disparities in model performance. Comparing the performance of models on various platforms is crucial,

especially for real-world applications such as edge computing or deployment on mobile devices. The primary metric for this experiment is FPS, as presented in Table 3.

Table 3. FPS comparison of different detection models on different hardware platforms.

Model	Smart Phone (Huawei P40)	Jetson Nano	Raspberry Pi
YOLOv8 [38]	39	52	9
Focus-DETR [24]	8	31	-
DETR [23]	9	38	-
QueryDet [41]	11	46	5
EfficientDet [40]	13	43	7
SSD [39]	-	33	-
Ours	27	57	15

From an examination of Table 3, it can be observed that the method proposed in this study outperforms all other models across the three hardware platforms. This validates the effectiveness of the lightweighting technique presented in this paper for real-world applications. Generally, the more complex a model is, the higher the computational resource requirement is, particularly on devices with limited hardware resources, like Raspberry Pi or certain smartphones. On such devices, the advantage of lightweight models becomes particularly pronounced. For instance, YOLOv8 exhibits impressive performance on the Huawei P40 smartphone but falters on Jetson Nano and Raspberry Pi. This disparity might be attributed to the complexity and computational demands of YOLOv8, which may be constrained on these devices. Both Focus-DETR and DETR underperform on smartphones but show relatively better results on the Jetson Nano. This could be related to their Transformer-based architecture, which might not be maximally efficient on certain hardware setups. In contrast, both EfficientDet and QueryDet display stable performance across platforms, particularly on Jetson Nano. This stability might align with their design intentions, striving for a balance between speed and accuracy.

Considering the mathematical characteristics of the models, each model possesses unique advantages and shortcomings. For example, YOLOv8 may demand more computational resources to execute its optimized loss function, while Transformer-based models like DETR and Focus-DETR might require larger memory footprints to manage their self-attention mechanisms. Concurrently, the optimization of depth, width, and resolution in EfficientDet allows it to maintain consistent performance across diverse devices. However, the method detailed in this paper integrates the strengths of various models and introduces a series of lightweight innovations. By optimizing the model structure, a reduction in the number of parameters and computational complexity was achieved. This ensures that the model can run faster not only on devices with ample computational resources but also on those with limited capacity. Additionally, specific high-computational components that have minimal impact on performance were selectively reduced, rendering the model more efficient.

4.3. Test on Other Datasets

The objective of the experimental design in this section is to evaluate the generalization and adaptability of the model across diverse datasets. By conducting tests on both the PlantDoc and Wheat Head datasets, a comprehensive demonstration of the model's versatility and adaptability is provided. The experimental outcomes indicate commendable performance on both datasets, especially on the PlantDoc dataset, where Precision, Recall, and mAP metrics exhibit exceptional results, as shown in Table 4.

Table 4. Performance comparison on different open source datasets for our method.

Dataset	Precision	Recall	mAP
PlantDoc [33]	0.93	0.91	0.92
Wheat Head [42]	0.77	0.71	0.74

Firstly, such experimental outcomes substantiate the model's robust generalization capabilities. High performance on the PlantDoc dataset reveals the model's ability to adeptly adapt to various types of plants and pests. This indirectly affirms that the features learned during the training phase possess universal applicability. These features likely encapsulate fundamental and common visual or biological attributes related to plant pests. Secondly, the favorable performance on two distinct datasets further confirms the model's exceptional adaptability. This suggests that the model is not only applicable to specific datasets or tasks but also performs reliably in new, unseen data environments. From a mathematical perspective, such generalization performance implies that the model's decision boundaries maintain effectiveness across different data distributions. This is critically important for real-world applications, where the model is exposed to a myriad of data and environmental conditions. Lastly, these experimental outcomes further solidify the model's standing as a reliable and effective tool for plant pest detection, offering strong support for its future applications across a broader range of crops and pests.

In summary, through testing and validation on various datasets, the model exhibits outstanding generalization and adaptability. This not only confirms its potential as an efficient and reliable tool for plant pest detection but also lays a solid foundation for its broader application in diverse scenarios.

