applied sciences

MDPI

Article

RS Transformer: A Two-Stage Region Proposal Using Swin Transformer for Few-Shot Pest Detection in Automated Agricultural Monitoring Systems

Tengyue Wu [1,2], Liantao Shi [1], Lei Zhang [2,*], Xingkai Wen [3], Jianjun Lu [4] and Zhengguo Li [1,*]

[1] Institute for Carbon-Neutral Technology, Shenzhen Polytechnic University, Shenzhen 518055, China; 202007020208@stu.bucea.edu.cn (T.W.); xiaoshi1108@outlook.com (L.S.)
[2] School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
[3] School of Mathematics and Statistics, Northeast Normal University, Changchun 130024, China; wenxk@nenu.edu.cn
[4] College of Economics and Management, China Agricultural University, Beijing 100083, China; ljjun@cau.edu.cn
* Correspondence: lei.zhang@bucea.edu.cn (L.Z.); lizhengguo@szpt.edu.cn (Z.L.)

Abstract: Agriculture is pivotal in national economies, with pest classification significantly influencing food quality and quantity. In recent years, pest classification methods based on deep learning have made progress. However, there are two problems with these methods. One is that there are few multi-scale pest detection algorithms, and they often lack effective global information integration and discriminative feature representation. The other is the lack of high-quality agricultural pest datasets, leading to insufficient training samples. To overcome these two limitations, we propose two methods called RS Transformer (a two-stage region proposal using Swin Transformer) and the Randomly Generated Stable Diffusion Dataset (RGSDD). Firstly, we found that the diffusion model can generate high-resolution images, so we developed a training strategy called the RGSDD, which was used to generate agricultural pest images and was mixed with real datasets for training. Secondly, RS Transformer uses Swin Transformer as the backbone to enhance the ability to extract global features, while reducing the computational burden of the previous Transformer. Finally, we added a region proposal network and ROI Align to form a two-stage training mode. The experimental results on the datasets show that RS Transformer has a better performance than the other models do. The RGSDD helps to improve the training accuracy of the model. Compared with methods of the same type, RS Transformer achieves up to 4.62% of improvement.

Keywords: Swin Transformer; pest detection; diffusion model; feature extraction; few-shot learning

check for updates

Citation: Wu, T.; Shi, L.; Zhang, L.; Wen, X.; Lu, J.; Li, Z. RS Transformer: A Two-Stage Region Proposal Using Swin Transformer for Few-Shot Pest Detection in Automated Agricultural Monitoring Systems. *Appl. Sci.* **2023**, *13*, 12206. https://doi.org/10.3390/app132212206

Academic Editor: Douglas O'Shaughnessy

Received: 12 October 2023
Revised: 4 November 2023
Accepted: 7 November 2023
Published: 10 November 2023

1. Introduction

Agriculture directly impacts people's lives and is essential to the development of the global economy. However, pests in crops often cause great losses. Therefore, it is necessary to control pests to ensure a high agricultural yield [1]. Because of developments in science and technology, pest detection methods are continually changing [2]. Early detection relies on field diagnosis by agricultural experts, but proper diagnosis is difficult due to the complexity of pest conditions, lack of qualified staff, and inconsistent experience at the grassroots level. Furthermore, incorrect pest identification by farmers has led to an escalation in pesticide usage. This in turn has bolstered pest resistance [3] and has exacerbated the harm inflicted upon the natural environment.

An effective integrated pest automated monitoring system relies on a high-quality algorithm. With the development of image processing technology and deep learning, scholars are increasingly using pest image data and deep learning to identify pests, which improves the effectiveness of agricultural pest detection and is also the first application example of intelligent diagnosis. Research in respect of the classification and detection of agricultural pests is crucial to help farmers effectively manage crops and take timely measures to reduce the harm caused by pests. Object detection models, which come in one-stage and two-stage varieties, are frequently employed in pest classification and detection. One-stage models like YOLO [4–6] and SSD [7] are renowned for their rapid detection capabilities. In contrast, two-stage models like Fast R-CNN [8] and Faster R-CNN [9] excel in achieving high accuracy, albeit at a slower processing speed compared to their one-stage counterparts. The Transformer model [10] has many potential applications in AI. Based on its effectiveness in natural language processing (NLP) [11], recent research has extended the Transformer to the field of computer vision (CV) [12]. In 2021, Swin Transformer [13] was proposed as a universal backbone for CV, achieving the latest SOTA on multiple dense prediction benchmarks. The differences between language and vision make the transition from language to vision difficult, such as the vast range of visual entity scales. However, Swin Transformer can solve this problem well. In this paper, we use a Vision Transformer with a shift window to detect pests.

Currently, two dataset-related issues affect pest detection. The first is the scarcity of high-quality datasets. There are only approximately 600 photos in eight pest datasets, reflecting the lack of agricultural pest datasets [14]. The second issue is the challenges involved in detecting pests at multiple scales. The size difference between large and microscopic pests is large, up to 30 times in some cases. For example, the relative size of the largest pest in the LMPD2020 dataset is 0.9%, while the relative size of the smallest pest is only 0.03%. When the size difference of the test object is large, it is difficult for the test results at multiple scales to achieve high accuracy simultaneously, and the problem of missing detection often occurs. Moreover, the Transformer also requires a large dataset for training.

In agriculture, there are few high-quality pest datasets available, and some datasets from the internet have poor clarity and different sizes. In recent years, with the development of AI-generated content technology, increasing numbers of large models of image generation based on a text description have been developed. The diffusion model [15], introduced as a sequence of denoising autoencoders, aims to remove Gaussian noise through continuous application during training with images. A new diffusion model [16] represents a novel state-of-the-art in-depth image generation. In picture-generating tasks, it outperforms the original SOTA, i.e., GAN (generative adversarial network) [17], and performs well in a variety of applications, including CV, NLP, waveform signal processing, time series modeling, and adversarial learning. The Denoising Diffusion Probabilistic Model was proposed later [18], applying to image generation. Then, Open AI's paper "Diffusion Models Beat GANs on Image Synthesis" [19] made machine-generated data even more realistic than GAN. DALL-E2 [20] allows us to use text descriptions to generate the desired image. To improve the accuracy of pest identification, we can enable models to learn more complex semantic information from training data and complement the agricultural dataset. We propose the Randomly Generated Stable Diffusion Dataset (RGSDD) method to help generate pest images.

We identified four years of representative pest detection papers, as shown in Table 1, and counted the algorithms used in the papers and the pest species included in the datasets. It was found that previous papers did not use Swin Transformer as a backbone network, nor did they use a diffusion model to generate datasets.

Table 1. Statistical pest detection algorithms and accuracy.

Year	Author Reference	Pest	Module	Performance	Generated Dataset
2019	Liu et al. [21]	16 butterfly species	CNN	mAP (75.46%)	×
2020	Jiao et al. [22]	24 agricultural pests	AF-RCNN	mAP (56.4%)	×
2020	Pattnaik et al. [23]	10 pest species	Deep CNN	Accuracy (88.83%)	×
2020	Lee et al. [24]	Leaf miner, tea thrip, tea leaf roller, and tea mosquito bug (TMB)	Faster RCNN	mAP (66.02%)	×
2021	Chen et al. [25]	*T. papillosa*	YOLOv3	mAP (0.93%)	×
2021	Wang et al. [26]	Agricultural pests	RPN	mAP (78.7%)	×
2022	Peng et al. [27]	102 pests	CNN, Transformer	Accuracy (74.90%)	×
2022	ULLAH et al. [28]	9 crop pests	CNN	Accuracy (100%)	×
2023	Our method	8 agricultural pests	RS Transformer	mAP (90.18%)	√

×: Not using the generated dataset; √: Using the generated dataset.

Overall, this paper makes the following contributions:

(1) RS Transformer: a novel model based on the region proposal network (RPN), Swin Transformer, and ROI Align, for few-shot detection of pests at different scales.

(2) RGSDD: a new training strategy method named the Randomly Generate Stable Diffusion Dataset is introduced to expand small pest images to effectively classify and detect pests in a few-shot learning scenario.

(3) Comprehensive experiments on the pest dataset confirm the success of our proposed methods, contrasting with SSD [7], Faster R-CNN [9], YOLOv3 [4], YOLOv4 [5], YOLOv5m [6], YOLOv8, and DETR [29].

