

Article

Enhanced Pest Recognition Using Multi-Task Deep Learning with the Discriminative Attention Multi-Network

Zhaojie Dong ¹, Xinyu Wei ², Yonglin Wu ³, Jiaming Guo ¹ and Zhixiong Zeng ^{1,*}

¹ College of Engineering, South China Agricultural University, Guangzhou 510642, China; kelaibing@stu.scau.edu.cn (Z.D.); jmguo@scau.edu.cn (J.G.)

² Institute of Facility Agriculture, Guangdong Academy of Agricultural Sciences (IFA, GDAAS), Guangzhou 510640, China; weixinyu@gdaas.cn

³ College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China; 517673647@stu.scau.edu.cn

* Correspondence: zhixzeng@scau.edu.cn; Tel.: +86-20-8528-2860

Abstract: Accurate recognition of agricultural pests is crucial for effective pest management and reducing pesticide usage. In recent research, deep learning models based on residual networks have achieved outstanding performance in pest recognition. However, challenges arise from complex backgrounds and appearance changes throughout the pests' life stages. To address these issues, we develop a multi-task learning framework utilizing the discriminative attention multi-network (DAM-Net) for the main task of recognizing intricate fine-grained features. Additionally, our framework employs the residual network-50 (ResNet-50) for the subsidiary task that enriches texture details and global contextual information. This approach enhances the main task with comprehensive features, improving robustness and precision in diverse agricultural scenarios. An adaptive weighted loss mechanism dynamically adjusts task loss weights, further boosting overall accuracy. Our framework achieves accuracies of 99.7% on the D0 dataset and 74.1% on the IP102 dataset, demonstrating its efficacy in training high-performance pest-recognition models.

Keywords: deep learning; pest recognition; multi-task learning



Citation: Dong, Z.; Wei, X.; Wu, Y.; Guo, J.; Zeng, Z. Enhanced Pest Recognition Using Multi-Task Deep Learning with the Discriminative Attention Multi-Network. *Appl. Sci.* **2024**, *14*, 5543. <https://doi.org/10.3390/app14135543>

Academic Editors: Salik Khanal and Vitor Filipe

Received: 1 June 2024

Revised: 22 June 2024

Accepted: 25 June 2024

Published: 26 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As the global population continues to grow rapidly, ensuring a sufficient and varied food supply is imperative. Agricultural pests present a significant threat to the yield and quality of crops, making accurate pest recognition vital for effective crop protection and yield enhancement [1,2]. In the realm of precision agriculture, the precise recognition of pest species is fundamental to devising optimal pest management strategies [3]. Traditionally, pest recognition has been manual, dependent heavily on the expertise of the observer, and often lacks consistency in accuracy and efficiency when dealing with large pest populations [4]. Advances in traditional machine learning have greatly enhanced automated pest-recognition technologies by implementing advanced feature extraction and model training techniques, which are essential for precise and efficient pest recognition [5–7]. However, these methods are vulnerable to variations in image quality, target posture, and environmental conditions such as lighting and background complexity, reducing their effectiveness in diverse and complex scenarios. Moreover, they typically rely on predefined feature extraction techniques that struggle to adapt to the variability of pests in natural settings, posing a significant challenge to achieving the necessary accuracy for practical applications [8]. Existing applications using these methods have shown limited success, often achieving high accuracy in controlled environments but encountering high error rates under diverse and complex conditions [9–11].

To address challenges in pest recognition, such as detecting tiny pests and managing intra-class variation, researchers have utilized deep learning to develop advanced networks.

Among these, the residual networks have been effective, capturing complex features and delivering impressive results. Cheng et al. [2] introduced deep residual learning to reduce training degradation in complex agricultural scenarios. Thenmozhi et al. [12] utilized transfer learning to increase the precision of pre-trained deep learning models. Kong et al. [13] implemented a spatial feature-enhanced attention module that improves the model's capacity to discern semantic relationships between different image regions. Combined with an innovative higher-order pooling module, this led to improved recognition accuracy for various pest categories. Additionally, Wei et al. [14] incorporated a multi-scale feature extraction module (MFE) into their neural network, capturing detailed features of pests at various scales and enhancing the network's capacity to differentiate between diverse pest species. Despite these advances, challenges such as insufficient model generalization and difficulty in capturing fine-grained features still exist. To this end, we propose an adaptive multi-task learning framework that utilizes task synergy and dynamically adjusts learning, enhancing the robustness and accuracy of pest recognition.

Motivated by the above challenges, we propose a multi-task learning framework consisting of a main task and a subsidiary task. The main task employs the discriminative attention multi-network (DAM-Net), which uses a multi-scale learning approach and attention mechanism to identify fine-grained features. The subsidiary task utilizes the residual network-50 (ResNet-50), a deep network with residual connections [15], to supplement texture details and global contextual information. This framework learns features in a multi-dimensional manner, overcoming the limitations of single-task systems and promoting robust feature learning. Furthermore, we implement an adaptive weighted loss mechanism that optimally prioritizes tasks within the framework, mitigating overfitting and enhancing accuracy in pest recognition. All of the necessary details on the proposed multi-task learning framework are further provided in the Methodology section.

As illustrated in Figure 1, our framework processes the input dataset simultaneously through two task pathways. The results from these two tasks are then combined to collaboratively identify pests. Specifically, the DAM-Net has three branches that learn global features, attention-enhanced precise features, and the most discriminative regions. Concurrently, the ResNet-50 improves the main task by providing texture details and global contextual information. The framework enhances its discriminative accuracy by adaptively adjusting the loss weights for each task based on its uncertainty level, prioritizing the task with higher confidence. We evaluate our framework using the D0 dataset, which comprises around 4500 images across 40 pest categories, and the IP102 dataset, which contains over 75,000 images spanning 102 categories. The framework achieves enhanced discernment and increased robustness by integrating DAM-Net's precise attention to detail with ResNet-50's broad feature extraction capabilities. The major contributions of our research are as follows:

- (1) We present the DAM-Net with three branches to enhance the extraction of complex and fine-grained pest features. We also incorporate a subsidiary task based on the ResNet-50, enriching coarse-grained and global contextual information, improving the generalizability and robustness of pest recognition;
- (2) We enhance both learning efficiency and recognition accuracy through a shared representation. Task loss weights, determined by homoscedastic uncertainty, are automatically learned from the data. This approach handles weight initialization reliably and balances task significance;
- (3) An enhanced pest-recognition framework based on multi-task learning achieves accuracies of 99.7% on the D0 dataset and 74.1% on the IP102 dataset.

The remainder of the study is organized as follows: Section 2 reviews the related work. Section 3 introduces the exposition of our multi-task framework. Section 4 details experimental results and corresponding analysis. Section 5 summarizes this study.

