

Optimizing Token Fusion Mechanisms in Swin Transformer for Improved Feature Representation and Fine-Grained Insect Classification

Qiming Li

*Faculty of Engineering, The University of New South Wales, Sydney, Australia
qiming.li1@student.unsw.edu.au*

Abstract: Effective pest monitoring is critical for ensuring global food security, as insect pests pose significant threats to crop yields and agricultural sustainability. Traditional pest identification methods rely on manual inspection, which is time-consuming, labor-intensive, and susceptible to human error. Deep learning, particularly Convolutional Neural Networks (CNNs), has been widely applied to automate insect classification; however, these models exhibit limitations in capturing long-range dependencies and hierarchical feature representations. Transformer-based architectures, such as Swin Transformer, have demonstrated improved performance in vision tasks, yet their effectiveness in fine-grained insect classification remains constrained by cross-window communication inefficiencies and suboptimal hierarchical feature aggregation. This study proposes an enhanced Swin Transformer model incorporating a Token Fusion mechanism to address these limitations. The proposed Cross-Window Token Fusion (CWTF) facilitates feature propagation across non-overlapping windows, while Hierarchical Token Fusion (HTF) enhances multi-scale representation learning. Experimental evaluations on the IP102 dataset, a large-scale benchmark for insect classification, demonstrate that the proposed model outperforms CNN-based and standard Transformer-based models in classification accuracy while maintaining computational efficiency. The findings highlight Token Fusion's effectiveness in strengthening feature interactions, thereby advancing the development of highly accurate and computationally efficient automated pest monitoring systems. This research contributes to the broader adoption of Transformer-based architectures in precision agriculture, enhancing real-time pest surveillance and sustainable crop management.

Keywords: IP102, Token Fusion, Swin Transformer, Long-Range Dependencies

1. Introduction

Ensuring global food security requires efficient and accurate monitoring of agricultural pests, which can significantly affect crop yields and economic stability. Traditional pest monitoring methods rely heavily on manual identification by agricultural experts, which is time-consuming, labor-intensive, and prone to human error [1]. The advent of deep learning and computer vision offers an opportunity to automate insect pest classification with high accuracy. However, real-world challenges such as class imbalance, subtle interspecies morphological variations, and lifecycle differences hinder the

widespread application of high-precision models. These challenges necessitate the development of more advanced models capable of fine-grained feature extraction and contextual learning.

Previous research in this domain has primarily focused on Convolutional Neural Networks (CNN)-based approaches for insect classification. Wu et al. (2019) introduced the IP102 dataset, the largest benchmark dataset for pest recognition, and evaluated various CNN-based models, highlighting their potential for automated pest identification [1]. While effective, these methods face challenges in capturing long-range dependencies and hierarchical representations, which are essential for fine-grained classification [2]. While CNNs such as MobileNetV2 have demonstrated strong performance in pest classification, Setiawan et al. struggle with long-range dependencies and hierarchical feature extraction, motivating the exploration of Transformer-based approaches [3]. Recently, the emergence of Transformer-based models has demonstrated significant improvements in computer vision tasks by capturing global contextual information [4]. Vision Transformers (ViTs) and their hierarchical extensions, such as Swin Transformers, have shown superior performance in multiple domains [2]. Nevertheless, standard Swin Transformers face limitations in cross-window communication and hierarchical feature aggregation, which can hinder their performance in complex visual recognition tasks involving small, closely related categories [5].

Recent advances in Vision Transformers have explored various token reduction techniques to enhance computational efficiency. Kim et al. (2023) proposed Token Fusion (ToFu), which dynamically integrates token pruning and token merging to optimize model performance while maintaining computational efficiency [6]. This paper proposes an enhanced Swin Transformer architecture with a Token Fusion mechanism to address these limitations. The proposed mechanism consists of two components: Cross-Window Token Fusion (CWTF), which improves token interaction across window boundaries, and Hierarchical Token Fusion (HTF), which facilitates multi-scale feature integration across different stages of the network. These enhancements aim to strengthen feature propagation and improve classification accuracy in fine-grained insect recognition. The remainder of this paper is organized as follows. Section 2 describes the IP102 dataset and presents the detailed architecture of the proposed model. Section 3 reports the experimental results and analyzes the effectiveness of the Token Fusion mechanism compared to baseline models. Section 4 concludes with a summary of the findings and discusses potential future directions for improving Transformer-based insect classification systems.