2. Materials and Methods

2.1. Pest Dataset

2.1.1. Real Pest Image Dataset

This study focuses on crops of high economic value. As a result, the selection of agricultural pests is based on small sample sizes. First, we went to the Beizang Village experimental field next to the Daxing Campus of Beijing University of Civil Engineering and Architecture to take photos using an iPhone 12 Pro Max and collected 400 pictures of pests. The photos were taken at a resolution of 3024 × 4032 pixels. Secondly, we searched for pests in the IPMImages database [30], National Bureau of Agricultural Insect Resources (NBAIR), Google, Bing, etc. The dataset has eight pest species as labels, which are as follows: Tetranychus urticae, TU; Bemisia argentifolii, BA; Zeugodacus cucurbitae, ZC; Thrips palmi, TP; Myzus persicae, MP; Spodoptera litura, SL; Spodoptera exigua, SE; and Helicoverpa armigera, HA. Figure 1 displays a few representative photos from the dataset. The final pest dataset includes 1009 images.

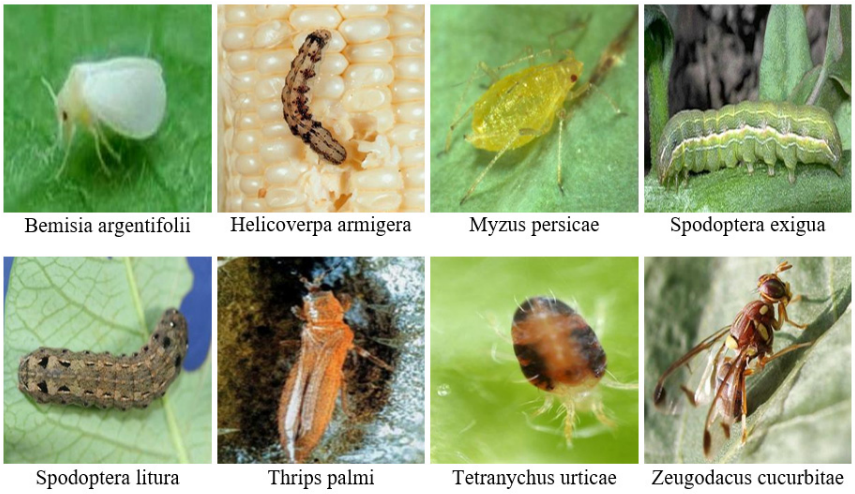


Figure 1. Pest dataset.

2.1.2. Dataset Generation

Stable diffusion was released by Open AI, a model that can be used to generate detailed images conditioned on text descriptions.

The diffusion model, which produces samples that fit the data after a finite amount of time, is a parameterized Markov chain trained via variational inference [18]. As shown in Figure 2, the forward process and the reverse process can be separated from the entire diffusion model. It is commonly understood that the forward diffusion process is constantly adding Gaussian noise to the image, making it unrecognizable, while the reverse process reduces the noise and then restores the image. The core formula of the diffusion model is

$$x_t = \sqrt{a_t}x_{t-1} + \sqrt{1 - a_t}z_1 \tag{1}$$

where a_t is an experimental constant that decreases as t increases; z_1 is a standard Gaussian noise distribution $N(0, I)$.

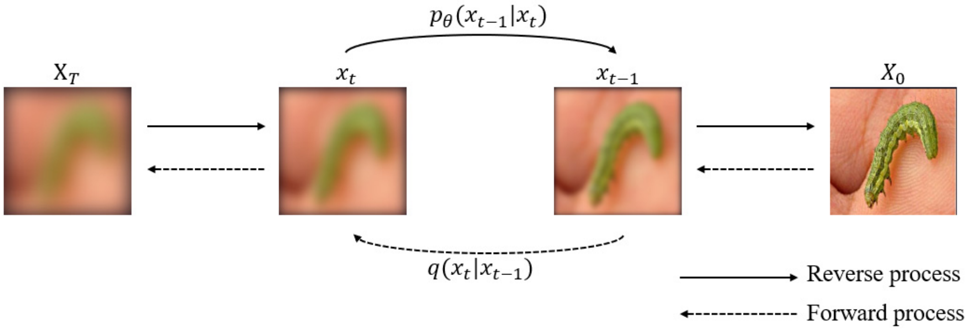


Figure 2. Diffusion processes.

The overall structure of the diffusion model is shown in Figure 3. It contains three models. The first is the CLIP model (Contrastive Language-Image Pre-Training), which is a text encoder that converts text into vectors as input. The image is then generated using the diffusion model. This is performed in the potential space of the compressed image, so the input and output of the expanded model are the image features of the potential space, not the pixels of the image itself. During the training of the latent diffusion model, an encoder is used to obtain the potentials of the picture training set, which are used in the forward diffusion process (each step adds more noise to the latent representation). At inference generation, the decoder part of the VAE (Variational Auto-Encoder) converts the denoised latent signal generated by the reverse diffusion process back into an image format.

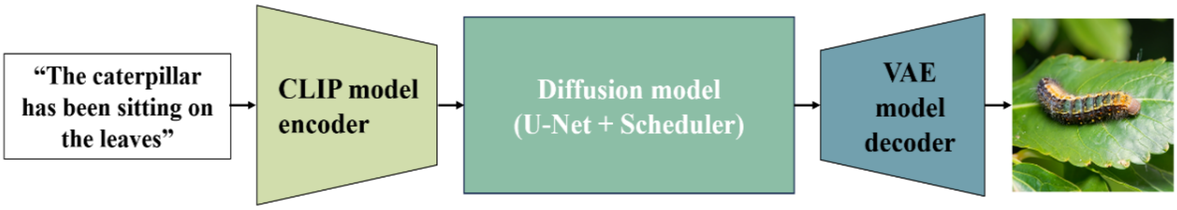

"The caterpillar has been sitting on the leaves"
CLIP model encoder
Diffusion model (U-Net + Scheduler)
VAE model decoder

Figure 3. The framework of the diffusion model.

The stable diffusion model was trained using a real pest dataset. The images generated by stable diffusion are 299 × 299, as shown in Figure 4. To increase the chance of generating pest images, we chose captions that contained any word from the following list of words: [BA, HA, MP, SE, SL, TP, TU, ZC]. We input some keywords and text information into the diffusion model to describe the desired picture, such as pest on the tree, pest on the leaf, pest chewing on the leaf, worm chewing on the trunk, worm swarm, cornfield, leaf, and field. After carefully eliminating the last few false positives, we obtained a dataset of 512 pest images. There were 64 high-resolution images for each pest category.

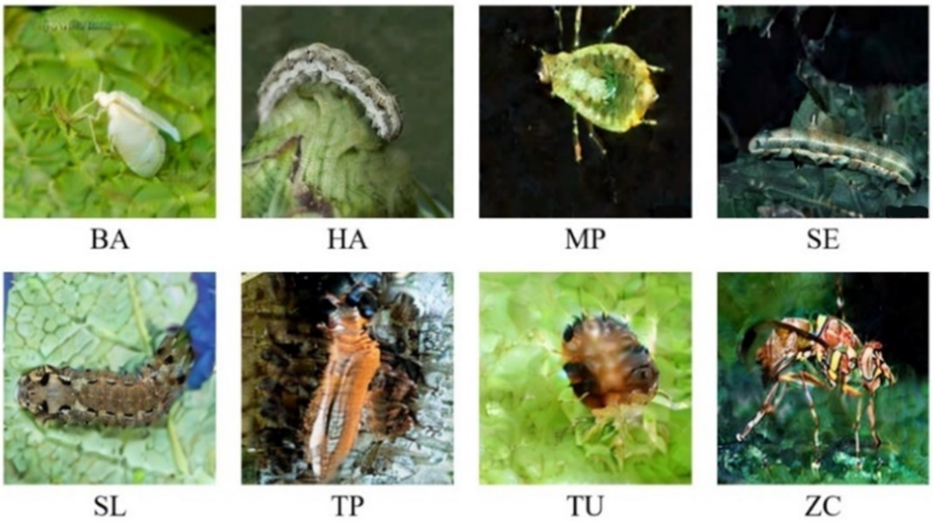


Figure 4. Generated pest dataset.

2.1.3. Dataset Enhancement

In this study, the original image was processed using enhancement methods such as rotation, translation, flipping, and noise addition, and the enhancement technique AutoAugmentation [31] was applied to determine the color of the images. Finally, we obtained 36,504 pest images and the details are shown in Table 2.

Table 2. Details regarding the number of images in the dataset, including generated dataset, real data, and datasets from the internet.

Dataset	Number of Images
Captured images	400
Images from other datasets	609
Generated images	512
Enhanced images	36,504

With the data-enhanced images, we trained RS Transformer. In the first stage, we did not use the generated RGSDD data, and first trained with real images to obtain detailed RS Transformer data. In the second stage, we mixed the generated images in the RGSDD according to the training ratio in Table 3, and we applied this method in YOLOv8, DETR, and other models.