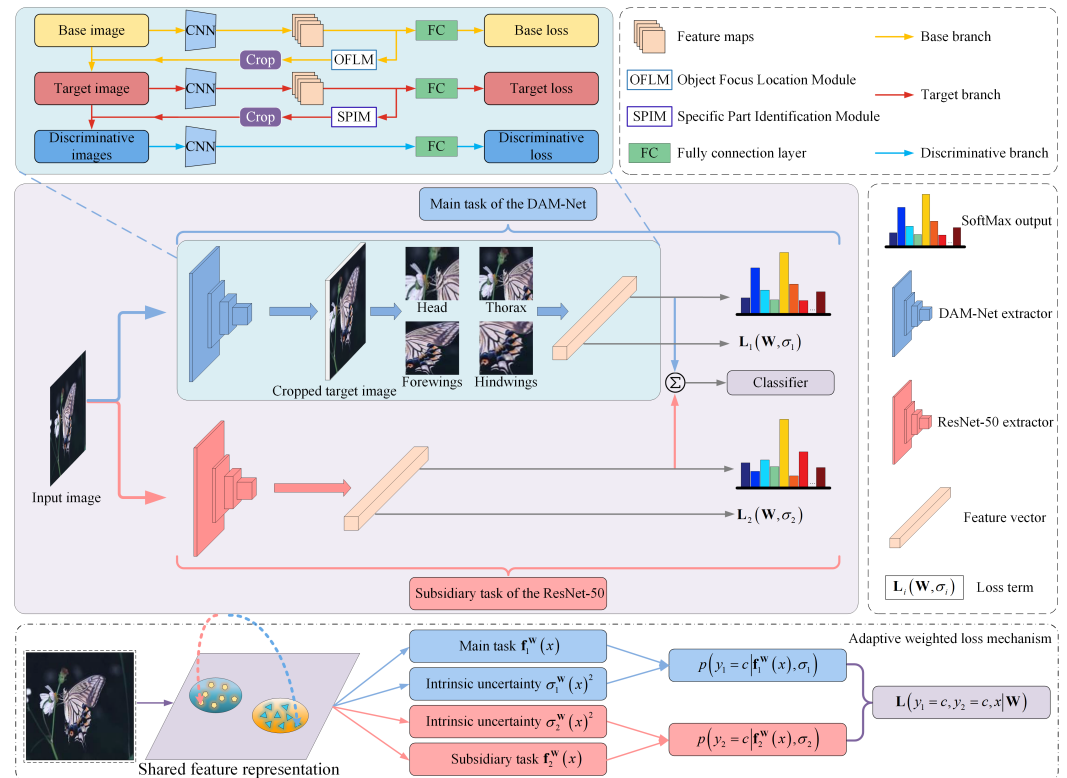


Figure 1. The overview of our proposed multi-task deep learning framework with the discriminative attention multi-network (DAM-Net). In the DAM-Net, the parameters of the convolutional neural networks (CNN) and fully connected (FC) layers in the three branches are shared. The loss function $L_1(W, \sigma_1)$ combines the losses from all three branches, allowing the model to learn both global and fine-grained features. The noise parameter σ_i adjusts the task weights based on data reliability.

2. Related Work

Early pest recognition relied on manual identification by experts, which was labor-intensive, time-consuming, and prone to inaccuracies due to individual and environmental factors [16]. Advances in computer vision have significantly improved efficiency and accuracy in digital and precision agriculture. These methods are categorized into traditional machine learning [10,17–19] and deep learning [2,12,13,20].

Traditional machine learning approaches typically involve extracting features using techniques like the histogram of oriented gradients (HOG) or scale-invariant feature transform (SIFT), followed by classification algorithms such as support vector machines (SVM) or artificial neural networks (ANN). For instance, Xie et al. [17] combined sparse-coding histograms with multiple kernel learning (MKL) for feature fusion. Dimililer et al. [18] used backpropagation neural networks with geometric feature extraction. Xie et al. [19] developed a dictionary-based approach for robust pest recognition. Kasinathan et al. [10] employed multiple feature extraction techniques to enhance classification precision through a majority voting strategy. Despite their effectiveness, traditional methods depend on expert knowledge and manually designed features, which may not capture all variations in pest appearances, leading to unstable outcomes and reduced accuracy.

Deep learning methods, particularly those using CNNs, automatically extract robust features, eliminating the need for manual design and achieving higher recognition efficiency and accuracy [21,22]. Cheng et al. [2] used the ResNet-101 architecture, surpassing traditional methods. Thenmozhi et al. [12] applied transfer learning to improve accuracy and reduce training time. Li et al. [23] fine-tuned GoogLeNet, achieving high accuracy but at a significant computational cost. Bollis et al. [24] proposed weakly supervised learning with multi-instance learning (MIL) for better classification. Kong et al. [13] refined a CSP-based network for precise feature extraction, and Wei et al. [14] introduced innovative

modules for multi-scale and deep feature extraction. Chen et al. [20] used an effective feature localization module (EFLM) for pest location identification and an adaptive filtering fusion module (AFFM) to refine features, with soft voting (SV) for final categorization. However, their method relies heavily on diverse pest images to enrich feature input, making it challenging to extract fine-grained and global context information effectively in real-world applications.

Despite the accuracy and flexibility of deep learning in pest recognition, real-world environmental noise remains a challenge due to factors such as varying lighting conditions, occlusion by foliage, and the presence of similar non-target objects [25,26]. However, applying data augmentation techniques and optimization algorithms can enhance model accuracy and generalization. For instance, Nanni et al. [27] used saliency methods for data augmentation, achieving high accuracies on pest datasets. Cabrera et al. [28] researched generative models for synthesizing pest images. Nanni et al. [29] combine CNN architectures and introduce Adam variants, Exp and ExpLR, enhancing pest recognition. However, this method is complex and lacks fine-grained feature extraction, making it difficult to handle high intra-class variability in pests.

While previous methods have enhanced pest recognition efficiency and accuracy, they struggle to capture fine-grained details and diverse contextual features, reducing accuracy in recognizing different life stages and distinguishing similar species. Therefore, we develop a multi-task learning framework to address these challenges.

3. Methodology

This section details our pest-recognition framework. The DAM-Net, serving as the main task model, excels in fine-grained feature recognition with input images at 448 pixels, learning global features, attention-enhanced features, and discriminative regions. Complementing this, the ResNet-50, as the subsidiary task model, operates on 224-pixel images to extract coarse-grained and global contextual information. By leveraging the diverse capabilities of both models, our framework achieves feature complementarity, enhancing robustness under shared feature extraction. Additionally, it dynamically adjusts loss weights based on task uncertainty, optimizing the integration of fine-grained details and global contextual information. The performance enhancement is demonstrated in the ablation study results shown in Section 4.6.

3.1. Discriminative Attention Multi-Network (DAM-Net)

Our main task is executed using the DAM-Net, designed for fine-grained feature recognition of pests. Figure 2 outlines the composition of the DAM-Net, which includes a base branch, a target branch, and a discriminative branch. The base branch captures global features, providing a holistic image representation. The object focus location module (OFLM) synthesizes feature maps from the base branch to determine the target object's bounding box, enabling cropping of the base image. This cropped image undergoes processing via the target branch to extract structural and fine-grained features. Complementing this, the specific part identification module (SPIM) isolates and crops the most discriminative and least redundant parts of the object. These regions are fed into the discriminative branch, which delves into intricate details across various scales. Equations (1)–(3) represent the loss functions for the three branches, respectively.

$$L_{base} = -\log(P_b(c)) \quad (1)$$

$$L_{target} = -\log(P_a(c)) \quad (2)$$

$$L_{discriminative} = -\sum_{n=0}^{N-1} \log(P_{d(n)}(c)) \quad (3)$$

In this model, c represents the actual category of the input image. The probabilities P_b and P_a are the softmax outputs from the base and target branches, respectively. Meanwhile,

$P_{d(n)}$ is the softmax output for the n -th image in the discriminative branch, with N being the total number of such discriminative images. The total loss is computed as

$$L_{total} = L_{base} + L_{target} + L_{discriminative} \quad (4)$$

The total loss combines the losses from the three branches, utilizing shared convolutional and classification layers to ensure unified feature extraction across the network. During backpropagation, this cumulative loss function adjusts the entire network by considering the losses from each branch, optimizing both global and local feature detection. To enhance efficiency in practical applications, we exclude the discriminative branch during testing, reducing complexity and accelerating predictions. The final results are achieved by merging the logits from the base and target branches.