2. Data and Methods

The dataset used in this study is IP102, a large-scale benchmark dataset for insect pest recognition. IP102 consists of 75,222 high-quality images covering 102 categories of common crop pests. The images were mainly collected from mainstream internet search engines and specialized agricultural websites, supplemented by frame images extracted from videos. To enhance data diversity, the dataset integrates images from multiple sources, leading to variations in image quality, distribution, and quantity [1].

IP102 employs a hierarchical classification system, categorizing pests into two super-classes and eight sub-classes. The dataset is partitioned into training, validation, and test sets in a 6:1:3 ratio, containing 45,095, 7,508, and 22,619 images, respectively. The number of images per category follows a natural long-tailed distribution, reflecting real-world pest population imbalances. This distribution increases the complexity of classification tasks and serves as a valuable benchmark for studying data imbalance learning. Additionally, IP102 includes images of pests at different life cycle stages—such as eggs, larvae, pupae, and adults—introducing further challenges, as their appearances can vary significantly across stages.

2.1. Methodology

2.1.1. ResNet-152

ResNet-152 is a deep neural network with 152 layers, designed to address the gradient vanishing problem in deep learning. Its core mechanism, residual connections (or skip connections), enables the network to learn the residual between input and output directly. This approach effectively mitigates gradient vanishing and prevents performance degradation as network depth increases [7,8].

The architecture of ResNet-152 consists of multiple stacked residual modules, each following a Bottleneck design that includes a 1×1 convolution for dimensionality reduction, a 3×3 convolution for feature extraction, and another 1×1 convolution for dimensionality expansion. The network begins with a 7×7 convolutional layer, followed by four stages of residual blocks, a global average pooling layer, and a fully connected layer for classification. Compared to ResNet-50, ResNet-152 has a significantly deeper structure and stronger feature extraction capabilities, enabling higher accuracy in large-scale image classification tasks.

2.1.2. YOLO11

YOLO11 is an advanced object detection model that achieves high accuracy and efficiency in real-time applications. Based on previous versions of YOLO (You Only Look Once), it introduces several improvements to enhance detection precision, speed, and adaptability in complex environments.

The main innovations of YOLO11 include an improved backbone network, an enhanced feature pyramid, and an adaptive anchor-free mechanism. The backbone adopts a more efficient convolutional architecture, which improves feature extraction while maintaining low computational cost. The Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) have been further optimized to enhance multi-scale detection, especially for small and densely packed objects. In addition, the adaptive anchor-free mechanism removes the dependence on predefined anchor boxes, making the model more flexible for objects of different sizes and aspect ratios [9].

Compared with previous versions, YOLO11 achieves higher detection accuracy and faster inference speed. It is particularly suitable for real-time applications such as autonomous driving, security surveillance, and agricultural pest detection. With a good balance between computational efficiency and detection performance, YOLO11 provides a reliable solution for modern object detection tasks.

2.1.3. Swin Transformer

Shifted Window Transformer (Swin Transformer) is an innovative network architecture designed for computer vision tasks. It addresses the high computational complexity and low global feature extraction efficiency of traditional ViTs by incorporating Shifted Window Attention and a hierarchical structure. This design enables Swin Transformer to efficiently model long-range dependencies while maintaining computational feasibility [2].

Similar to CNN, Swin Transformer employs a hierarchical structure to generate feature maps with progressively lower resolutions, making it well-suited for tasks such as image classification. The Shifted Window Attention mechanism enables self-attention computation within non-overlapping local windows, reducing computational complexity compared to global attention [2]. However, although this approach allows for efficient token interaction within windows, it restricts communication between tokens located in separate windows, thereby limiting the model's capacity to capture long-range dependencies.

To address this limitation, the Swin Transformer introduces Shifted Window Attention, where the window partitioning is shifted between successive layers to enable cross-window connections. This

design allows tokens at window boundaries to interact with those in neighboring windows, effectively expanding the receptive field across layers and improving contextual modeling without incurring significant computational cost. As a result, Swin Transformer achieves better feature extraction for fine-grained classification tasks, such as insect classification. The network primarily consists of a Patch Embedding module, multiple Swin Transformer Blocks, and a feature downsampling module. Compared to ViTs, Swin Transformer demonstrates superior performance and efficiency, making it a strong candidate for large-scale vision applications [2].