Table 3. Details regarding the number of images using the RGSDD method.

Dataset	Real Images	Generated Images
Primary	24,216	0
10% RGSDD	24,216	1229
20% RGSDD	24,216	2458
30% RGSDD	24,216	3686
40% RGSDD	24,216	4915
50% RGSDD	24,216	12,288

2.2. Framework of the Proposed Method

In this paper, R-CNN [32] is replaced by Swin Transformer and applied to pest target detection tasks. Additionally, we propose a novel object detection method called RS Transformer. Our scheme offers several advantages. Firstly, we introduce a new feature extraction method specifically designed for Swin Transformer, which enhances the alignment of global features. This improvement leads to enhanced localization accuracy, while also significantly reducing the computational cost of the Transformer through the implementation of the shift window model. Secondly, we propose the RS Transformer, which incorporates essential components such as RPN, ROI Align, and feature maps. These additions further enhance the performance and capabilities of the proposed method. Lastly, we propose a new data composition method called RGSDD. This method involves training the stable diffusion model using real images collected beforehand and subsequently generating 512 images by randomly mixing them with 10%, 20%, 30%, 40%, and 50% of the number of real images. Overall, our approach combines the advancements of Swin Transformer, the novel RS Transformer, and the innovative RGSDD data composition method to achieve improved results in pest target detection tasks.

2.3. RS Transformer

RS Transformer is a two-stage model (Figure 5). It first extracts features using Swin Transformer and then generates a series of region proposals.

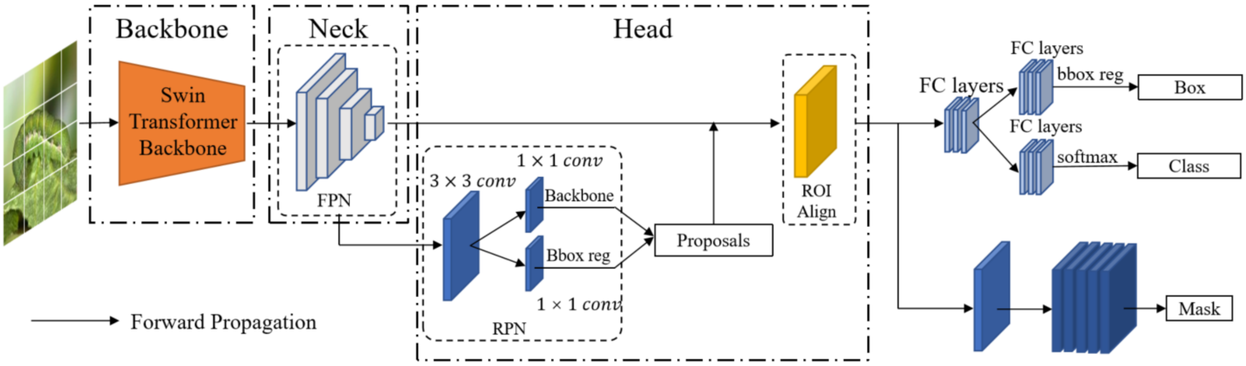


Figure 5. Structure diagram of RS Transformer.

2.3.1. Swin Transformer Backbone

The Swin Transformer backbone is introduced in Figure 6. Compared to traditional CNN models, it has stronger feature extraction capabilities, incorporates CNN's local and hierarchical structure, and utilizes attention mechanisms to produce a more interpretable model and examine the attention distribution.

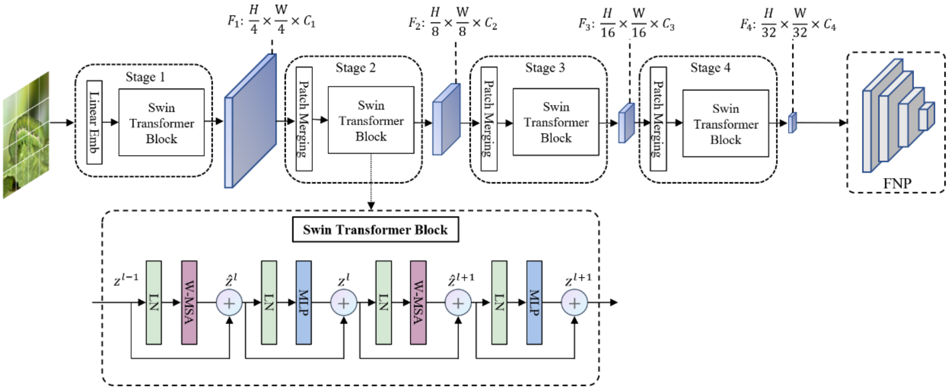


Figure 6. Swin Transformer backbone.

A 2-layer MLP (multi-layer perceptron) with GELU non-linearity follows a shifted-window-based MSA module (W-MSA) in the Swin Transformer block. Each MSA module (multi-head self-attention) and each MLP has an LN (layer norm) layer applied before it, and each module also has a residual connection applied after it. Supposing each window contains $M \times M$ patches, the computational complexities of a global MSA module and image-based window $h \times w$ patches are as follows:

$$\Omega(MSA) = 4hwC^2 + 2(hw)^2C \tag{2}$$

$$\Omega(W - MSA) = 4hwC^2 + 2M^2hwC \tag{3}$$

The shift window partitioning method can be used to compute the backbones of two consecutive Swin Transformers and is denoted as follows:

$$\hat{z}^l = W - MSA(LN(z^{l-1})) + z^{l-1} \tag{4}$$

$$z^l = MLP(LN(\hat{z}^l)) + \hat{z}^l \tag{5}$$

$$\hat{z}^{l+1} = SW - MSA(LN(z^l)) + z^l \tag{6}$$

$$z^{l+1} = MLP(LN(\hat{z}^{l+1})) + \hat{z}^{l+1} \tag{7}$$

where $\hat{z}^l$ and $\hat{z}^{l+1}$ represent the output of W-MSA and MLP of block l, respectively.

Swin Transformer constructs hierarchical feature graphs and adopts a complexity calculation method with a linear image size. A sample diagram of a hierarchy with a small patch size is shown in Figure 7. In the deeper Transformer layers, it begins with small patches and eventually integrates nearby patches. By using patch-splitting modules like ViT, RGB images are divided into non-overlapping patches and employ a patch size of 4×4, making each patch's feature dimension $4 \times 4 \times 3 = 48$. This fundamental feature is projected to any dimension (designated C) using a linear embedding layer.

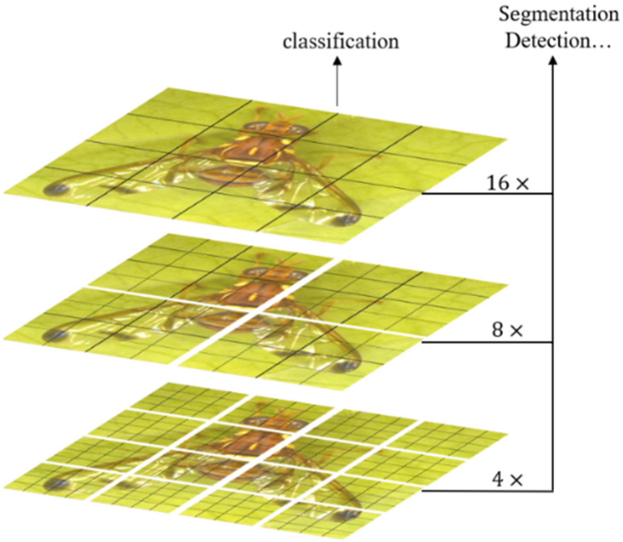


Figure 7. Sample diagram of a hierarchy with a small patch size.

2.3.2. RS Transformer Neck: FPN

An FPN (feature pyramid network) is proposed to achieve a better fusion of feature maps. As illustrated in Figure 8, the purpose of the FPN is to integrate feature maps from the bottom layer to the top layer to fully utilize the extracted features at each stage.

The FPN produces a feature pyramid, not just a feature map. The pyramid after the RPN will produce many region proposals. These region proposals are produced by the RPN, and the ROI is cut out according to the region proposal for subsequent classification and regression prediction. We use a formula to determine from which k the ROI of width w and height h should be cut:

$$k = k_0 + log_2\left(\sqrt{w \times h}/299\right) \tag{8}$$

where 299 represents the size of the image used for pre-training. k_0 represents the level at which the ROI of the area is $w \times h = 299 \times 299$. A large-scale ROI should be cut from a feature map of low resolution, which is conducive to the detection of large targets, and a small-scale ROI should be cut from a feature map of high resolution, which is conducive to the detection of small targets.