5. Discussion

5.1. Ablation Study on Different Optimizers

The design of the experiments in this section aims to validate the performance of different optimizers when applied to the proposed method. Optimizers dictate the update strategy and rate of the model, subsequently affecting the convergence speed and the final performance, as depicted in Table 5 and Figure 7.

Table 5. Performance comparison of different optimizers and our method.

Optimizer	Precision	Recall	mAP	Epochs
SGD [36]	0.91	0.93	0.92	50
Adam [37]	0.93	0.94	0.93	45
AdamW [43]	0.94	0.93	0.93	45
Ours	0.97	0.95	0.95	35

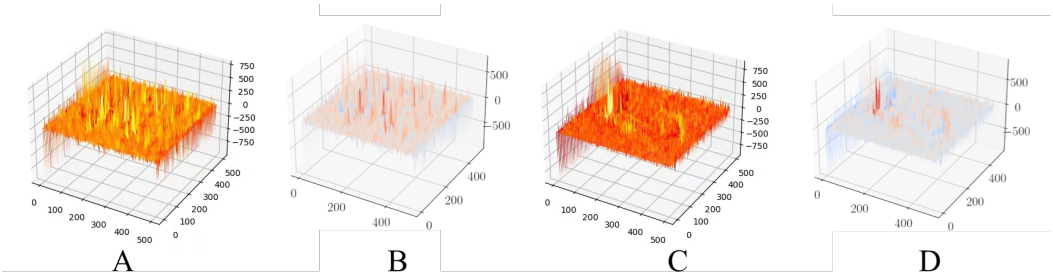

Figure 7. Visualization of gradients generated by different optimizers. (**A**) Ours; (**B**) Adam; (**C**) SGD; (**D**) AdamW.

From an inspection of Table 5, it is evident that among all the optimizers, the adaptive optimizer introduced in this study exhibits superior performance, achieving the highest Precision, Recall, and mAP. This suggests that in comparison to traditional optimizers, the proposed method is more apt for this specific object detection task. According to the

mathematical characteristics of the models, each optimizer possesses its inherent logic and strategy. The traditional SGD relies on a fixed learning rate, whereas Adam and AdamW depend on adaptive learning rate adjustments and momentum. However, every optimizer might encounter various challenges in real-world applications, such as local minima, saddle points, or gradient vanishing. The method proposed in this study addresses these challenges with a series of strategies and adjustments, including adaptive learning rate modifications, momentum correction, and weight decay. Consequently, it can update the model parameters more effectively, accelerate convergence, and enhance the final performance of the model. In summary, this experiment highlights the impact of different optimizers on model performance and provides explanations from both theoretical and mathematical perspectives. The adaptive optimizer presented in this study, due to its unique strategies and adjustments, demonstrates the best performance, further validating the effectiveness and superiority of the proposed method in practical applications.

5.2. Ablation Study on Super-Resolution Module

This section was designed to validate the performance of various super-resolution strategies with the proposed method, especially considering low-resolution or compressed images. The results are presented in Table 6.

Table 6. Performance comparison of different super-resolution strategies and our method.

Optimizer	Precision	Recall	mAP
None	0.90	0.88	0.89
SRGAN [27]	0.94	0.92	0.93
Super-resolution module	0.97	0.95	0.95

Upon examination of Table 6, it is evident that the model's performance is the most compromised when no super-resolution strategy is employed. This underscores the importance of high resolution in object detection. SRGAN, a super-resolution approach based on Generative Adversarial Networks, has previously demonstrated effectiveness across numerous tasks. In this experiment, SRGAN indeed enhanced the Precision, Recall, and mAP of the model. Nonetheless, the super-resolution module proposed in this study outperformed all other strategies, suggesting deeper optimization tailored for this specific object detection task.