Formally, let $X \in R^{N \times D}$ denote the input token sequence, where N is the number of tokens and D is the feature dimension. The query (Q_W), key (K_W) and value (V_W) matrices are computed by applying learned linear projections to X , and are further partitioned into non-overlapping windows of size $M \times M$ (i.e., M^2 patches per window). Self-attention is computed independently within each window W according to the following equation:

$$A_W = \text{Softmax} \left(\frac{Q_W K_W^T}{\sqrt{d}} + B_W \right) V_W \quad (1)$$

where $B_W \in R^{M^2 \times M^2}$ denotes the relative positional bias specific to the window. This mechanism ensures efficient token communication at window boundaries, enhancing the model's ability to capture both local and non-local dependencies across hierarchical levels.

2.1.4. Token Fusion

2.1.4.1. Cross-Window Token Fusion

Although Shifted Window Attention improves token communication at window boundaries, it does not fully address the challenge of aggregating information across non-overlapping windows within the same layer. To address this limitation, this work proposes CWFT, which enhances cross-window interactions by aggregating adjacent token embeddings before self-attention computation [9]. This ensures more comprehensive feature propagation while maintaining computational efficiency.

Unlike the standard Swin Transformer, which primarily relies on self-attention within fixed local windows, CWFT employs depthwise separable convolution and channel attention to enable cross-window feature aggregation. The CWFT module consists of two key components:

Depthwise Separable Convolution Module: A 3×3 depthwise convolution is applied to capture local spatial dependencies, followed by a 1×1 pointwise convolution to fuse information across channels. This design ensures efficient and effective cross-window feature propagation with minimal computational overhead. The transformation is formally defined as:

$$X' = \text{ReLU}(\text{Conv}_{1 \times 1}^{PW}(\text{ReLU}(\text{Conv}_{3 \times 3}^{DW}(X)))) \quad (2)$$

Where $\text{Conv}_{3 \times 3}^{DW}$ represents a depthwise convolution that processes each channel independently, while $\text{Conv}_{1 \times 1}^{PW}$ performs cross-channel fusion.

Channel Attention Module: To enhance adaptive feature selection, a global average pooling (GAP) operation extracts a compact global feature representation, which is then passed through a two-layer MLP-based attention mechanism. The computed channel-wise attention weights are applied to the transformed feature map, allowing adaptive emphasis on the most relevant channels. The attention mechanism is expressed as:

$$A = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot \text{GAP}(X')) + b_1) + b_2 \quad (3)$$

$$X_f = X' \cdot A \quad (4)$$

Where $\text{GAP}(X')$ computes a spatial average over each channel, $W_1 \in \mathbb{R}^{\frac{C}{4} \times C}$ and $W_2 \in \mathbb{R}^{C \times \frac{C}{4}}$ are learnable weight matrices in the two-layer MLP, b_1 and b_2 are bias terms, and σ denotes the Sigmoid function, ensuring that the attention weights remain within a normalized range.

2.1.4.2. Hierarchical Token Fusion

While Cross-Window Token Fusion focuses on enhancing local token interactions across windows within the same layer, HTF aims to integrate multi-scale contextual information by aggregating features extracted at different receptive field sizes. This process ensures the retention of both fine-grained local details and broader global structures [4].

In Swin Transformer, each stage progressively reduces the spatial resolution of token embeddings through downsampling. However, this can result in information loss, especially for fine-scale structures. To address this issue, HTF incorporates a multi-scale convolutional feature extraction mechanism combined with spatial attention, enabling effective hierarchical feature fusion.

HTF consists of three main components:

Local Feature Extraction: A 3×3 depthwise separable convolution captures fine-grained local details while maintaining computational efficiency.

$$X_{local} = \text{Conv}_{3 \times 3}^{DW}(X) \quad (5)$$

Global Feature Extraction: A 5×5 depthwise separable convolution captures larger spatial dependencies, enriching the representation with broader contextual information.

$$X_{global} = \text{Conv}_{5 \times 5}^{DW}(X) \quad (6)$$

Feature Fusion and Spatial Attention: The outputs from the 3×3 and 5×5 convolutions are concatenated along the channel dimension and passed through a 1×1 convolutional bottleneck, reducing redundancy and integrating complementary information. Finally, a 7×7 spatial attention mechanism learns an adaptive weighting map to refine the fused features by emphasizing informative spatial regions. The fusion operation is expressed as:

$$X_{fusion} = \text{Conv}_{1 \times 1}([X_{local}, X_{global}]) \quad (7)$$

$$A = \sigma(\text{Conv}_{7 \times 7}(X_{fusion})) \quad (8)$$

$$X_f = X_{fusion} \cdot A \quad (9)$$

Where $\text{Conv}_{5 \times 5}^{DW}(X)$ and $\text{Conv}_{3 \times 3}^{DW}(X)$ represent depthwise convolutions for local and global feature extraction, respectively, and $\text{Conv}_{1 \times 1}$ reforms pointwise feature fusion. The final spatial attention is computed using a 7×7 convolution, ensuring effective selection of important spatial features.