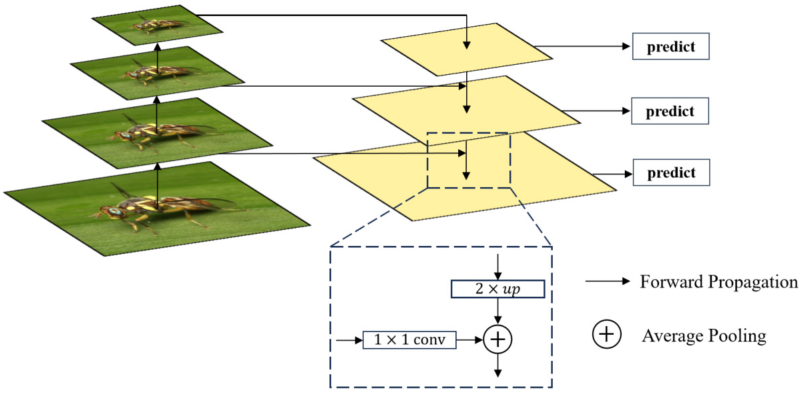


Figure 8. FPN structure diagram.

2.3.3. RS Transformer Head: RPN, ROI Align

To achieve the prediction of coordinates and scores of each regional suggestion box while extracting features, the RPN network adds a regression layer (reg-layer) and a classification layer (cls-layer) to Swin Transformer. Figure 9 depicts the RPN working principle. The RPN centers on a pixel of the last layer feature map and traverses the feature map through a 3 × 3 sliding window. The pixel points mapped from the center of the sliding window to the original image are anchor points. Taking the anchor point as the original image center, using 15 preset anchor boxes with 5 different areas (32 × 32, 64 × 64, 128 × 128, 256 × 256, 512 × 512) and 3 distinct aspect ratios (2:1, 1:1, and 1:2), the original candidate region k = 15 is obtained. The RPN sends the candidate regions in the k anchor boxes to the regression layer and the category layer, respectively, for boundary regression and classification prediction. The regression layer predicts the frame coordinates (X, Y, W, H), so the output is 4k; the classification layer predicts the type, target, and background, so the output is 2k. Each anchor is then evaluated with initial over-boundary screening and non-maximum suppression (NMS) from largest to smallest to retain the top 1000 or 2000 scores. Finally, the candidate boundaries of prediction as the background in the classification layer are removed, and the candidate boundaries of prediction as a target are retained.

ROI Align

The function of ROI Pool and ROI Align is to find the feature map corresponding to the candidate box and then process a feature map of different size proportions into a fixed size, so that it can be input into the subsequent fixed-size network. Mask RCNN proposes an ROI alignment [33] based on ROI Pool. The bilinear interpolation method is used to determine the eigenvalue of each pixel in the region of interest of the original image, which avoids the error caused by quantization operation and improves the accuracy of frame prediction and mask prediction.

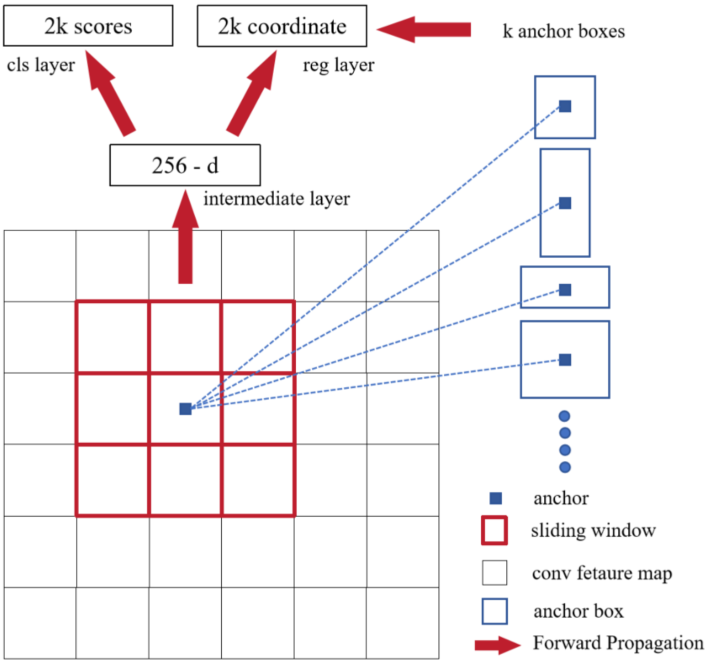


Figure 9. RPN working principle diagram.

The ROI Align algorithm's primary steps are as follows: (1) Each candidate region is traversed on the feature map, keeping the floating-point boundary unquantized. (2) In Figure 10, the candidate region is evenly divided into k × k bins, and the edge of each bin retains the floating-point number without quantization. (3) In this step, 2 × 2 sample points are taken for each bin, and the bilinear interpolation method is used to calculate the pixel values of each sampling point's neighboring four pixels. (4) Finally, the pixel value in each bin is maximized to obtain the value of each bin.

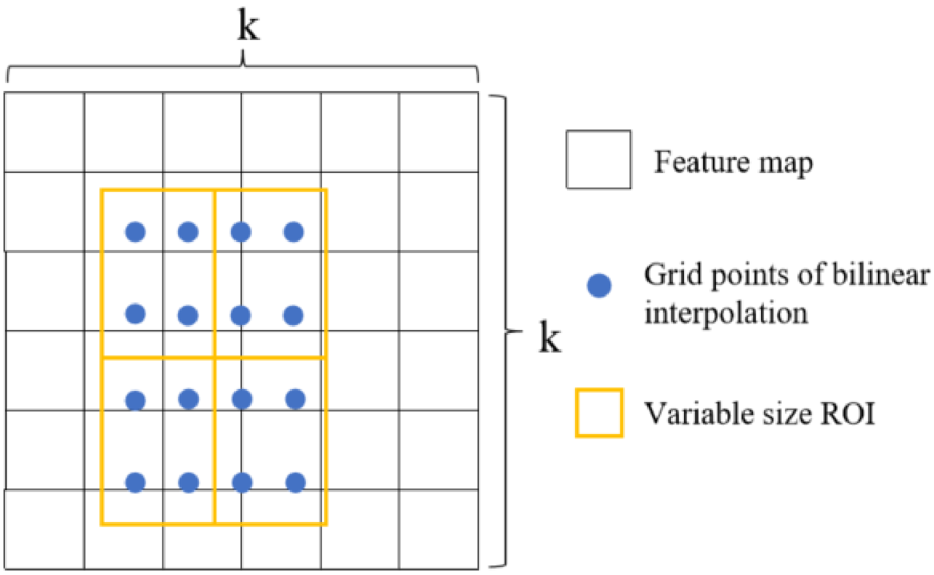


Figure 10. ROI Align diagram.

2.4. Experimental Setup

Experiments were conducted on the Autodl platform, which provides low-cost GPU computing power and a configuration environment that can be rented at any time. For researchers and universities without high-performance GPUs or servers, Autodl offers a wide range of high-performance GPUs to use. The experiments were implemented using the Pytorch 1.10.0 framework, Python 3.8, CUDA 11.3, and Nvidia RTX 2080Ti GPUs with 11 GB memory.

2.5. Evaluation Indicators

To evaluate the performance of the proposed model, we used the accuracy, precision, recall, average precision (AP), mAP, and F1 score:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$Percision = \frac{TP}{FP + TP} \tag{10}$$

where *TP* indicates true positive, *FP* indicates false positive, and *FN* indicates false negative.

Average precision (*AP*): The average precision under different recall rates. The higher the accuracy, the higher the AP.

$$AP = \int_0^1 p(r)dr = \frac{TP}{TP + FP} \tag{11}$$

Recall: The average recall rate at different levels of precision. The higher the recall, the higher the AR.

$$Recall = \frac{TP}{FN + TP} \tag{12}$$

mAP: The picture categorization procedure is usually a multi-classification problem. According to the above calculation process, the *AP* of each analog is obtained, and the average value is the *mAP*.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{13}$$

The F_1 *score* is a metric that combines precision and recall to evaluate the performance of a binary classification model.

$$F_1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

2.6. Experimental Baselines

To evaluate the performance of RS Transformer, SSD [7], Faster R-CNN [9], YOLOv3 [4], YOLOv4 [5], YOLOv5m [6], YOLOv8, and DETR [34] were chosen as baseline models for comparison, as shown in Table 4.