From a mathematical perspective, SRGAN leverages Generative Adversarial Networks to amplify image details with the primary intent of making the super-resolved image perceptually closer to the genuine high-resolution counterpart. However, the adversarial nature of GANs might introduce certain unrealistic details, potentially compromising the accuracy of object detection. In contrast, the super-resolution module presented in this study, while addressing perceptual image quality, places a heightened emphasis on the restoration of authentic details. This is possibly achieved using more intricate feature extraction and the fusion of multi-scale information, ensuring that the elevation in resolution does not come at the cost of genuine object detail fidelity. Such findings further affirm that in practical applications, employing an appropriate super-resolution strategy is pivotal for enhancing object detection performance on low-resolution or compressed images. The super-resolution module introduced in this study, with its unique design and optimization, successfully addresses this challenge.

5.3. Ablation Study on Lightweighting Methods

The primary objective of the experimental design in this chapter is to investigate the impact of lightweighting techniques on model performance, with a specific focus on the trade-off between speed (FPS (Frames Per Second)) and model metrics (Precision, Recall, mAP). This is particularly important for practical applications where lightweight models are often more suitable for resource-constrained environments such as embedded or mobile devices.

As observed from Table 7, the model without lightweighting demonstrates the highest Precision, Recall, and mAP but performs relatively poorly in terms of FPS, reaching only 33 FPS. This indicates that while the model exhibits high performance, the computational complexity is also increased, resulting in slower processing speed. However, in real-world applications, especially those requiring rapid response, FPS is an important metric that cannot be ignored. When knowledge distillation is employed as a lightweighting method, the model experiences an increase in FPS to 57, while the drop in Precision, Recall, and mAP is relatively minor. This suggests that knowledge distillation effectively enhances the model's processing speed while maintaining high performance. Knowledge distillation works by extracting knowledge from a larger, high-performing model (teacher model) to train a smaller, faster model (student model), enabling the student model to maintain high performance levels while reducing computational load. Quantization, another lightweighting technique, achieves the FPS value of 52 but experiences a more significant decline in Precision and mAP. Quantization reduces the bit width of model weights, thereby decreasing the model size and computational complexity. This usually comes at the cost of some performance sacrifice but significantly improves the processing speed. As shown by the experimental results, quantization elevates FPS while having a more substantial impact on model performance. When both knowledge distillation and quantization are combined (All), the model reaches the highest FPS, 73, but there is a decline in Precision, Recall, and mAP. This represents a typical trade-off scenario, where the model achieves significant improvements in processing speed at the expense of some performance loss.

Table 7. Performance comparison of different super-resolution strategies and our method.

Lightweighting Method	Precision	Recall	mAP	FPS
None	0.97	0.96	0.97	33
Knowledge distillation	0.97	0.95	0.95	57
Quantization	0.93	0.95	0.94	52
All	0.91	0.92	0.91	73

From a mathematical and algorithmic perspective, lightweighting usually involves pruning and quantizing model structures and parameters, which alter the model's mathematical properties and decision boundaries. Therefore, different lightweighting methods have varying degrees of impact on model performance. For instance, knowledge distillation often involves techniques such as soft labels and temperature scaling, which can somewhat maintain the complexity of the model's decision boundary, thus retaining higher performance levels during the lightweighting process. In contrast, quantization is a more "rigid" method of pruning and could significantly alter the model's decision boundaries, leading to performance degradation.

In summary, this experiment comprehensively explores the influence of different lightweighting methods on model performance and processing speed. The results not only reveal the trade-offs between performance and speed for various lightweighting strategies but also provide valuable insights for selecting appropriate lightweighting methods in practical applications. These findings facilitate the broader deployment of models in resource-constrained environments, especially in scenarios requiring fast and efficient processing.