2.2. Model Training and Optimization

The models in this study are trained and evaluated using the IP102 Insect Classification Dataset. The dataset is split into training (60%), validation (10%), and testing (30%) sets. Standard augmentation techniques, including resizing images to 224×224 , normalization based on ImageNet statistics, and random horizontal flipping, are applied during training to enhance generalization [1]. To handle class imbalance, a class-balanced weighted cross-entropy loss is utilized, where each class is assigned a weight inversely proportional to its frequency in the dataset. The loss function is defined as:

$$L = - \sum_{i=1}^C w_i y_i \log(y_i) \quad (10)$$

where w_i represents the inverse frequency of class i , ensuring that rare classes contribute more significantly to the optimization process. The optimization strategy follows the AdamW optimizer with a batch size of 128 and a weight decay of $1e-4$. For better performance, the learning rate is set to 0.001 for ResNet152 and YOLO11, while Swin Transformer uses a learning rate of 0.0001. To mitigate overfitting, early stopping with a patience of five epochs is employed.

Model performance is evaluated based on several key metrics, including Top-1 accuracy to assess classification correctness, F1-score to balance precision and recall, and AUC (Area Under the ROC Curve) to measure overall classification confidence. The primary objective of this evaluation is to analyze the efficacy of Token Fusion in improving Swin Transformer performance. Comparisons are made against ResNet-152, a CNN-based deep architecture; YOLO11, a Transformer-based object detection model adapted for classification; and the baseline Swin Transformer without Token Fusion. The analysis encompasses classification accuracy, computational efficiency, and the effectiveness of feature representation across different model architectures. Results indicate that Swin Transformer with Token Fusion achieves a superior balance of accuracy and efficiency, outperforming baseline models while maintaining reasonable computational complexity. These findings validate the role of Token Fusion in enhancing token interactions and improving feature extraction in fine-grained insect classification tasks.

3. Result Analysis

To evaluate the effectiveness of different deep learning models in fine-grained insect classification, we compare Swin Transformer, ResNet-152, and YOLO11 under identical training and evaluation settings. The models are assessed using Top-1 accuracy, F1-score, AUC, and loss, while inference time and GFLOPs are analyzed to quantify computational efficiency. Table 1 presents the classification performance, highlighting the trade-offs between accuracy, model complexity, and inference speed.

Among the three models, Swin Transformer achieves the highest classification accuracy of 71.630%, significantly outperforming ResNet-152 (61.75%) and YOLO11 (60.66%). Similarly, Swin Transformer demonstrates superior robustness in terms of F1-score (71.35%) and AUC (0.9853), indicating better ability to distinguish fine-grained insect classes compared to ResNet-152 (F1-score: 60.62%, AUC: 0.9749) and YOLO11 (F1-score: 60.01%, AUC: 0.9526). Loss values further validate these observations. Swin Transformer achieves the lowest loss (1.28), signifying more stable convergence and improved optimization, whereas ResNet-152 (1.52 loss) and YOLO11 (4.12 loss) exhibit higher error rates. The notably high loss of YOLO11 suggests difficulties in effectively learning fine-grained insect features, which aligns with its lower accuracy and F1-score.

From a computational perspective, YOLO11 has the fastest inference time (5.81ms) due to its lightweight design, making it favorable for real-time applications. However, its lower classification accuracy and higher loss indicate trade-offs in predictive performance. In contrast, Swin Transformer requires slightly more inference time (9.36ms) but achieves significantly better classification results while maintaining computational efficiency. Notably, Swin Transformer has the lowest GFLOPs (3.12), substantially lower than ResNet-152 (11.53 GFLOPs), making it a computationally efficient alternative.