Table 4. Different baselines.

Model	Backbone	Parameters (M)
SSD	VGG16	28.32
Faster R-CNN	VGG16	138
YOLOv3	Darknet-53	64.46
YOLOv4	CSPDarknet53	5.55
YOLOv5m	CSPDarknet53	20.66
YOLOv8	C2f	30.13
DETR	ResNet-50	40.34
RS Transformer	Swin Transformer	30.17

3. Results and Discussion

3.1. Experimental Results and Analysis

On a dataset with eight models, we assessed the performance of popular deep learning models to illustrate the performance of the proposed model (Table 5). We used a fixed image resolution with a size of 299 × 299 pixels.

Table 5. Comparison of different indexes.

Model	mAP (%)	F1 Score (%)	Recall (%)	Precision (%)	Accuracy (%)	mDT (ms)
SSD	76.91	67.62	70.12	66.23	77.11	22.9
Faster R-CNN	72.65	65.57	69.31	67.10	73.52	24.5
YOLOv3	60.38	52.38	57.78	53.64	60.32	17.7
YOLOv4	76.31	69.55	74.97	68.91	76.99	10.7
YOLOv5m	80.29	75.58	79.14	77.33	79.35	13.6
YOLOv8	84.72	80.32	82.11	79.59	83.49	9.8
DETR	85.56	81.18	82.82	80.43	86.12	19.2
RS Transformer	90.18	85.89	87.31	89.91	90.08	20.1

Compared to other models, our proposed method achieved significant improvements, with an mAP of 90.18%, representing gains of 13.27%, 17.53%, 29.8%, 13.97%, 9.89%, 5.46%, and 4.62% over SSD, Faster R-CNN, YOLOv3, YOLOv4, YOLOv5m, YOLOv8, and DETR, respectively. The proposed method achieved 20.1 ms mDT for the detection time of each image.

To visually analyze the classification results of each pest in RS Transformer, we utilized a confusion matrix as shown in Figure 11. These data were obtained using real images for training. The confusion matrix provides an intuitive representation of the classification performance. In the matrix, rows represent predicted pest categories, columns represent actual pest categories, and the values on the main diagonal represent the classification accuracy for each category. From the confusion matrix diagram, it can be observed that the color on the main diagonal of the RS Transformer's confusion matrix is the darkest, indicating the highest values in each row and column. This indicates that RS Transformer exhibits excellent classification performance for each type of pest.

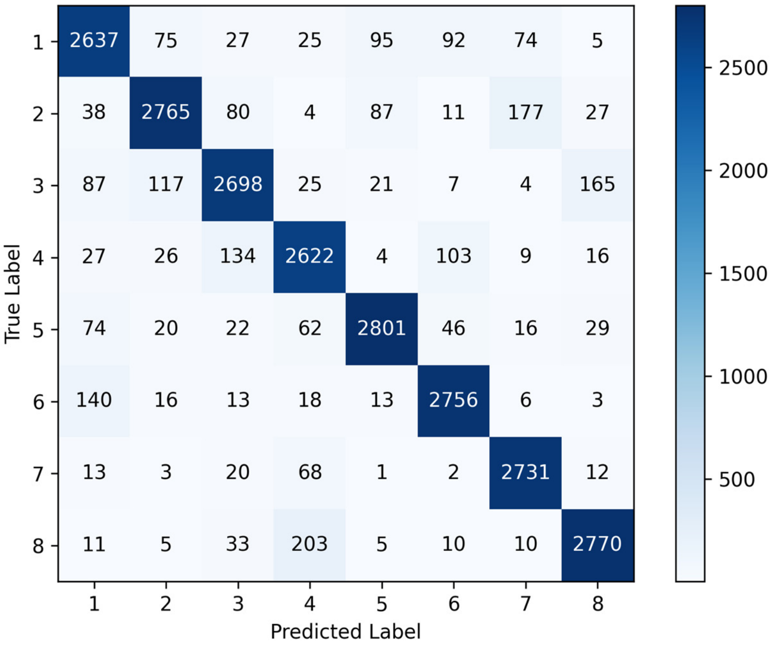


Figure 11. Confusion matrix for RS Transformer.

The contrast in mAP is visually presented in Figure 12. It is evident that the mAP of the three compared models exhibits an upward trend during the training process, albeit with substantial fluctuations. Conversely, our model's mAP shows a more consistent trajectory, stabilizing at 77.73% after approximately 75 epochs. Subsequently, the RS Transformer model attains its peak performance, achieving a maximum mAP of 90.18%. These findings collectively confirm the stability of RS Transformer, its capacity to enhance network performance, and its ability to expedite convergence.

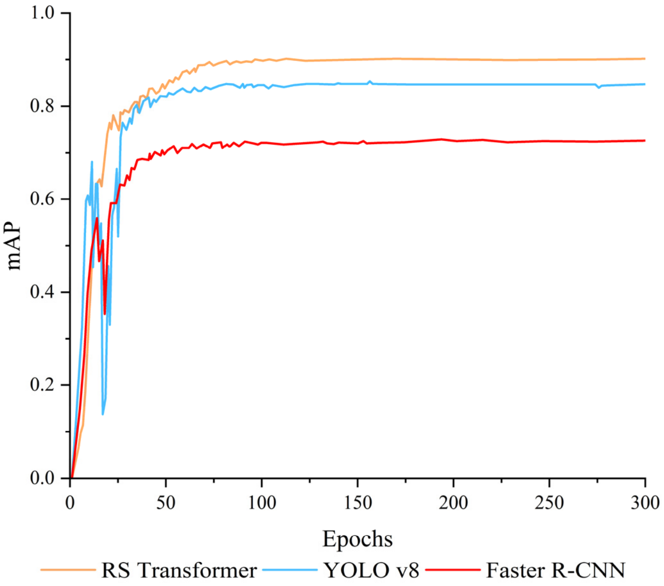


Figure 12. Comparisons of mAP.

RS Transformer exhibits a robust capacity for discerning similar pests and demonstrates superior overall performance compared to other models, as detailed in Table 6 (models' mAP) and illustrated in Figure 13. Furthermore, in challenging scenarios such as the TU dataset, the model maintains a remarkable recognition rate of 90.24%.

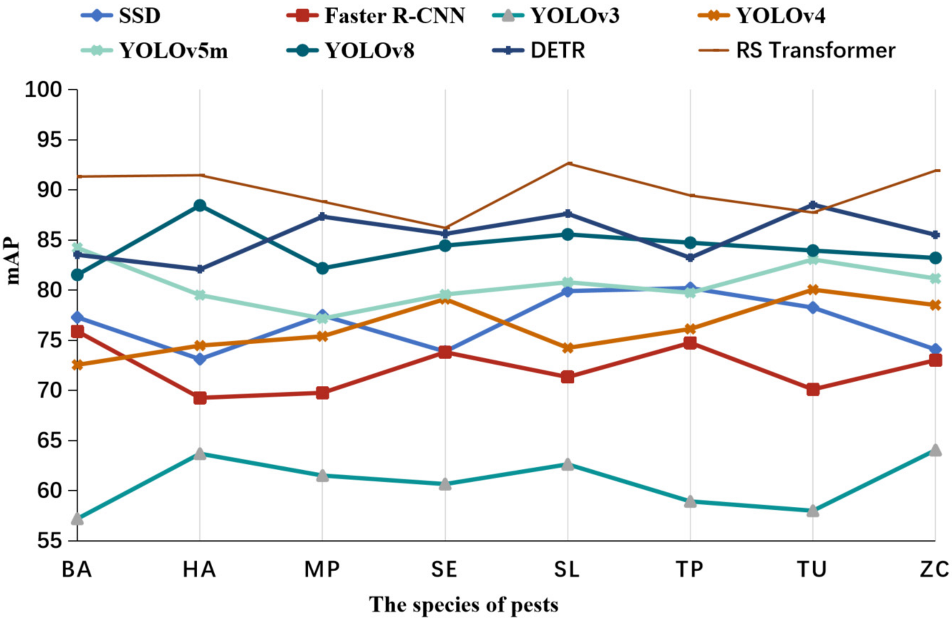


Figure 13. Comparison of mAPs to identify similar pests.

Table 6. Comparison of different mAP indexes.