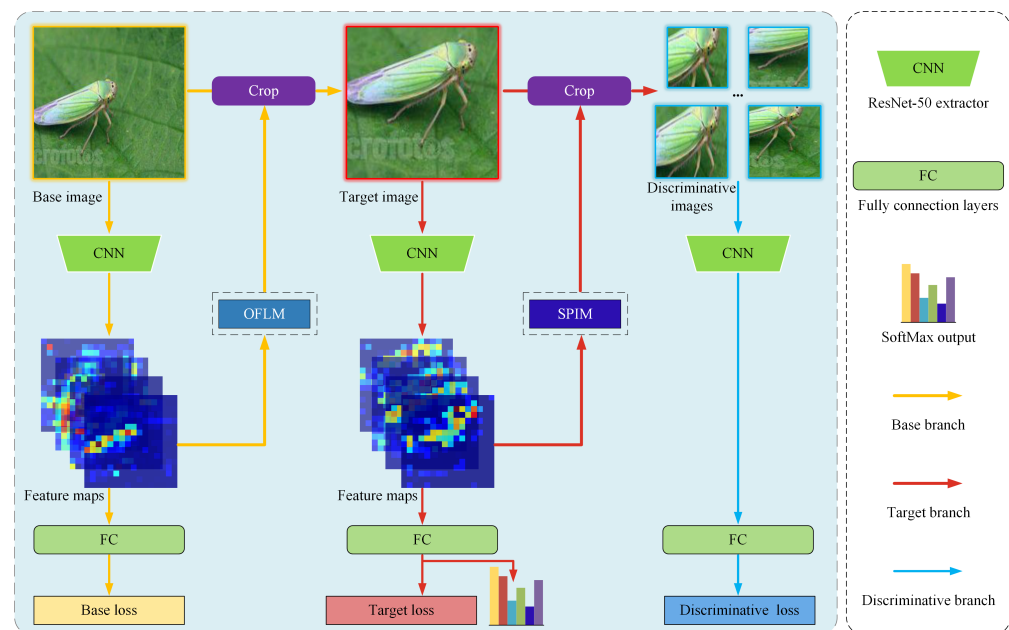


Figure 2. The structure of the DAM-Net.

3.1.1. Object Focus Location Module (OFLM)

The OFLM is designed to precisely delineate the boundary boxes of target pests by focusing on salient features. Feature maps from the final convolutional layer of an input image X are represented as $F \in R^{C \times H \times W}$, where C is the number of channels, and $H \times W$ is the spatial size. The i 'th feature map of the corresponding channel is denoted as f_i . By aggregating these feature maps F , we obtain an activation map A , as defined in Equation (5):

$$A = \sum_{i=0}^{C-1} f_i \quad (5)$$

where the activation map A indicates the focus areas of deep neural networks for recognition and accurately locates target regions, as illustrated in Figure 3. The mean value of A , denoted as $\bar{a}$, is calculated using Equation (6):

$$\bar{a} = \frac{\sum_{x=0}^{W-1} \sum_{y=0}^{H-1} A(x, y)}{H \times W} \quad (6)$$

where $A_{(x,y)}$ represent the pixel intensity value at position (x, y) of the activation map. $\bar{a}$ is used as the threshold to generate a coarse mask map $\tilde{M}_{conv_5c}$ from the final convolutional layer $Conv_5c$ according to Equation (7):

$$\tilde{M}_{(x,y)} = \begin{cases} 1 & \text{if } A_{(x,y)} > \bar{a} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

A bounding box that encloses the largest connected region is used to locate the target. Using the same method, we compute an additional activation map from the preceding convolutional block *Conv_5b* and a corresponding mask $\tilde{M}_{conv_5b}$. We then refine the target mask M_{target} by intersecting $\tilde{M}_{conv_5c}$ and $\tilde{M}_{conv_5b}$, as indicated in Equation (8):

$$M_{target} = \tilde{M}_{conv_5c} \cap \tilde{M}_{conv_5b} \quad (8)$$

A threshold is applied to the final target mask to highlight the most significant regions of interest within the image. A binary mask $\tilde{B}$ is computed using a predefined threshold value τ , such that

$$\tilde{B}_{(x,y)} = \begin{cases} 1 & \text{if } A_{(x,y)} > \tau \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where τ is established as 0.5 for this analysis. The final target mask M_{final} is then obtained using

$$M_{final} = M_{target} \odot \tilde{B} \quad (10)$$

This method treats pixels in the activation map with intensity values above a threshold as significant regions. As a result, the final mask effectively distinguishes between the foreground, which includes features of interest like pests, and the background.

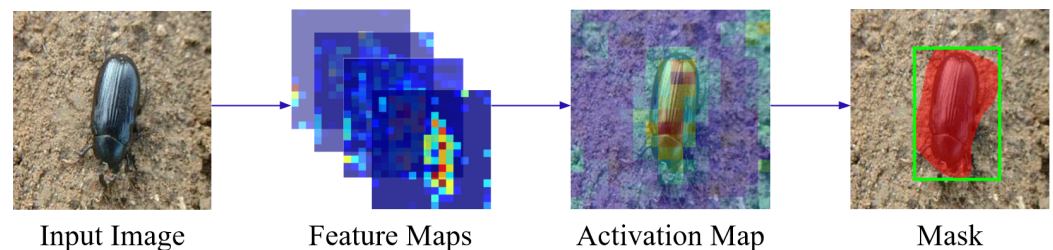


Figure 3. The process of the OFLM.

3.1.2. Specific Part Identification Module (SPIM)

To overcome the challenge of the OFLM typically detecting incomplete sections of a target, we enhance our model's robustness by introducing a SPIM and a discriminative branch. Analysis of the activation map A reveals that regions with higher activation values are often associated with crucial parts of the pest, like its head or tail. To capture these informative areas, we employ a sliding window technique that isolates windows containing critical information as discriminative images. This process is further refined by adopting a full CNN, reducing computational effort in a manner similar to how OverFeat [30] processes feature maps. Subsequently, we perform channel-wise aggregation of the activation maps, A_w , for each window and determined the mean activation value, following the procedure outlined in Equation (11):

$$\bar{a}_w = \frac{\sum_{x=0}^{W_w-1} \sum_{y=0}^{H_w-1} A_w(x,y)}{H_w \times W_w} \quad (11)$$

H_w and W_w represent the height and width of a feature map corresponding to a particular window. We arrange the windows in order based on their $\bar{a}_w$ values, which denote the informativeness of the regions they encompass. A higher $\bar{a}_w$ value signifies greater significance, as shown in Figure 4. To select the most representative windows while avoiding redundancy, we employ the non-maximum suppression (NMS). This technique allows us to choose a diverse set of windows as discriminative images that vary in scale, ensuring more robust and comprehensive localization of target parts.

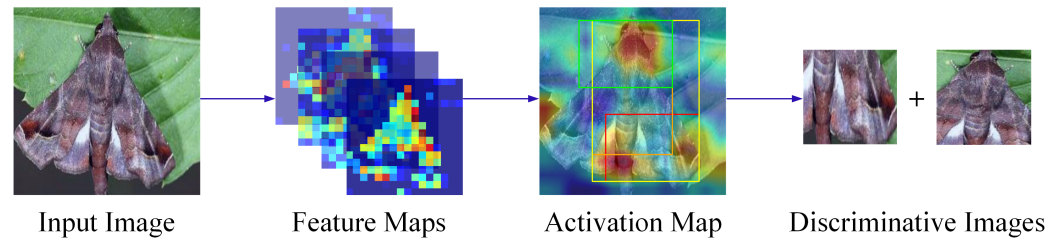


Figure 4. The process of the SPIM. We employ red, orange, yellow, and green to rank average activation values of windows, with red denoting the highest and green the lowest.

3.2. Residual Network-50 (ResNet-50)

The ResNet-50, a deep 50-layer CNN, leverages residual learning to simplify training deeper networks [15]. Figure 5 illustrates a residual block with skip connections that facilitate uninterrupted information flow during forward propagation. This architecture focuses on learning the residual $F(x) = H(x) - x$ between the input x and the output $H(x)$, promoting identity mapping and enabling some layers to be bypassed. During back-propagation, skip connections provide a pathway for gradients to flow directly through the network, migrating the vanishing gradient issue. This architecture allows for the construction of deeper networks to learn more complex features, significantly enhancing deep learning capabilities.