Table 1: Classification Performance of Different Baseline Models

Model	Top-1 Accuracy	Loss	F1-score	AUC	Inference Time	GFLOPs
ResNet-152	61.75%	1.52	60.62%	0.9749	8.60ms	11.53
YOLO11	60.66%	4.12	60.87%	0.9526	5.81ms	6.78
Swin Transformer	71.63%	1.27	71.35%	0.9853	9.36ms	3.12

The results indicate that Token Fusion improves Swin Transformer by addressing the token interaction limitations in window-based attention. Compared to the baseline Swin Transformer, both CWTF and HTF show improvements in classification performance. HTF achieves an accuracy of 72.49%, representing a 0.86% increase over the baseline. CWTF achieves 71.52% accuracy, which remains comparable. Both approaches introduce a small increase in computational cost, with CWTF adding 0.03 GFLOPs and HTF adding 1.04 GFLOPs. The inference time shows minor variations, increasing by 0.05ms and 0.29ms, respectively. The AUC of Hierarchical Token Fusion reaches 0.9872, compared to 0.9853 for the baseline, suggesting that the method contributes to distinguishing fine-grained insect species.

Table 2: Ablation Study on Token Fusion Mechanisms

Model	Top-1 Accuracy	Loss	F1-score	AUC	Inference Time	GFLOPs
Swin Transformer (Baseline)	71.63%	1.27	71.35%	0.9853	9.36ms	3.12
+ Cross-Window Token Fusion	71.52%	1.35	71.15%	0.9852	9.41ms	3.15
+ Hierarchical Token Fusion	72.49%	1.14	72.02%	0.9872	9.65ms	4.16
+ Both	71.83%	1.34	70.51%	0.9857	9.71ms	4.22

To analyze the effects of CWTF and HTF, an ablation study is conducted. The results in Table 2 suggest that both mechanisms contribute to classification accuracy and feature learning. The CWTF facilitates token exchange between adjacent windows, which may reduce local feature constraints. However, its accuracy (71.52%) is slightly lower than the baseline (71.63%), which may be attributed to a higher loss value (1.35 vs. 1.28). In contrast, the HTF achieves 72.49% accuracy, while also reducing loss to 1.14, a decrease of 0.14 compared to the baseline. Recent studies have explored token fusion strategies to enhance the efficiency of ViTs. Lee et al. (2024) proposed Multi-Criteria Token Fusion (MCTF), which considers similarity, informativeness, and token size to improve feature aggregation. The next approach extends this idea by integrating CWTF and HTF to enhance both local and global feature propagation [10]. When both fusion mechanisms are combined, the accuracy reaches 71.83%, indicating a balance between their individual effects. The combined model maintains an AUC of 0.9857, supporting its effectiveness in fine-grained classification.

The results further highlight the impact of Token Fusion on model efficiency and scalability. While both CWTF and HTF contribute to classification performance, their effects on computational cost and inference latency differ. CWTF primarily enhances local feature propagation, leading to only a marginal increase in GFLOPs (3.15 vs. 3.12) and inference time (9.41 ms vs. 9.36 ms) compared to the baseline Swin Transformer. This suggests that local token interactions can be improved with minimal additional computational burden. On the other hand, the HTF strengthens global token integration across hierarchical levels, leading to a more pronounced accuracy gain (72.49%) but also a noticeable increase in GFLOPs (4.16) and inference time (9.65 ms). This indicates that improving global feature flow requires additional computations, yet it provides greater benefits in fine-grained classification scenarios.