Model	BA	HA	MP	SE	SL	TP	TU	ZC
SSD	77.29	73.12	77.48	73.88	79.91	80.21	78.26	74.08
Faster R-CNN	75.89	69.26	69.76	73.81	71.33	74.75	70.10	73.02
YOLOv3	57.20	63.69	61.51	60.66	62.63	58.93	58.00	64.05
YOLOv4	72.55	74.47	75.40	79.11	74.24	76.13	80.05	78.51
YOLOv5m	84.22	79.51	77.17	79.57	80.79	79.73	83.06	81.16
YOLOv8	81.53	88.45	82.18	84.44	85.56	84.73	83.95	83.21
DETR	83.53	82.07	87.33	85.61	87.62	83.23	88.52	85.52
RS Transformer	91.33	91.46	88.83	86.21	92.63	89.44	87.74	91.92

The dataset was generated using the diffusion model (see Figure 14) and subsequently combined at varying proportions of 10%, 20%, 30%, 40%, and 50%. These datasets were then utilized as inputs for the RS Transformer model, followed by rigorous testing procedures, culminating in the presentation of the results in Table 7.

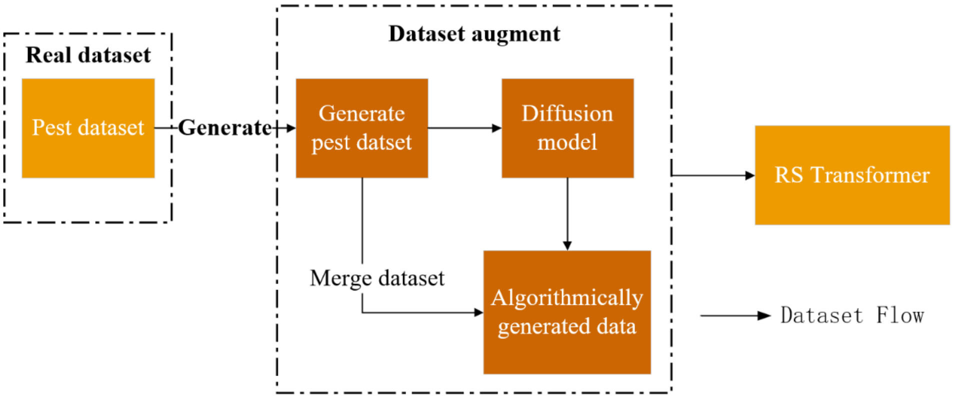


Figure 14. Mixed data model diagram.

Table 7. RGSDD using RS Transformer.

Model	Percentage	mAP (%)	F1 Score (%)	Recall (%)	mDT (ms)
RS Transformer	0%	90.18	85.89	87.31	20.1
	10%	90.98	85.13	83.53	20.1
	20%	93.64	86.75	90.42	20.1
	30%	95.71	94.82	92.47	20.2
	40%	95.56	90.67	93.10	20.2
	50%	94.98	91.03	93.06	20.2

Applying the RGSDD method to RS Transformer, it is evident that upon incorporating 30% of the generated data, the model attains its peak performance, resulting in a notable increase of 5.53% in mAP.

The RGSDD methodology was also applied to enhance the performance of the Faster R-CNN, YOLOv5m, YOLOv8, and DETR models. The results of these experiments demonstrate that RGSDD positively contributes to model enhancement, as evidenced in Tables 8–11.

These data underscore the practical applicability of RGSDD, as shown in Figure 15. Specifically, in the case of the YOLOv8 model with 30% incorporation, it yielded a substantial 3.79% improvement in mAP. Similarly, for the DETR model with 40% incorporation, there was a noticeable enhancement of 4.36% in mAP. Furthermore, it is evident that when 50% of the generated data are included, the model's performance experiences a significant decline. This subset of data appears to introduce interference and is potentially treated as noise to some extent, resulting in adverse effects on model performance.

Table 8. RGSDD using Faster R-CNN.

Model	Percentage	mAP (%)	F1 Score (%)	Recall (%)	mDT (ms)
Faster R-CNN	0%	72.65	65.57	69.31	24
	10%	75.07	68.83	69.73	24
	20%	73.47	67.26	70.62	24
	30%	73.72	67.37	74.84	24
	40%	71.80	69.78	72.39	24.1
	50%	73.13	68.29	70.47	24.1

Table 9. RGSDD using YOLOv5m.

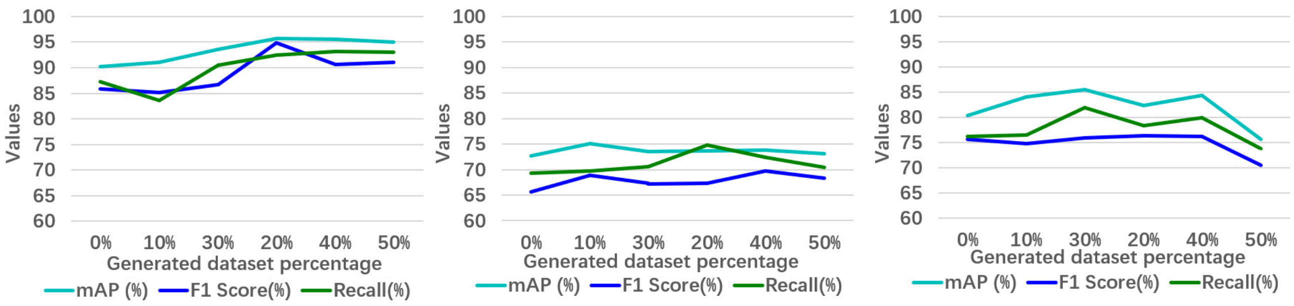
Model	Percentage	mAP (%)	F1 Score (%)	Recall (%)	mDT (ms)
YOLOv5m	0%	80.29	75.58	76.14	13.6
	10%	83.96	74.72	76.48	13.6
	20%	85.43	75.90	81.91	13.6
	30%	82.31	76.24	78.38	13.6
	40%	84.37	76.12	79.82	13.7
	50%	75.53	70.41	73.76	13.7

Table 10. RGSDD using YOLOv8.

Model	Percentage	mAP (%)	F1 Score (%)	Recall (%)	mDT (ms)
YOLOv8	0%	84.72	80.32	82.11	9.8
	10%	87.38	75.77	72.31	9.8
	20%	88.42	85.17	84.78	9.8
	30%	88.51	85.89	85.31	9.8
	40%	82.32	81.76	80.11	9.9
	50%	75.35	70.32	71.58	9.9

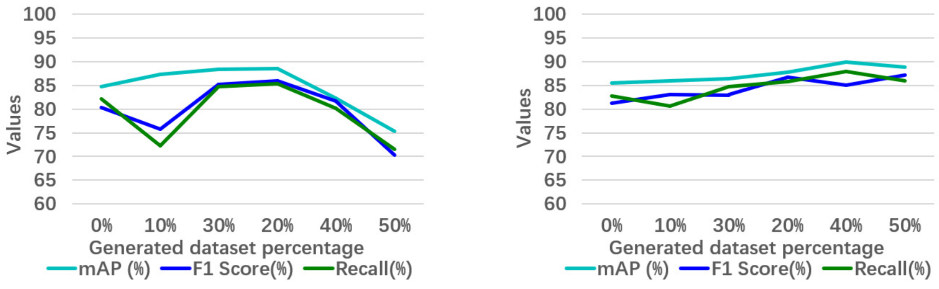
Table 11. RGSDD using DETR.

Model	Percentage	mAP (%)	F1 Score (%)	Recall (%)	mDT (ms)
DETR	0%	85.56	81.18	82.82	20.1
	10%	85.94	83.10	80.62	20.1
	20%	86.37	82.99	84.67	20.1
	30%	87.71	86.75	85.72	20.2
	40%	89.92	85.02	87.89	20.2
	50%	88.90	87.19	85.97	20.2



(a) (b) (c)

Figure 15. *Cont*.



(d) (e)

Figure 15. (**a**) RS Transformer with RGSDD, (**b**) Faster R-CNN with RGSDD, (**c**) YOLOv5m with RGSDD, (**d**) YOLOv8 with RGSDD, and (**e**) DETR with RGSDD.

Comparing the mAP, F1 score, and recall of different networks, it can be seen that RS Transformer is still better than the others, even when the RGSDD is used. At the optimal value, mAP outperforms Faster R-CNN by 9.29% and YOLOv5m by 4.95%.

Figure 16 presents the outcomes achieved by the RS Transformer model integrated with the RGSDD. Notably, the results highlight the RGSDD's exceptional accuracy in effectively identifying multi-scale pests across various species.