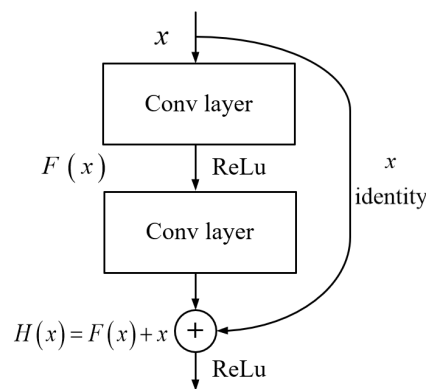


Figure 5. A residual block of a residual network. Within this, x represents the input to the block, $F(x)$ denotes the residual function representing the transformations within the block, and $H(x)$ denotes the target mapping, which is the output of the residual block.

Our ResNet-50 model acts as a subsidiary task, enriching the DAM-Net through a shared representation in a multi-task learning framework. It complements the framework's coarse-grained and global contextual information, enhancing the generalization and robustness of the main task, and ultimately increasing overall recognition accuracy.

3.3. Adaptive Weighted Loss Mechanism

In our multi-task framework for pest recognition, we use the uncertainty of each task as the basis for weighting the various loss functions. Specifically, we utilize homoscedastic uncertainty, a form of aleatoric uncertainty that stays constant for a specific task, regardless of input variations. This allows us to measure a task's inherent noise level or reliability. Our framework utilizes an adaptive weighted loss mechanism, adjusting the contribution of each loss based on its associated homoscedastic uncertainty. This prioritizes tasks with higher confidence, enhancing the overall performance of the framework.

In our pest-recognition task, the neural network's output for an input x is denoted by $\mathbf{f}^{\mathbf{W}}(x)$, where $\mathbf{W}$ represents the network's weights. Then, the likelihood of recognition is adjusted by introducing a parameter σ . This allows for a calibrated confidence in the output, accounting for data uncertainty:

$$p(\mathbf{y}|\mathbf{f}^{\mathbf{W}}(\mathbf{x}), \sigma) = \text{Softmax}\left(\frac{1}{\sigma^2}\mathbf{f}^{\mathbf{W}}(\mathbf{x})\right) \quad (12)$$

In addition, the loss function is defined as the negative logarithm of the conditional probability of the actual class c . This definition aligns with the principles of maximum-likelihood estimation, as illustrated in the following equation:

$$\mathbf{L}(\mathbf{W}, \sigma) = -\log p(\mathbf{y} = c|\mathbf{f}^{\mathbf{W}}(\mathbf{x}), \sigma) \quad (13)$$

Within this model, the log likelihood can be formulated as

$$\log p(\mathbf{y} = c|\mathbf{f}^{\mathbf{W}}(\mathbf{x}), \sigma) = \frac{1}{\sigma^2}f_c^{\mathbf{W}}(\mathbf{x}) - \log \sum_{c'} \exp\left(\frac{1}{\sigma^2}f_{c'}^{\mathbf{W}}(\mathbf{x})\right) \quad (14)$$

where $f_c^{\mathbf{W}}(\mathbf{x})$ is the c 'th element of the output vector $\mathbf{f}^{\mathbf{W}}(\mathbf{x})$.

Finally, by defining $\mathbf{L}_1(\mathbf{W}) = -\log \text{Softmax}(\mathbf{y}_1, \mathbf{f}^{\mathbf{W}}(\mathbf{x}))$, we can derive the following formula:

$$\begin{aligned} \mathbf{L}_1(\mathbf{W}, \sigma_1) &= -\log p(\mathbf{y} = c|\mathbf{f}^{\mathbf{W}}(\mathbf{x})) + \frac{1}{\sigma_1^2}\mathbf{L}_1(\mathbf{W}) - \frac{1}{\sigma_1^2}\mathbf{L}_1(\mathbf{W}) \\ &= \frac{1}{\sigma_1^2}\mathbf{L}_1(\mathbf{W}) + \log \frac{\sum_{c'} \exp\left(\frac{1}{\sigma_1^2}f_{c'}^{\mathbf{W}}(\mathbf{x})\right)}{\left(\sum_{c'} \exp(f_{c'}^{\mathbf{W}}(\mathbf{x}))\right)^{\frac{1}{\sigma_1^2}}} \\ &\approx \frac{1}{\sigma_1^2}\mathbf{L}_1(\mathbf{W}) + \log \sigma_1 \end{aligned} \quad (15)$$

Introducing a simplification for computational tractability, we approximate the softmax normalization constant as $\frac{1}{\sigma_n} \sum_{c'} \exp\left(\frac{1}{\sigma_n^2}f_{c'}^{\mathbf{W}}(\mathbf{x})\right) \approx \left(\sum_{c'} \exp(f_{c'}^{\mathbf{W}}(\mathbf{x}))\right)^{\frac{1}{\sigma_n}}$, which holds exactly as $\sigma_n \rightarrow 1$.

In the multi-task pest recognition framework, we can derive the following loss function:

$$\begin{aligned} \mathbf{L}(\mathbf{W}, \sigma_1, \sigma_2) &= -\log p(\mathbf{y}_1, \mathbf{y}_2 | \mathbf{f}^{\mathbf{W}}(\mathbf{x}), \sigma_1, \sigma_2) \\ &= \frac{1}{\sigma_1^2}\mathbf{L}_1(\mathbf{W}_1) + \frac{1}{\sigma_2^2}\mathbf{L}_2(\mathbf{W}_2) + \log \sigma_1 + \log \sigma_2 \end{aligned} \quad (16)$$

Our objective is to minimize a composite loss function comprising two subtask losses: $\mathbf{L}_1(\mathbf{W})$ for the DAM-Net and $\mathbf{L}_2(\mathbf{W})$ for the ResNet-50. Each loss is modulated by a noise parameter, σ , which adjusts the contribution of each task based on data reliability. Specifically, an increase in the noise parameter σ_1 for $\mathbf{y}_1$ reduces the influence of $\mathbf{L}_1(\mathbf{W})$, thus decreasing reliance on potentially unreliable data; conversely, a decrease in noise increases the weight of the corresponding loss, prioritizing more reliable data. The last term in the function regularizes the noise parameters, preventing them from becoming excessively large and causing the framework to ignore task-specific data.

4. Experiment

4.1. Experimental Setup

Our experiments ran on a platform with an Intel Xeon 8255C CPU (12 vCPUs at 2.50 GHz, Intel Corporation, Santa Clara, CA, USA), an NVIDIA Tesla V100-SXM2 GPU (32 GB VRAM, NVIDIA Corporation, Santa Clara, CA, USA), running Ubuntu 18.04. We used CUDA 11.3 and PyTorch 1.12.1 to develop and test our framework.

4.2. Datasets

In our deep learning experiments, we employ the D0 dataset [19], which comprises around 4500 images across 40 distinct pest categories derived from various field crops, including corn, soybean, wheat, and canola, in authentic agricultural settings. The D0 dataset is available at <https://www.dlearningapp.com/web/DLFautoinsects.htm>, (accessed on 24 June 2024). Moreover, for a comprehensive evaluation of the framework's robustness and generalizability, we utilize a second dataset, IP102 [31], which contains 75,222 images across 102 classes. This dataset presents considerable challenges due to its large intra-class variation and small inter-class variation, encompassing various life stages and exhibiting a pronounced class imbalance, with image counts per class ranging from 71 to 5740. A notable hurdle is the visual similarity between the pests and their backgrounds, which necessitates sophisticated feature extraction for accurate recognition. Figure 6 displays a selection of pest images from IP102, showcasing the visual challenges posed by the varied categories. The IP102 dataset, featuring pest species from diverse crops under varied agricultural conditions, facilitates the evaluation of the framework's adaptability across different contexts. These two benchmark datasets are detailed in Table 1.