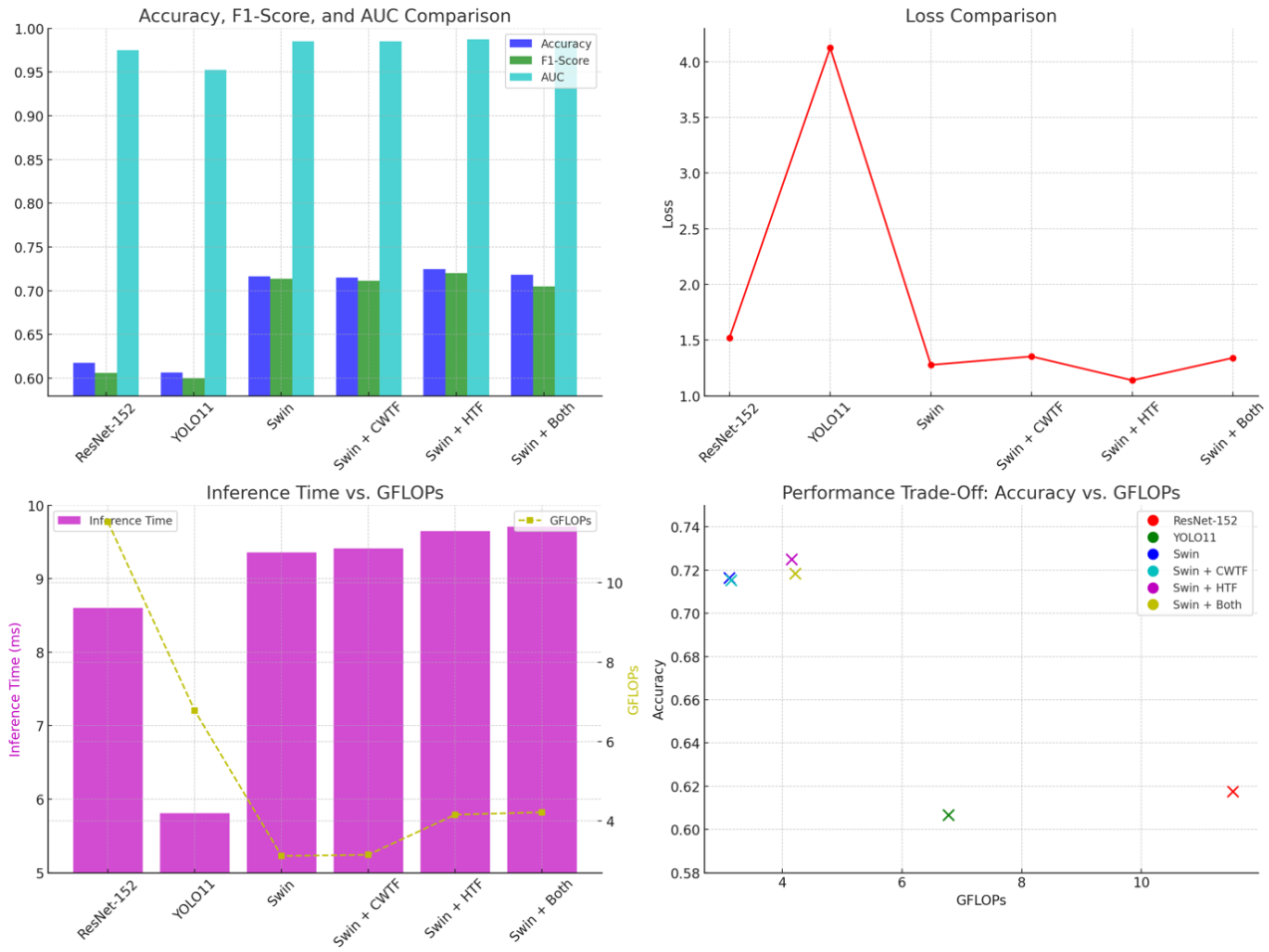


Figure 1: Performance and Efficiency Trade-Offs in Fine-Grained Insect Classification.

From Figure 1, it is evident that there exists a trade-off between computational complexity and classification performance. The bottom-right scatter plot (Performance Trade-Off: Accuracy vs. GFLOPs) demonstrates that Swin Transformer maintains a strong balance between efficiency and accuracy, achieving high classification performance with significantly lower computational cost compared to ResNet-152. Meanwhile, YOLO11, despite its efficiency, suffers from significantly lower accuracy, making it less suitable for fine-grained insect classification. The introduction of Token Fusion slightly increases GFLOPs but leads to noticeable accuracy improvements, particularly for HTF. The bottom-left bar chart (Inference Time vs. GFLOPs) further confirms this trend. Swin Transformer and its Token Fusion variants achieve superior classification accuracy with relatively low inference times compared to ResNet-152. While HTF and Swin + Both require slightly higher computational resources, their accuracy gains justify the trade-off in applications where fine-grained classification is essential. Notably, the CWTF incurs only a minimal increase in inference time while preserving computational efficiency, reinforcing its viability for real-time or low-latency scenarios.

Interestingly, the combined approach (Swin + Both) does not surpass the HTF in terms of accuracy (71.83% vs. 72.49%), despite integrating both token fusion mechanisms. This suggests that global token interactions play a more critical role in fine-grained feature extraction than localized feature refinement alone. However, the combined method balances computational cost and performance, making it a viable option for applications requiring both efficiency and classification precision.