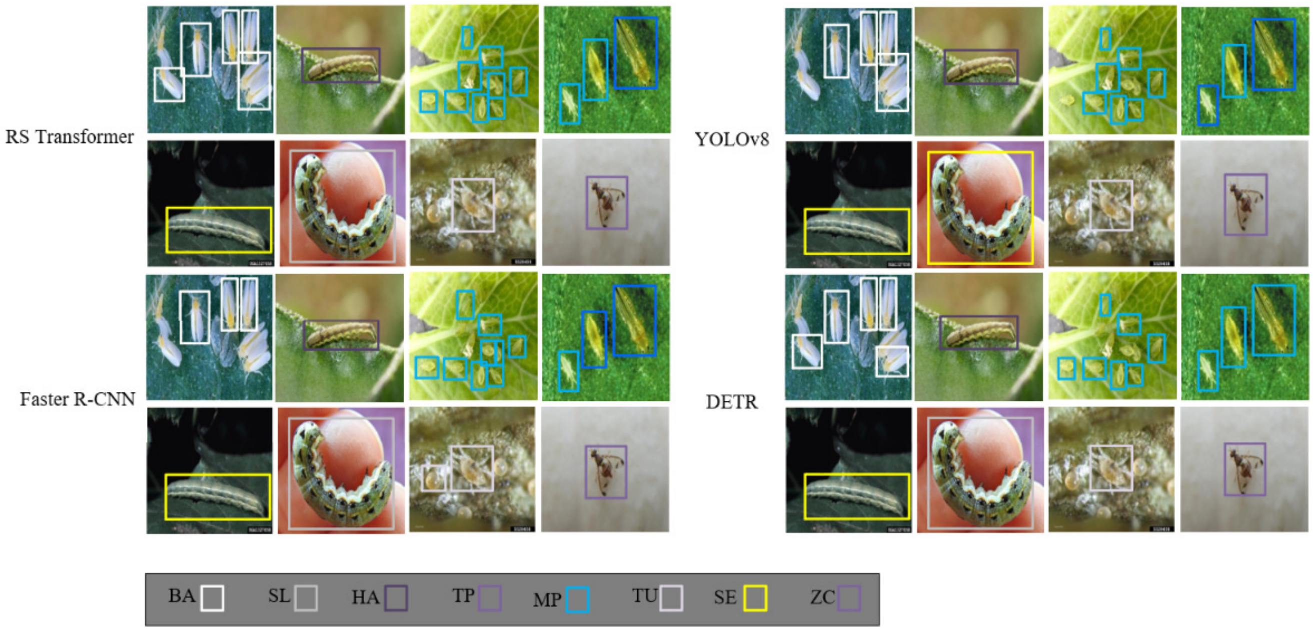


Figure 16. RS Transformer, Faster R-CNN, YOLOv8, and DETR output through the RGSDD system.

3.2. Comparison Results Summary

The performance comparison of the proposed method with other existing methods for eight pest datasets is shown in Table 12. Setiawan et al. [35] applied a CNN and MoblieNetV2. They used the Adam optimizer for large-scale pest classification and achieved an accuracy of 82.95% for eight classes of agriculture pests. Their model was trained for large-scale pest classification. However, due to the CNN, the ideal effect was not achieved in the case of large-scale differences in pest images. Liu et al. [36] used a novel Transformer auto-encoder to capture the features and benefits in the classification accuracy. In the case of eight pest images, as well as small samples, the method proposed by the authors reached 85.17% for mAP. We can see that models such as Vision in Transformer (ViT) models that require large datasets for training do not work well on datasets containing images of small targets such as pests. In this case, it is difficult for ViT to capture image features, resulting in inaccurate recognition. At the same time, the field environment is complex, and the image quality is full of uncertainties due to the large influence of factors such as sunlight and region when taking pictures, which lead to reductions in accuracy. In order to improve the accuracy of other models, we mixed the pest pictures generated by the RGSDD into the total training dataset in a 30% proportion, and we found that Setiawan et al. [35]'s method was significantly improved by 6.40% and Liu et al.'s method was improved by 3.06%, which proved the universality and practicability of the RGSDD method. From the experimental results, our proposed method comprising RS Transformer and the RGSDD provides good performance in few-shot learning for pest classification.

Table 12. Related work and accuracy results (%) summary.

Model	Method	Dataset	RGSDD	mAP
Setiawan et al. [35]	CNN, MobileNetV2	8 pests	×	82.95
Liu et al. [36]	ViT		×	85.17
Our proposal	Swin Transformer		×	90.18
Setiawan et al. [35]	CNN, MobileNetV2		30%	89.35
Liu et al. [36]	ViT		30%	88.23
Our proposal	Swin Transformer		30%	95.71

3.3. Discussion

In the analysis of the results, it was clearly shown that RS Transformer performed well. Since Swin Transformer was proposed, which performed better than the CNN did, a large number of application algorithms based on Swin Transformer have been proposed [37–39]. However, a common feature among these algorithms is that a large number of datasets are required to train Swin Transformer to realize its ability to extract features globally. Therefore, we added an FPN, RPN, and ROI Align on the basis of Swin Transformer, which reduces the computational complexity and improves the feature extraction capability. Then, using the RGSDD method to generate a dataset to assist with training, we not only achieved the purpose of expanding the dataset, but also improved the training accuracy of the model. The RS Transformer achieved 9.08% accuracy, which was higher than that of the DETR universal model at 1.41% and higher than that of the YOLOv8 model at 6.59%. Its superior multi-scale feature extraction capabilities effectively help improve accuracy.

In a two-stage model like that of Dong [40], the author used ResNet-50 as a backbone. Even though the model was improved and deep convolutional neural networks (DCNNs) were used, it still failed to achieve ideal results at a small scale, with an mAP value of only 67.9%. Jiao [22] used VGG-16 as a backbone and trained with a large number of datasets comprising about 25.4k images. However, Jiao only obtained an mAP of 56.40%. In a large number of training datasets, the algorithms proposed by the authors still fail to reach the required application. On the one hand, the pest scale is small; on the other hand, the feature extraction ability of the CNN is limited. In deep learning, we explain which backbone or which model has absolute advantages in an application field, but in our experiment, we found that RS Transformer does have certain advantages.

Before this study, there was no research on agricultural pest identification based on AIGC. For the first time, we used a diffusion model for agricultural pest training and image generation and achieved unexpectedly good results. After adding 30% of the generated images, RS Transformer; YOLOv3, 4, 5, and 8; and DETR were all improved, up to 8.93%. This kind of high-resolution generated image is less noisy, is more conducive to model training, and helps to quickly locate and extract effective features.

In general, the quality and size of the dataset, the appropriate improvement strategy, and the underlying model architecture all have important effects on the detection accuracy. A multi-stage algorithm is faster and has a lighter weight on the basis of ensuring accuracy, while a single-stage algorithm improves the detection accuracy on the basis of maintaining the advantages of speed and model size. Achieving higher performance levels and achieving a balance of performance such as accuracy, speed, and magnitude are the current trends.

4. Conclusions

Swin Transformer, introduced here as the foundational network for pest detection, represents a pioneering contribution. In conjunction with this innovation, RS Transformer was developed, building upon the inherent strengths of the R-CNN framework. Furthermore, we employed a diffusion model to create a novel pest dataset, accompanied by introducing an innovative training approach tailored for the Randomly Generated Stable Diffusion Dataset (RGSDD). This approach involves the judicious fusion of synthetic data generated through the RGSDD with real data, calibrated as a percentage of the total dataset. Our study comprehensively compared the performance of RS Transformer and the RGSDD against established models including SSD, Faster R-CNN, YOLOv3, YOLOv4, YOLOv5m, YOLOv8, and DETR. The experimental results unequivocally demonstrate the superiority of RS Transformer and the efficacy of the RGSDD dataset, surpassing prevailing benchmarks. Importantly, our method achieves an optimal balance between accuracy and network characteristics. These findings have substantial implications for future ecological informatics research, offering fresh insights into the domain of ecological pest and disease control. The presented approach promises to advance the state of the art and contribute to more effective ecological management strategies.

RS Transformer can be used not only for agricultural pest detection, but also for multi-scale target detection tasks in complex environments such as transportation, medicine, and industrial devices. In addition, the RGSDD, an image generation method based on a diffusion model, is helpful for expanding the dataset and improving accuracy. Hopefully, we can undertake more research based on the method in this paper in the future.

Author Contributions: T.W.: Conceptualization, software, validation, formal analysis, investigation, data curation, writing—original draft preparation and visualization; L.S.: Methodology, writing—review and editing; L.Z.: Conceptualization, methodology, resources, writing—review and editing, supervision; X.W.: Data curation, visualization; J.L.: Supervision, resource; Z.L.: funding acquisition, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program "Industrial Software" Key Special Project (2022YFB3305602), Social Science Planning Foundation of Beijing(20GLC059), and Humanities and Social Sciences Planning Fund of the Ministry of Education (22YJA630111, 22YJAZH110).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy restrictions.