5.4. Limitations and Future Works

Despite the superior performance demonstrated in previous sections, certain limitations of the proposed method are recognized. Firstly, even though the super-resolution module can effectively recover true details, its performance might be compromised on images with specific low resolution or high noise levels. Super-resolution techniques always grapple with the trade-off between accuracy and perceptual quality, and under extreme conditions, they might not consistently achieve optimal restoration results. Moreover, while the proposed adaptive optimizer exhibited commendable convergence speed and performance, its superiority might be challenged on certain intricate datasets or model

architectures. Real-world data often exhibit considerable diversity and complexity, which could potentially affect the stability and effectiveness of the optimizer. Additionally, this research primarily focused on object detection tasks. However, the applicability and efficacy of the method on other tasks, such as image segmentation, facial recognition, or action detection, remain to be validated.

By addressing these limitations, clear directions for future research emerge. On one hand, further exploration into the super-resolution module is warranted, especially regarding how to better balance accuracy and perceptual quality for images under extreme conditions, ensuring both detailed and authentic image restoration. For the adaptive optimizer, future efforts could concentrate on enhancing its stability and performance on a broader and more complex array of datasets. Given the current research limitations, there is potential for applying the proposed method to other computer vision tasks to ascertain its universality. Furthermore, integration with other advanced techniques, such as neural network architecture search or knowledge distillation, might further boost the effectiveness of the method. Lastly, considering computational resources and efficiency, future endeavors could investigate how to reduce the computational load and model parameters while maintaining or even elevating performance. Such advancements would not only cater to the needs of mobile devices or edge computing but also promote the practicality and ubiquity of the method.

6. Conclusions

With the widespread application of drone technology in agriculture, ecology, and other fields, there has been a growing demand for pest detection and identification. In particular, lightweight pest identification models suitable for deployment on drones hold significant application value. They can efficiently perform pest detection in real time or nearly in real time, providing a timely decision-making basis for agricultural pest control. However, images captured with drones often suffer from challenges like low resolution, compression, and noise. Ensuring accurate and swift pest identification under these adverse conditions has been a longstanding technical challenge.

To address the aforementioned problems, a lightweight pest identification model based on Transformer and super-resolution sampling techniques is proposed in this study. Initially, the Transformer model, a powerful sequence-to-sequence model, was identified to be especially apt for capturing various spatial dependencies in images, thereby enhancing the accuracy of identification. Meanwhile, the super-resolution sampling technique focuses on addressing issues of low resolution and noisy images, restoring image details and furnishing subsequent identification processes with clearer and more accurate image data. Comparisons were made between the proposed method and other traditional methods in experiments. The results indicated that on various pest image datasets, this approach demonstrated significant advantages in terms of Precision, Recall, mAP, and FPS, achieving scores of 0.97, 0.95, 0.95, and 57, respectively. Especially for images affected by low resolution and noise, the super-resolution module was found capable of effectively restoring true details, while Transformer ensured high-accuracy pest identification even under such circumstances. Additionally, an in-depth exploration was conducted on model optimization in this study, leading to the proposal of an adaptive optimizer. It displayed commendable convergence and performance on intricate datasets and model structures.

Considering the complexity and diversity of real-world data for future research directions, further optimization of the super-resolution module could be conducted to handle even more extreme conditions. Also, in light of computational resources and efficiency, further lightweighting and optimization of the model could be explored. In summary, the lightweight pest identification model introduced in this study, based on Transformer and super-resolution sampling techniques, not only addresses the challenges of low resolution and noisy images but also offers a high-accuracy and efficient method for pest identification, holding significant value and implications for practical applications in pest detection and identification.