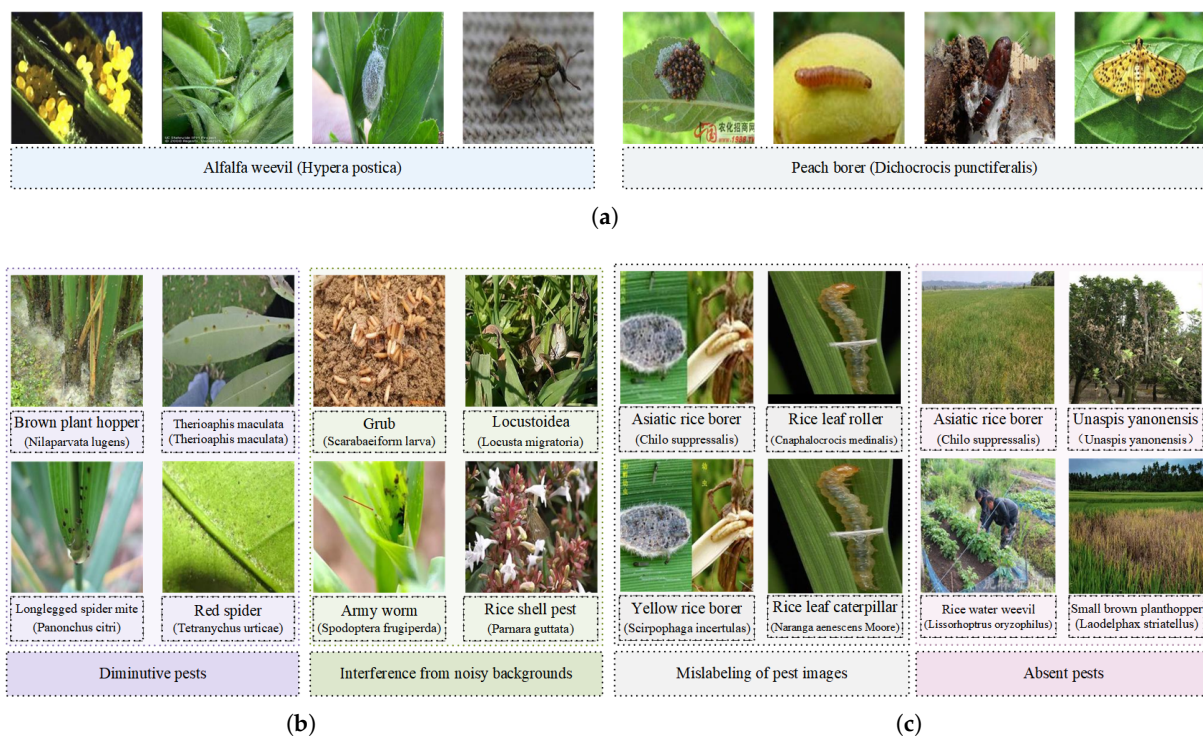


Figure 6. Examples of pest images from the IP102 dataset. All pest names are first shown with their original category names from the dataset, followed by their Latin names in parentheses. (a) Various morphological forms of the same species across different life stages. (b) Images with tiny pests or background interference. (c) Quality issues in the IP102 dataset, including mislabeled and missing pest targets.

Table 1. Details of the two datasets used in this study.

Dataset	Class	Total Count	Training	Validation	Test
D0 [19]	40	4508	3156	405	947
IP102 [31]	102	75,222	45,095	7508	22,619

4.3. Experimental Settings

In our multi-task framework, we propose the DAM-Net as the main task model and the ResNet-50 as the subsidiary model. During training, images for the DAM-Net are

resized to 480 pixels and randomly cropped to 448 pixels, while images for the ResNet-50 are resized to 256 pixels and cropped to 224 pixels. For testing, center cropping is used for both tasks. We intentionally avoid additional data augmentation like rotation and color jittering to focus on assessing the framework’s inherent performance.

Optimizing hyperparameters is essential for deep learning classification tasks. For the DAM-Net, we employ an SGD optimizer starting with a learning rate of 0.001, reduced by a factor of 0.1 at the 15th and 30th epochs through a step scheduler. For the ResNet-50, we utilize an Adam optimizer with an initial learning rate of 0.0001, adjusted each epoch using an exponential decay scheduler. Both tasks incorporate L2 regularization to prevent overfitting. Training is halted by an early stopping mechanism if validation accuracy does not improve for 25 consecutive epochs, optimizing both generalization and resource efficiency. Detailed hyperparameter settings are provided in Table 2. These adjustments were carefully calibrated to enhance pest-recognition accuracy.

Table 2. General training settings for the multi-task framework.

Task/ Hyper-Parameters	Main Task (DAM-Net)	Subsidiary Task (ResNet-50)
Batch size	12	12
Optimizer	SGD <i>momentum</i> = 0.9	Adam <i>betas</i> = (0.9, 0.999)
Learning rate	0.001	0.0001
Scheduler	MultiStep decay rate = 0.1	Exponential decay rate = 0.96
Weight decay	0.0001	0.0001
Dropout	0	0.5
Image size	448 × 448	224 × 224
Loss	Categorical cross-entropy	
Maximum epochs	100	

4.4. Evaluation Metrics

To comprehensively assess our framework’s performance, particularly in addressing class imbalances in the D0 and IP102 datasets, we adopt the same evaluation metrics used in existing research [29,32–34]. Specifically, we utilize five reliable evaluation metrics: accuracy (Acc), macro-average precision (MPre), macro-average recall (MRec), macro-average F1-score (MF1), and geometric mean (GM). The Acc provides an overall performance measure, while the macro-average metrics offer insights into class-wise performance, and the GM balances sensitivity and specificity. These metrics ensure a robust and comparable evaluation of our framework’s effectiveness.

4.5. Experimental Results and Analysis

To evaluate our proposed method, we conduct comparative experiments against established benchmarks on the D0 and IP102 datasets. Table 3 presents the comparison results. For the D0 dataset, the multi-level classification framework with unsupervised learning and a multiple kernel boosting algorithm [19] achieved the lowest Acc at 89.3%. Enhanced pre-trained models using transfer learning [12] reached an Acc of 96.0%. Furthermore, a two-stream network combining ConvNeXt and Swin Transformer models [32] achieved an Acc of 98.5%. The GAEnsemble method [33], which uses a genetic algorithm-based weighted voting mechanism, achieved an Acc of 98.8%. However, it struggles to capture fine-grained pest features due to its lack of detailed feature extraction. In contrast, our framework achieves an Acc of 99.7%, and has a shorter inference time of 74 ms compared to their 106 ms, demonstrating superior efficiency and performance.

Regarding the IP102 dataset, the ResNet-50 model [31] had the lowest performance. The DMF-ResNet model [35] integrated multi-scale representation learning from three branches and incorporated an SFR module to refine channel-wise feature responses, achiev-

ing an Acc of 59.1%. A feature fusion network synthesizing CNN and Vision Transformer (ViT) models [36] achieved an Acc of 65.6%. An enhanced pre-trained Inception-v3 model [37] achieved an Acc of 67.9%. Additionally, the saliency-guided discriminative learning network (SGDL-Net) [38], utilizing a multi-task learning framework with a dual-branch architecture for global and fine-grained feature extraction, achieved an Acc of 71.2% and an MF1 of 60.6%. However, our multi-task framework, integrating the DAM-Net for the main task and the ResNet-50 for the subsidiary task, outperforms the SGDL-Net with Acc and MF1 improvements of 2.9% and 2.0%, respectively. A model using a genetic algorithm-based hyperparameter optimization strategy [34] achieved an Acc of 71.8%. An ensemble method [29] combined multiple CNN architectures and variant Adam optimization algorithms, achieving an Acc of 73.6%. A model [20] using an EFLM and an AFFM coupled with soft voting achieved an MF1 of 73.6%. Despite using attention mechanisms, it struggles with diverse contextual features and varying pest appearances, achieving an Acc of 73.9%, lower than our 74.1%. In contrast, our framework uses multi-task learning and an adaptive weighted loss mechanism, capturing a broader range of features and adapting better to real-world conditions for superior performance.