From an efficiency perspective, Token Fusion introduces a scalable approach to balancing accuracy and computational cost. The trade-off between accuracy gains and computational overhead

suggests that different Token Fusion strategies can be selected based on the target application. For real-time insect monitoring or edge-based deployment, CWTF might be preferable due to its minimal impact on inference latency. Meanwhile, for high-accuracy scenarios such as automated pest identification in controlled environments, HTF provides superior classification performance at a reasonable computational cost.

These findings demonstrate that Token Fusion not only improves classification accuracy but also introduces a scalable framework where different token interaction strategies can be tailored to application-specific constraints. Future work could explore dynamic or adaptive token fusion mechanisms that adjust token interactions based on image complexity, further optimizing the trade-off between accuracy and computational efficiency.

4. Conclusion

This study investigates Transformer-based architectures for fine-grained insect classification and proposes an enhanced Swin Transformer model integrating Token Fusion mechanisms to address limitations in cross-window communication and hierarchical feature aggregation. The CWTF enhances feature propagation across window boundaries, mitigating local feature isolation, while the HTF facilitates multi-scale feature integration across hierarchical stages. The IP102 benchmark dataset is used to evaluate the proposed approach, demonstrating that the enhanced Swin Transformer outperforms CNN-based and standard Transformer models in classification accuracy and robustness. The experimental results indicate that the proposed model achieves a Top-1 accuracy of 72.49%, outperforming the standard Swin Transformer and CNN-based baselines. The ablation study confirms the effectiveness of Token Fusion, with CWTF improving intra-window interactions and HTF refining hierarchical feature representations. Despite a marginal increase in computational complexity, the proposed approach achieves a favorable balance between accuracy and efficiency, making it a viable solution for real-world agricultural pest monitoring.

Future research could explore adaptive token fusion strategies that dynamically adjust token interactions based on input complexity, further optimizing the trade-off between computational cost and classification accuracy. Additionally, integrating self-supervised learning could enhance feature extraction for species with limited labeled data. Extending the dataset to include more diverse insect species, environmental conditions, and life cycle variations could further improve model generalization. This study contributes to the deployment of Transformer-based models in automated pest monitoring systems, supporting advancements in precision agriculture, ecological conservation, and global food security.

References

- [1] Wu, X., Zhan, C., Lai, Y.K., Cheng, M.M. and Yang, J. (2019) IP102: A large-scale benchmark dataset for insect pest recognition. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8787-8796. Available at: https://openaccess.thecvf.com/content_CVPR_2019/papers/Wu_IP102_A_Large-Scale_Benchmark_Dataset_for_Insect_Pest_Recognition_CVPR_2019_paper.pdf
- [2] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. (2021) Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012-10022. Available at: <https://arxiv.org/abs/2103.14030>
- [3] Setiawan, A., Yudistira, N. and Wihandika, R.C. (2022) Large-scale pest classification using efficient convolutional neural network with augmentation and regularizers. *Computers and Electronics in Agriculture*, 200. Available at: <https://www.sciencedirect.com/science/article/pii/S0168169922005191>
- [4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017) Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30.
- [5] Yoo, D., Kim, J. and Yoo, J. (2024) FSwin Transformer: Feature-Space Window Attention Vision Transformer for Image Classification. *IEEE Access*, 12, pp. 72598-72606. Available at: <https://ieeexplore.ieee.org/abstract/document/10509828#citations>

- [6] Kim, M., Gao, S., Hsu, Y.-C., Shen, Y. and Jin, H. (2024) *Token Fusion: Bridging the Gap between Token Pruning and Token Merging*. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 372-382. Available at: <https://arxiv.org/abs/2312.01026>
- [7] He, K., Zhang, X., Ren, S. and Sun, J. (2016) *Deep residual learning for image recognition*. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778. Available at: <https://arxiv.org/abs/1512.03385>
- [8] Khanam, R. and Hussain, M. (2024) *YOLOv11: An Overview of the Key Architectural Enhancements*. Available at: <https://doi.org/10.48550/arXiv.2410.17725>
- [9] Ma, J., Tang, L., Fan, F., Huang, J., Mei, X. and Ma, Y. (2022) *SwinFusion: Cross-domain long-range learning for general image fusion via Swin Transformer*. *IEEE/CAA Journal of Automatica Sinica*, 9, pp. 1200-1217. doi: 10.1109/JAS.2022.105686
- [10] Lee, S., Choi, J. and Kim, H.J. (2024) *Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 741-751. Available at: <https://arxiv.org/abs/2403.10030>.