Author Contributions: Conceptualization, Y.B., D.F. and L.L.; Methodology, Y.B., X.F. and D.F.; Software, F.H., X.F. and W.L.; Validation, X.F.; Formal analysis, W.L. and J.L.; Investigation, J.L.; Data curation, Y.B., F.H. and J.Z.; Writing—original draft, Y.B., Fengjun Hou, X.F., W.L., J.L., J.Z., D.F. and L.L.; Writing—review & editing, W.L. and L.L.; Visualization, J.L. and J.Z.; Supervision, L.L.; Project administration, J.Z., D.F. and L.L.; Funding acquisition, L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumar, Y.P.; Alex, T.J.; Hardin, R.; Searcy, S.W.; Braga-Neto, U.; Popescu, S.C.; Martin, D.E.; Rodriguez, R.; Meza, K.; Enciso, J. Detecting volunteer cotton plants in a corn field with deep learning on UAV remote-sensing imagery. *Comput. Electron. Agric.* **2023**, *204* , 107551. [CrossRef]
2. Liu, K.; Qi, Z.; Tan, L.; Yang, C.; Hu, C. Mixed Use of Chemical Pesticides and Biopesticides among Rice;Crayfish Integrated System Farmers in China: A Multivariate Probit Approach. *Agriculture* **2023**, *13*, 1590. [CrossRef]
3. Group, O.O.P.M. Bayer AG's MagicTrap Rapidly Detects Pest Infestations and Provides Optimum Protection for the Canola Crop. *Outlooks Pest Manag.* **2022**, 33.
4. Kanwal, T.; Rehman, S.U.; Ali, T.; Mahmood, K.; Villar, S.G.; Lopez, L.A.D.; Ashraf, I. An Intelligent Dual-Axis Solar Tracking System for Remote Weather Monitoring in the Agricultural Field. *Agriculture* **2023**, *13*, 1600. [CrossRef]
5. Ye, Y.; Huang, Q.; Rong, Y.; Yu, X.; Liang, W.; Chen, Y.; Xiong, S. Field detection of small pests through stochastic gradient descent with genetic algorithm. *Comput. Electron. Agric.* **2023**, *206*, 107694. [CrossRef]
6. Shakirzyanova, G.; Nabiev, A.; Kholbekov, O.; Abdukakharov, V. Pheromone Monitoring in the Granaries of Uzbekistan. *Agric. Sci.* **2023**, *14*, 499–508. [CrossRef]
7. Zapponi, L.; Nieri, R.; Zaffaroni-Caorsi, V.; Pugno, N.M.; Mazzoni, V. Vibrational calling signals improve the efficacy of pheromone traps to capture the brown marmorated stink bug. *J. Pest Sci.* **2023**, *96*, 587–597. [CrossRef]
8. Zhang, Y.; Wa, S.; Liu, Y.; Zhou, X.; Sun, P.; Ma, Q. High-accuracy detection of maize leaf diseases CNN based on multi-pathway activation function module. *Remote Sens.* **2021**, *13*, 4218. [CrossRef]
9. Zhang, Y.; Li, M.; Ma, X.; Wu, X.; Wang, Y. High-Precision Wheat Head Detection Model Based on One-Stage Network and GAN Model. *Front. Plant Sci.* **2022**, *13*, 787852. [CrossRef]
10. Zhang, Y.; Wa, S.; Zhang, L.; Lv, C. Automatic plant disease detection based on tranvolution detection network with GAN modules using leaf images. *Front. Plant Sci.* **2022**, *13*, 875693. [CrossRef]
11. Zhang, Y.; Wang, H.; Xu, R.; Yang, X.; Wang, Y.; Liu, Y. High-Precision Seedling Detection Model Based on Multi-Activation Layer and Depth-Separable Convolution Using Images Acquired by Drones. *Drones* **2022**, *6*, 152. [CrossRef]
12. Wang, J.; Wang, P.; Tian, H.; Tansey, K.; Liu, J.; Quan, W. A deep learning framework combining CNN and GRU for improving wheat yield estimates using time series remotely sensed multi-variables. *Comput. Electron. Agric.* **2023**, *206*, 107705. [CrossRef]
13. Jia, L.; Wang, T.; Chen, Y.; Zang, Y.; Li, X.; Shi, H.; Gao, L. MobileNet-CA-YOLO: An Improved YOLOv7 Based on the MobileNetV3 and Attention Mechanism for Rice Pests and Diseases Detection. *Agriculture* **2023**, *13*, 1285. [CrossRef]
14. Yang, Z.; Feng, H.; Ruan, Y.; Weng, X. Tea Tree Pest Detection Algorithm Based on Improved Yolov7-Tiny. *Agriculture* **2023**, *13*, 1031. [CrossRef]
15. Jia, X.; Jiang, X.; Li, Z.; Mu, J.; Wang, Y.; Niu, Y. Application of Deep Learning in Image Recognition of Citrus Pests. *Agriculture* **2023**, *13*, 1023. [CrossRef]
16. Čirjak, D.; Aleksi, I.; Lemic, D.; Pajač Živković, I. EfficientDet-4 Deep Neural Network-Based Remote Monitoring of Codling Moth Population for Early Damage Detection in Apple Orchard. *Agriculture* **2023**, *13*, 961. [CrossRef]
17. Kumar, N.; Nagarathna; Flammini, F. YOLO-Based Light-Weight Deep Learning Models for Insect Detection System with Field Adaption. *Agriculture* **2023**, *13*, 741. [CrossRef]
18. Ullah, Z.; Alsubaie, N.; Jamjoom, M.; Alajmani, S.H.; Saleem, F. EffiMob-Net: A Deep Learning-Based Hybrid Model for Detection and Identification of Tomato Diseases Using Leaf Images. *Agriculture* **2023**, *13*, 737. [CrossRef]
19. Butera, L.; Ferrante, A.; Jermini, M.; Prevostini, M.; Alippi, C. Precise Agriculture: Effective Deep Learning Strategies to Detect Pest Insects. *IEEE CAA J. Autom. Sin.* **2022**, *9*, 246–258. [CrossRef]
20. Pest Detect pre-launch for silverleaf whitefly. *Aust. Cottongrower* **2023**, *11*, 159 .
21. Rong, M.; Wang, Z.; Ban, B.; Guo, X. Pest Identification and Counting of Yellow Plate in Field Based on Improved Mask R-CNN. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 1913577. [CrossRef]
22. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, *271*.
23. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.