Table 3. Comparison of our multi-task framework with previous methods. (“-” indicates that the original study did not provide the corresponding metric result. The bold highlights indicate the best results. F1 scores reported in the original studies are assumed to be the macro-average F1 scores.)

Dataset	Method	Inference Time (ms)	Acc (%)	MPre (%)	MRec (%)	MF1 (%)	GM (%)
D0	MLLF+MKB [19]	-	89.3	-	-	-	-
	CNNs+Transfer learning [12]	-	96.0	-	-	-	-
	FNSTC [32]	-	98.5	-	98.2	98.4	-
	GAEnsemble [33]	106	98.8	98.9	98.8	98.8	-
	Our method	74	99.7	99.5	99.7	99.6	99.5
IP102	ResNet-50 [31]	-	49.4	-	40.1	-	-
	DMF-ResNet [35]	-	59.1	-	58.1	-	-
	ResNet152+ViT+Swin-T [36]	-	65.6	60.9	59.7	60.3	-
	Pre-trained Inception-v3 [37]	-	67.9	-	-	60.6	-
	SGDL-Net [38]	-	71.2	-	-	63.9	-
	IRNV2 [34]	-	71.8	65.9	63.2	64.1	-
	DGrad-Adam+CNNs+EM [29]	-	73.6	-	-	72.9	84.7
	EFLM+AFFM+SV [20]	-	73.9	-	-	73.6	-
	Our method	74	74.1	64.0	69.9	65.9	59.5

Our multi-task framework outperformed previous methods in several key metrics. On the D0 dataset, our method achieved the highest performance across all metrics, highlighting its effectiveness in capturing both fine-grained details and diverse contextual information. On the IP102 dataset, our framework achieved an Acc of 74.1% and an MRec of 69.9%, surpassing existing methods. The inference time was 74 ms, further demonstrating the framework’s efficiency.

To further assess our framework’s effectiveness, Figure 7 presents a visual analysis of the accuracy and loss curves on the D0 and IP102 datasets. This comparison provides a comprehensive overview of the performance across these distinct datasets. During training, our framework shows a consistent improvement in accuracy for both datasets. Initially, both training and validation losses decrease, indicating effective learning. Over time, while the validation loss exhibits a minor increase, the validation accuracy continues to rise slightly. This suggests that the framework maintains its generalization ability.

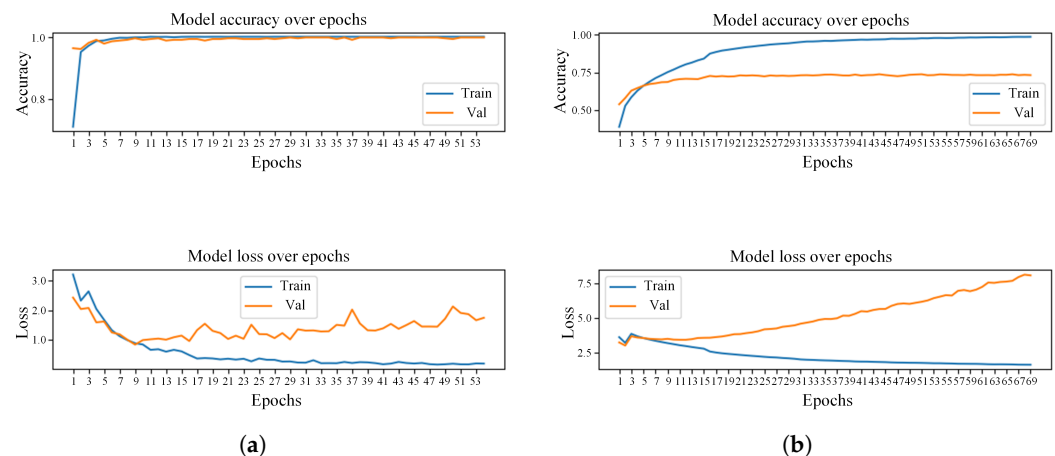


Figure 7. Training vs. validation accuracy and loss curves of the proposed framework. (a) Curves on D0 dataset. (b) Curves on IP102 dataset.

4.6. Ablation Study

In this section, we perform ablation studies to analyze the effects of different components in our proposed framework for pest recognition.

To assess the performance of the DAM-Net when isolated from the ResNet-50 model, we conduct experiments on the D0 and IP102 datasets. The DAM-Net achieves an Acc of 99.6%, an MF1 score of 99.5%, and a GM of 99.5% on the D0 dataset. On the IP102 dataset, the results are an Acc of 71.4%, an MF1 of 63.7%, and a GM of 58.0%. These results underscore the discriminative power and robustness of the DAM-Net, confirming its suitability as the main task. The DAM-Net excels in learning global, attention-enhanced, and distinctive regional features, facilitating the recognition of subtle details and discriminative features for classification.

We also train the ResNet-50 independently to evaluate its standalone performance. On the D0 dataset, the ResNet-50 achieves an Acc of 99.3%, an MF1 of 99.1%, and a GM of 99.1%. On the IP102 dataset, it achieves an Acc of 70.9%, an MF1 of 63.5%, and a GM of 59.5%. These results highlight the competent classification performance of the ResNet-50. By providing extensive global contextual information and an enriched feature representation, the ResNet-50 complements the main task, enhancing the overall performance within the multi-task learning framework.

In our multi-task learning framework, we integrate the DAM-Net as the main task and the ResNet-50 as the subsidiary task, using an adaptive weighted loss mechanism. As shown in Table 4, this configuration improves performance compared to each model operating independently. On the D0 dataset, Acc increases to 99.7%, MF1 to 99.6%, and GM to 99.5%. On the IP102 dataset, Acc improves to 74.1%, MF1 to 65.9%, and GM to 59.5%. These enhancements confirm that combining the DAM-Net and the ResNet-50 leverages complementary features, boosting classification performance and enhancing generalization through task knowledge sharing. The adaptive weighted loss mechanism adjusts the importance of each task based on uncertainties. This approach benefits complex real-world scenarios by focusing on the most confident and informative features and reducing the impact of noisy data. It optimizes the integration of fine-grained details and global contextual information, thereby enhancing overall performance.

The ablation study validates the critical role of each component within our multi-task learning framework and demonstrates the superiority of the multi-task learning approach over single-task models. The results emphasize the pivotal role of the adaptive weighted loss mechanism in achieving further enhancements in multi-task recognition efforts.

Table 4. Ablation study results for the multi-task learning framework on D0 and IP102 datasets. The bold highlights indicate the best results.

	# D0			# IP102		
	Acc (%)	MF1 (%)	GM (%)	Acc (%)	MF1 (%)	GM (%)
DAM-Net	99.6	99.5	99.5	71.4	63.7	58.0
ResNet-50	99.3	99.1	99.1	70.9	63.5	59.5
Multi-task framework	99.7	99.6	99.5	74.1	65.9	59.5

4.7. Qualitative Analysis

Our framework demonstrates superior performance on the D0 dataset, but faces challenges on the IP102 dataset due to the varied appearances of pests at different growth stages. This variability complicates classification, as species within the same genus can exhibit significant differences across stages, and unrelated species can appear similar at specific stages. To evaluate the framework’s robustness, Figure 8 displays instances of misclassifications, highlighting difficulties in pest recognition during the egg, larval, and pupal stages. These stages often lack distinct morphological features, leading to a homogenized appearance that confuses the framework. In contrast, adult pests develop unique features that aid correct recognition. However, the framework’s performance is also impacted by issues such as mislabeled categories and absent targets in the IP102 dataset images [20,39], as shown in Figure 6c.