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

References

1. Merle, I.; Hipólito, J.; Requier, F. Towards integrated pest and pollinator management in tropical crops. *Curr. Opin. Insect Sci.* **2022**, *50*, 100866. [CrossRef] [PubMed]
2. Kannan, M.; Bojan, N.; Swaminathan, J.; Zicarelli, G.; Hemalatha, D.; Zhang, Y.; Ramesh, M.; Faggio, C. Nanopesticides in agricultural pest management and their environmental risks: A review. *Int. J. Environ. Sci. Technol.* **2023**, *20*, 10507–10532. [CrossRef]
3. Bras, A.; Roy, A.; Heckel, D.G.; Anderson, P.; Karlsson Green, K. Pesticide resistance in arthropods: Ecology matters too. *Ecol. Lett.* **2022**, *25*, 1746–1759. [CrossRef]
4. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *Int. J. Comput. Vis.* **2018**, *127*, 74–91. [CrossRef]
5. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. *Electronics* **2020**, *9*, 1719. [CrossRef]
6. Wang, J.; Chen, Y.; Dong, Z.; Gao, M. Improved YOLOv5 network for real-time multi-scale traffic sign detection. *Neural. Comput. Appl.* **2023**, *35*, 7853–7865. [CrossRef]

7. 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 European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37. [CrossRef]
8. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1440–1448. [CrossRef]
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Montreal, QC, Canada, 7–12 December 2015; Curran Associates, Inc.: Red Hook, NY, USA, 2015; pp. 91–99.
10. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 5998–6008.
11. Brasoveanu, A.M.P.; Andonie, R. Visualizing Transformers for NLP: A Brief Survey. In Proceedings of the 2020 24th International Conference Information Visualisation (IV), Melbourne, Australia, 7–11 September 2020; IEEE: Melbourne, Australia, 2020; pp. 270–279.
12. 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 2021. *arXiv* **2021**, arXiv:2010.11929. [CrossRef]
13. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows 2021. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 17–21 October 2021. [CrossRef]
14. Li, W.; Zheng, T.; Yang, Z.; Li, M.; Sun, C.; Yang, X. Classification and Detection of Insects from Field Images Using Deep Learning for Smart Pest Management: A Systematic Review. *Ecological. Inform.* **2021**, *66*, 101460. [CrossRef]
15. Sohl-Dickstein, J.; Weiss, E.A.; Maheswaranathan, N.; Ganguli, S. Deep Unsupervised Learning Using Nonequilibrium Thermodynamics 2015. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015. [CrossRef]
16. Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Zhang, W.; Cui, B.; Yang, M.-H. Diffusion Models: A Comprehensive Survey of Methods and Applications. *ACM Comput. Surv.* **2022**, *10*, 123–145. [CrossRef]
17. Aggarwal, A.; Mittal, M.; Battineni, G. Generative Adversarial Network: An Overview of Theory and Applications. *Int. J. Inf. Manag. Data Insights* **2021**, *1*, 100004. [CrossRef]
18. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models 2020. *arXiv* **2020**, arXiv:2006.11239. [CrossRef]
19. Dhariwal, P.; Nichol, A. Diffusion Models Beat GANs on Image Synthesis. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 8780–8794. [CrossRef]
20. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125.
21. Liu, L.; Wang, R.; Xie, C.; Yang, P.; Wang, F.; Sudirman, S.; Liu, W. PestNet: An End-to-End Deep Learning Approach for Large-Scale Multi-Class Pest Detection and Classification. *IEEE Access* **2019**, *7*, 45301–45312. [CrossRef]
22. Jiao, L.; Dong, S.; Zhang, S.; Xie, C.; Wang, H. AF-RCNN: An Anchor-Free Convolutional Neural Network for Multi-Categories Agricultural Pest Detection. *Comput. Electron. Agric.* **2020**, *174*, 105522. [CrossRef]
23. Pattnaik, G.; Shrivastava, V.K.; Parvathi, K. Transfer Learning-Based Framework for Classification of Pest in Tomato Plants. *Appl. Artif. Intell.* **2020**, *34*, 981–993. [CrossRef]
24. Lee, S.; Lin, S.; Chen, S. Identification of Tea Foliar Diseases and Pest Damage under Practical Field Conditions Using a Convolutional Neural Network. *Plant Pathol.* **2020**, *69*, 1731–1739. [CrossRef]
25. Chen, C.-J.; Huang, Y.-Y.; Li, Y.-S.; Chen, Y.-C.; Chang, C.-Y.; Huang, Y.-M. Identification of Fruit Tree Pests with Deep Learning on Embedded Drone to Achieve Accurate Pesticide Spraying. *IEEE Access* **2021**, *9*, 21986–21997. [CrossRef]
26. Wang, R.; Jiao, L.; Xie, C.; Chen, P.; Du, J.; Li, R. S-RPN: Sampling-Balanced Region Proposal Network for Small Crop Pest Detection. *Comput. Electron. Agric.* **2021**, *187*, 106290. [CrossRef]
27. Peng, Y.; Wang, Y. CNN and Transformer Framework for Insect Pest Classification. *Ecol. Inform.* **2022**, *72*, 101846. [CrossRef]
28. Ullah, N.; Khan, J.A.; Alharbi, L.A.; Raza, A.; Khan, W.; Ahmad, I. An Efficient Approach for Crops Pests Recognition and Classification Based on Novel DeepPestNet Deep Learning Model. *IEEE Access* **2022**, *10*, 73019–73032. [CrossRef]
29. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection 2021. *arXiv* **2020**, arXiv:2010.04159. [CrossRef]
30. Letourneau, D.K.; Goldstein, B. Pest Damage and Arthropod Community Structure in Organic vs. Conventional Tomato Production in California. *Arthropod. Community Struct. J. Appl. Ecol.* **2001**, *38*, 557–570. [CrossRef]
31. Cubuk, E.D.; Zoph, B.; Mane, D.; Vasudevan, V.; Le, Q.V. AutoAugment: Learning Augmentation Strategies from Data. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 5 June 2019; IEEE: Long Beach, CA, USA, 2019; pp. 113–123.
32. Thenmozhi, K.; Srinivasulu Reddy, U. Crop Pest Classification Based on Deep Convolutional Neural Network and Transfer Learning. *Comput. Electron. Agric.* **2019**, *164*, 104906. [CrossRef]
33. Gong, T.; Chen, K.; Wang, X.; Chu, Q.; Zhu, F.; Lin, D.; Yu, N.; Feng, H. Temporal ROI Align for Video Object Recognition. *AAAI* **2021**, *35*, 1442–1450. [CrossRef]

34. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-End Object Detection with Transformers. In *Computer Vision–ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2020; Volume 12346, pp. 213–229, ISBN 978-3-030-58451-1.
35. Setiawan, A.; Yudistira, N.; Wihandika, R.C. Large Scale Pest Classification Using Efficient Convolutional Neural Network with Augmentation and Regularizers. *Comput. Electron. Agric.* **2022**, *200*, 107204. [CrossRef]
36. Liu, H.; Zhan, Y.; Xia, H.; Mao, Q.; Tan, Y. Self-Supervised Transformer-Based Pre-Training Method Using Latent Semantic Masking Auto-Encoder for Pest and Disease Classification. *Comput. Electron. Agric.* **2022**, *203*, 107448. [CrossRef]
37. Huang, J.; Fang, Y.; Wu, Y.; Wu, H.; Gao, Z.; Li, Y.; Ser, J.D.; Xia, J.; Yang, G. Swin Transformer for Fast MRI. *Neurocomputing* **2022**, *493*, 281–304. [CrossRef]
38. Lin, A.; Chen, B.; Xu, J.; Zhang, Z.; Lu, G.; Zhang, D. DS-TransUNet: Dual Swin Transformer U-Net for Medical Image Segmentation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–15. [CrossRef]
39. He, X.; Zhou, Y.; Zhao, J.; Zhang, D.; Yao, R.; Xue, Y. Swin Transformer Embedding UNet for Remote Sensing Image Semantic Segmentation. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]
40. Dong, S.; Wang, R.; Liu, K.; Jiao, L.; Li, R.; Du, J.; Teng, Y.; Wang, F. CRA-Net: A Channel Recalibration Feature Pyramid Network for Detecting Small Pests. *Comput. Electron. Agric.* **2021**, *191*, 106518. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.