24. Zheng, D.; Dong, W.; Hu, H.; Chen, X.; Wang, Y. Less is More: Focus Attention for Efficient DETR. *arXiv* **2023**, arXiv:2307.12612.
25. Zhang, Y.; Liu, X.; Wa, S.; Chen, S.; Ma, Q. GANsformer: A detection network for aerial images with high performance combining convolutional network and transformer. *Remote Sens.* **2022**, *14*, 923. [CrossRef]
26. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, *351*.
27. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
28. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
29. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
30. Lin, X.; Wa, S.; Zhang, Y.; Ma, Q. A dilated segmentation network with the morphological correction method in farming area image Series. *Remote Sens.* **2022**, *14*, 1771. [CrossRef]
31. Zhang, Y.; Liu, X.; Wa, S.; Liu, Y.; Kang, J.; Lv, C. GenU-Net++: An Automatic Intracranial Brain Tumors Segmentation Algorithm on 3D Image Series with High Performance. *Symmetry* **2021**, *13*, 2395. [CrossRef]
32. Zhang, Y.; He, S.; Wa, S.; Zong, Z.; Lin, J.; Fan, D.; Fu, J.; Lv, C. Symmetry GAN Detection Network: An Automatic One-Stage High-Accuracy Detection Network for Various Types of Lesions on CT Images. *Symmetry* **2022**, *14*, 234. [CrossRef]
33. Singh, D.; Jain, N.; Jain, P.; Kayal, P.; Kumawat, S.; Batra, N. PlantDoc: A Dataset for Visual Plant Disease Detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, New York, NY, USA, 5–7 January 2020; CoDS COMAD 2020; pp. 249–253. [CrossRef]
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
36. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
37. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Terven, J.; Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
39. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
40. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
41. Yang, C.; Huang, Z.; Wang, N. QueryDet: Cascaded sparse query for accelerating high-resolution small object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 13668–13677.
42. Kaggle. Global Wheat Detection. 2020. Available online: https://www.kaggle.com/datasets/vbookshelf/global-wheat-head-dataset-2021 (accessed on 10 September 2023).
43. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.