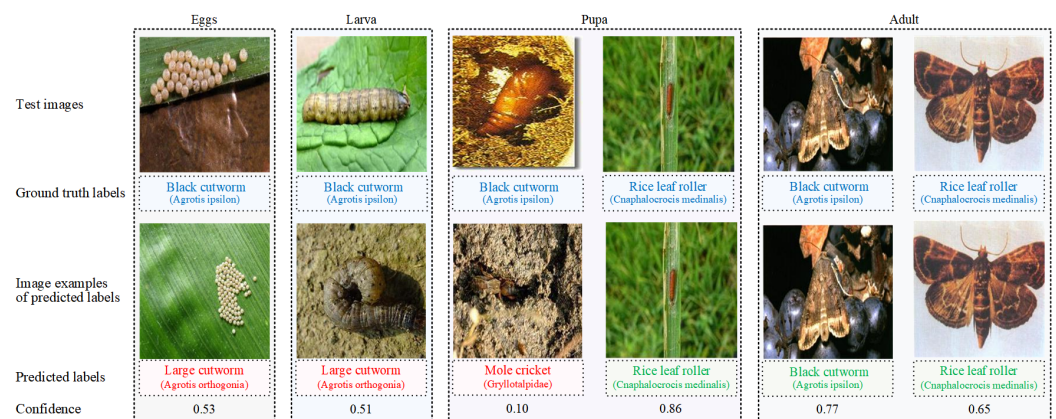


Figure 8. Instances of misclassifications by our multi-task framework on the IP102 dataset. Predictions that are correct are marked in green, while incorrect predictions are highlighted in red. All pest names are shown first with their original dataset category names, followed by their Latin names in parentheses.

To further illustrate the strengths and limitations of our framework, Table 5 compares our multi-task framework with several models on the IP102 dataset. MobileNetV3, ideal for resource-constrained environments due to its small number of parameters and low FLOPs, performs poorly in terms of Acc and MF1. ConvNeXt base outperforms standard CNNs like GoogLeNet and ResNet-50, showing superior overall performance. While the Swin Transformer base achieves high performance, its complex hierarchical architecture results in 88 million parameters and 15.7 GFLOPs. Additionally, it lacks specialized fine-grained feature recognition to effectively handle variations in pest appearances. In contrast, our multi-task learning framework integrates the DAM-Net and the ResNet-50 with an adaptive weighted loss mechanism, enhancing feature extraction and reducing overfitting. This approach achieves superior accuracy of 74.1% with only 49 million parameters and 20.3 GFLOPs, demonstrating the effectiveness and efficiency of our framework.

Our method incurs higher FLOPs due to processing 448×448 inputs, compared to the 224×224 used by other methods. However, this increased computational load allows for capturing finer details and richer contextual information, enhancing pest recognition

accuracy. Our framework is designed for cloud server deployment, where computational resources are less constrained. Although network transmission delays exist, they increase tolerance for computational demands. Thus, this study focuses on achieving superior pest recognition performance rather than minimizing computational cost.

Table 5. The comparative analysis of parameters and performance of various methods on the IP102. Parameters are in million (M) and FLOPs are in gigaflop (G). Note that within our multi-task learning framework, the DAM-Net processes inputs of size 448×448 , while other methods use 224×224 . This results in higher FLOPs for our method due to the increased input dimensions. The bold highlights indicate the best results.

Method	Parameters (M)	FLOPs (G)	Acc (%)	MRec (%)	MF1 (%)
MobileNetV3 [40]	5.4	0.3	63.4	58.6	59.7
GoogLeNet [41]	23	1.8	67.8	58.7	57.9
ResNet-50 [15]	25	4.1	70.9	64.3	63.5
Our method	49	20.3	74.1	69.9	65.9
ViT small [42]	49	9.3	65.5	57.7	61.4
EfficientNet b7 [43]	66	4.9	66.3	60.4	64.2
DenseNet-201 [27]	69	12.2	61.9	56.4	59.2
ConvNeXt base [44]	89	8.2	71.4	67.5	65.8
Swin Transformer base [45]	88	15.7	73.2	69.1	69.9

5. Conclusion and Future Work

We develop a multi-task framework for enhanced pest recognition, leveraging the strengths of the DAM-Net and the ResNet-50 for the main and subsidiary tasks, respectively. We confirm that the DAM-Net effectively identifies fine-grained features, with the ResNet-50 enhancing this by enriching texture details and global context, thus improving the framework's generalizability. The framework incorporates an adaptive weighted loss mechanism to dynamically adjust loss weights based on task uncertainty, aiming to improve overall accuracy. Tests on the D0 and IP102 datasets prove the robustness of our method, reaching accuracies of 99.7% and 74.1%, respectively, and outperforming existing methods. Despite the framework's efficacy in pest recognition, it demonstrates limitations when dealing with imbalanced datasets. To address this, it is crucial to collect a comprehensive collection of high-quality photos of insects at various stages of development. Future work will leverage diffusion models to augment scarce categories by synthesizing high-fidelity images, thus enhancing the framework's generalizability across diverse pest scenarios and improving its performance in pest control and crop protection amid data scarcity.

Author Contributions: Conceptualization, Z.D. and Z.Z.; methodology, Z.D., X.W.; software, Z.D., Y.W.; validation, X.W., Y.W.; formal analysis, Z.D.; investigation, Z.D.; data curation, Y.W.; writing—original draft, Z.D., X.W.; writing—review and editing, Z.Z. and J.G.; visualization, Z.D.; supervision, Z.Z. and J.G.; project administration, Z.Z. and J.G.; funding acquisition, Z.Z. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Litchi and Longan Industry Technology System of China (CARS-32-11), Key Laboratory of Modern Agricultural Intelligent Equipment in South China, Ministry of Agriculture and Rural Affairs, China (HNZJ202209), Special Fund for the Rural Revitalization Strategy of Guangdong (2023TS-3, 2024TS-3) and Guangzhou Key Research and Development Project (2023B03J1363).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Two public datasets including D0 and IP102 can be obtained from the links below: D0: <https://www.dlearningapp.com/web/DLFautoinsects.htm>, accessed on 24 June 2024; IP102: <https://github.com/xpwu95/IP102>, accessed on 24 June 2024.

Acknowledgments: We express our gratitude for the constructive guidance and valuable suggestions offered by the review experts and the editor.

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

References

1. Crowder, D.W.; Northfield, T.D.; Strand, M.R.; Snyder, W.E. Organic agriculture promotes evenness and natural pest control. *Nature* **2010**, *466*, 109–112. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Cheng, X.; Zhang, Y.; Chen, Y.; Wu, Y.; Yue, Y. Pest identification via deep residual learning in complex background. *Comput. Electron. Agric.* **2017**, *141*, 351–356. [\[CrossRef\]](#)
3. Shafi, U.; Mumtaz, R.; García-Nieto, J.; Hassan, S.A.; Zaidi, S.A.R.; Iqbal, N. Precision agriculture techniques and practices: From considerations to applications. *Sensors* **2019**, *19*, 3796. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Al-Hiary, H.; Bani-Ahmad, S.; Reyalat, M.; Braik, M.; Alrahamneh, Z. Fast and accurate detection and classification of plant diseases. *Int. J. Comput. Appl.* **2011**, *17*, 31–38. [\[CrossRef\]](#)
5. Larios, N.; Deng, H.; Zhang, W.; Sarpola, M.; Yuen, J.; Paasch, R.; Moldenke, A.; Lytle, D.A.; Correa, S.R.; Mortensen, E.N.; et al. Automated insect identification through concatenated histograms of local appearance features: Feature vector generation and region detection for deformable objects. *Mach. Vis. Appl.* **2008**, *19*, 105–123. [\[CrossRef\]](#)
6. Solis-Sánchez, L.O.; Castañeda-Miranda, R.; García-Escalante, J.J.; Torres-Pacheco, I.; Guevara-González, R.G.; Castañeda-Miranda, C.L.; Alaniz-Lumbreras, P.D. Scale invariant feature approach for insect monitoring. *Comput. Electron. Agric.* **2011**, *75*, 92–99. [\[CrossRef\]](#)
7. Wen, C.; Guyer, D. Image-based orchard insect automated identification and classification method. *Comput. Electron. Agric.* **2012**, *89*, 110–115. [\[CrossRef\]](#)
8. 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. [\[CrossRef\]](#)
9. Rani, R.U.; Amsini, P. Pest identification in leaf images using SVM classifier. *Int. J. Comput. Intell. Inform.* **2016**, *6*, 248–260.
10. Kasinathan, T.; Uyyala, S.R. Machine learning ensemble with image processing for pest identification and classification in field crops. *Neural Comput. Appl.* **2021**, *33*, 7491–7504. [\[CrossRef\]](#)
11. Mekha, J.; Parthasarathy, V. An automated pest identification and classification in crops using artificial intelligence—A state-of-art-review. *Autom. Control Comput. Sci.* **2022**, *56*, 283–290. [\[CrossRef\]](#)
12. Thenmozhi, K.; Reddy, U.S. Crop pest classification based on deep convolutional neural network and transfer learning. *Comput. Electron. Agric.* **2019**, *164*, 104906. [\[CrossRef\]](#)
13. Kong, J.; Wang, H.; Yang, C.; Jin, X.; Zuo, M.; Zhang, X. A spatial feature-enhanced attention neural network with high-order pooling representation for application in pest and disease recognition. *Agriculture* **2022**, *12*, 500. [\[CrossRef\]](#)
14. Wei, D.; Chen, J.; Luo, T.; Long, T.; Wang, H. Classification of crop pests based on multi-scale feature fusion. *Comput. Electron. Agric.* **2022**, *194*, 106736. [\[CrossRef\]](#)
15. 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.
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [\[CrossRef\]](#)
17. Xie, C.; Zhang, J.; Li, R.; Li, J.; Hong, P.; Xia, J.; Chen, P. Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning. *Comput. Electron. Agric.* **2015**, *119*, 123–132. [\[CrossRef\]](#)
18. Dimililer, K.; Zarrouk, S. ICSP: Intelligent classification system of pest insects based on image processing and neural arbitration. *Appl. Eng. Agric.* **2017**, *33*, 453. [\[CrossRef\]](#)
19. Xie, C.; Wang, R.; Zhang, J.; Chen, P.; Dong, W.; Li, R.; Chen, T.; Chen, H. Multi-level learning features for automatic classification of field crop pests. *Comput. Electron. Agric.* **2018**, *152*, 233–241. [\[CrossRef\]](#)
20. Chen, Y.; Chen, M.; Guo, M.; Wang, J.; Zheng, N. Pest recognition based on multi-image feature localization and adaptive filtering fusion. *Front. Plant Sci.* **2023**, *14*, 1282212. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Júnior, T.D.C.; Rieder, R. Automatic identification of insects from digital images: A survey. *Comput. Electron. Agric.* **2020**, *178*, 105784. [\[CrossRef\]](#)
22. 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. *Ecol. Inform.* **2021**, *66*, 101460. [\[CrossRef\]](#)
23. Li, Y.; Wang, H.; Dang, L.M.; Sadeghi-Niaraki, A.; Moon, H. Crop pest recognition in natural scenes using convolutional neural networks. *Comput. Electron. Agric.* **2020**, *169*, 105174. [\[CrossRef\]](#)
24. Bollis, E.; Pedrini, H.; Avila, S. Weakly supervised learning guided by activation mapping applied to a novel citrus pest benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 70–71.
25. Yang, J.; Lan, G.; Li, Y.; Gong, Y.; Zhang, Z.; Ercisli, S. Data quality assessment and analysis for pest identification in smart agriculture. *Comput. Electr. Eng.* **2022**, *103*, 108322. [\[CrossRef\]](#)

26. Wang, K.; Chen, K.; Du, H.; Liu, S.; Xu, J.; Zhao, J.; Chen, H.; Liu, Y.; Liu, Y. New image dataset and new negative sample judgment method for crop pest recognition based on deep learning models. *Ecol. Inform.* **2022**, *69*, 101620. [\[CrossRef\]](#)
27. Nanni, L.; Maguolo, G.; Pancino, F. Insect pest image detection and recognition based on bio-inspired methods. *Ecol. Inform.* **2020**, *57*, 101089. [\[CrossRef\]](#)
28. Cabrera, J.; Villanueva, E. Investigating generative neural-network models for building pest insect detectors in sticky trap images for the Peruvian horticulture. In Proceedings of the Annual International Conference on Information Management and Big Data, Virtual Event, 1–3 December 2021; pp. 356–369.
29. Nanni, L.; Manfè, A.; Maguolo, G.; Lumini, A.; Brahnam, S. High performing ensemble of convolutional neural networks for insect pest image detection. *Ecol. Inform.* **2022**, *67*, 101515. [\[CrossRef\]](#)
30. Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv* **2013**, arXiv:1312.6229.
31. Wu, X.; Zhan, C.; Lai, Y.K.; Cheng, M.M.; Yang, J. Ip102: A large-scale benchmark dataset for insect pest recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8787–8796.
32. Wang, C.; Zhang, J.; He, J.; Luo, W.; Yuan, X.; Gu, L. A two-stream network with complementary feature fusion for pest image classification. *Eng. Appl. Artif. Intell.* **2023**, *124*, 106563. [\[CrossRef\]](#)
33. Ayan, E.; Erbay, H.; Varçın, F. Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks. *Comput. Electron. Agric.* **2020**, *179*, 105809. [\[CrossRef\]](#)
34. Ayan, E. Genetic algorithm-based hyperparameter optimization for convolutional neural networks in the classification of crop pests. *Arab. J. Sci. Eng.* **2024**, *49*, 3079–3093. [\[CrossRef\]](#)
35. Liu, W.; Wu, G.; Ren, F. Deep multibranch fusion residual network for insect pest recognition. *IEEE Trans. Cogn. Dev. Syst.* **2020**, *13*, 705–716. [\[CrossRef\]](#)
36. An, J.; Du, Y.; Hong, P.; Zhang, L.; Weng, X. Insect recognition based on complementary features from multiple views. *Sci. Rep.* **2023**, *13*, 2966. [\[CrossRef\]](#) [\[PubMed\]](#)
37. Coulibaly, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D. Explainable deep convolutional neural networks for insect pest recognition. *J. Clean. Prod.* **2022**, *371*, 133638. [\[CrossRef\]](#)
38. Luo, Q.; Wan, L.; Tian, L.; Li, Z. Saliency guided discriminative learning for insect pest recognition. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
39. Peng, H.; Xu, H.; Gao, Z.; Zhou, Z.; Tian, X.; Deng, Q.; He, H.; Xian, C. Crop pest image classification based on improved densely connected convolutional network. *Front. Plant Sci.* **2023**, *14*, 1133060. [\[CrossRef\]](#) [\[PubMed\]](#)
40. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
41. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
42. 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.
43. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning; PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
44. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
45. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.

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.