National College of Ireland

Project Submission Sheet

Student Name:	Nachiket Mehendale		
Student ID:	x23272473		
Programme:	Master of Science in Artificial Intelligence	**Year:**	Sep 2024
Module:	MSCAI1 Research Practicum Part 2		
Lecturer:	Prof. Ade Fajemisin		
Submission Due Date:	11/08/2025		
Project Title:	Enhanced Crop Weed Detection Using an Optimized Swin Transformer Architecture for Precision Agriculture		
Word Count:	2922		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
<u>ALL</u> internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Nachiket Mehendale

Date: 07/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhanced Crop Weed Detection Using an Optimized Swin Transformer Architecture for Precision Agriculture

Mehendale, Nachiket Anil
National College of Ireland, D01 K6W2, Ireland

ABSTRACT

Under the framework of precision agriculture, machine learning technology enables better crop handling, increased agricultural output, and reduced harm to the environment (Liakos et al., 2018). A key use of this technology is to accurately tell the difference between crops and weeds on farms, so machines can remove weeds automatically and reduce the need for chemical weed killers (Mortensen et al., 2016). Current machine learning approaches for this task include Convolutional Neural Networks (CNNs), Vision Transformers, and Swin Transformers (Dosovitskiy et al., 2021). Among these, Swin Transformers work better for crop-weed detection because they process both detailed local features and broader image context in a hierarchical manner (Liu et al., 2021). However, Swin Transformers require significant computing power, which limits their use on farms with basic hardware. Our research introduced an optimized version of the Swin Transformer by applying three innovative optimization strategies and comparing four distinct model architectures: CNN, Vision Transformer, baseline Swin Transformer, and the proposed optimized Swin Transformer for crop-weed identification performance. The optimized Swin Transformer achieved respectable accuracy alongside the other three models. To evaluate the model's effectiveness across varied farming conditions, the optimized Swin Transformer was retrained on a new and distinct set of crops and weed images. The evaluation showed that the model can adapt well to unfamiliar agricultural contexts and maintain good performance when tested on multiple crop species and farming settings. The primary goal of this project is to minimize the computational cost associated with Swin Transformers without compromising their accuracy and effectiveness through Dynamic Token Clustering (DTC), Adaptive Patch Splitting (APS), and Selective Cross-Scale Attention (SCSA). The optimized architecture demonstrated significant gains across multiple computational metrics while keeping accuracy like that of the baseline model. The lowered processing demands allow for feasible deployment on low-resource agricultural devices like mobile platforms and edge systems.

Keywords: Precision Agriculture, Crop-Weed Classification, Swin Transformer Optimization, Computational Efficiency, Transfer Learning

1. INTRODUCTION

Weeds are types of plants that grow in areas that are not desirable and can cause crops to struggle to grow well (Zimdahl, 2013). They compete with main crops for space, sunlight, and water. Manually removing weeds can take time and effort, which is another reason image-based weed detection technique is growing in popularity among farmers (Dyrmann et al., 2016). Farms have large acreage, and manually determining areas of weed development would take considerable time and effort relative to image matching algorithms and computer driven models. Image detection models can show farms a good opportunity for weed identification and critical thresholding based on crop density (Kamilaris and Prenafeta-Boldú, 2018). The model we implement is based on a Swin Transformer- a type of vision transformer that evaluates images in "windows" and learns patterns of data (Liu et al., 2021). Although the Swin Transformer performs well in a variety of tasks, it is still slow, complex and memory intensive (Dosovitskiy et al., 2021). Therefore, we are exploring optimizing the Swin Transformer to enhance speed while still producing results that are significantly useful in detecting weeds in crops.

Past studies have applied Swin transformers in agriculture for disease detection, crop classification, and weed detection (Chen et al., 2022). These studies have shown that Swin transformers resulted in impressive performance; however, the models often still had room for improvement in speed and efficiency. Our main hypothesis is that we can improve the Swin transformer by using careful interventions that do not sacrifice accuracy. The three introduced optimization methods are: 1: Dynamic Window Size Adaptation in which the model will use window types of many sizes depending on the complexity of the image. 2: Hierarchical Sparse Token Pruning which removes less useful parts of the image but maintains useful parts of the image. 3: Cross-Layer Weight Sharing with Adaptive Modulation in which parts of the model share weight to reduce size within the model while still modal to the needs of each layer.

We trained two models: one standard Swin Transformer and one optimized version. We used publicly available crop-weed dataset on Kaggle and applied image-augmentation techniques to improve the flexibility of the model and reduce overfitting (Shorten and Khoshgoftaar, 2019). We tested both models using accuracy and other metrics such as model size, memory usage, and inference time to evaluate resource optimization. Additionally, we evaluated our optimized model against CNN and Vision Transformer architectures to demonstrate its comparative advantages across different deep learning approaches (Khan et al., 2020). We also conducted transfer learning experiments using the optimized Swin Transformer on an independent agricultural dataset to assess the model's generalization capability and adaptability to different crop-weed scenarios with varying environmental conditions.

Our optimized model performed less computation and at the same time provided good accuracy. This research can help farmers find weeds quickly. It can reduce the use of chemicals and help crops grow better. We believe that it shall make farming easier and more lasting.

2. RELATED WORK

The physical systems which are used during processing of applications can be termed as computing substrate (Wang et al., 2023). This approach improved the recognition of weeds and crops with similar visual characteristics by utilizing hierarchical feature extraction and applying contrastive loss to further improve feature differences between categories. The method achieved high accuracy on both private and public datasets in comparison to Convolutional Neural Network. But it requires heavy memory resources, which could be a challenge for real-time applications. This study discusses the potential of advanced transformer architectures but also highlights their limitations in terms of computational complexity.

A new transformer-based model for identifying weeds in maize farms was introduced (Jiang et al., 2023). It performed better than models like DeepLabv3 and PSANet and scored 96.52% mPA and 93.71% mIoU. The model used crop masks from segmentation to recognize maize and weeds more accurately thereby reducing the need for labeling tasks. It was faster than CNN and its variants but still too slow for real-time detection because it required a lot of memory resources.

Maize crops are crucial for food security. Accurately identifying areas with maize disease is a key challenge in modern farming. This is particularly because corn leaf diseases can appear in many areas and in different shapes, making them difficult to detect. A combined transformer-based model for segmenting maize leaf disease regions was presented (Yang et al., 2024). The model uses a SENet module and a hybrid loss function to help the model learn better. The model produces satisfactory results and much better results than other CNN-based models like U-Net with VGG or ResNet50 and DeepLabV3+ with MobileNet or Xception. However, researchers agree that the model should need fewer parameters to work faster and more efficiently.

In one study, three different transformer models were evaluated for weed segmentation (Jiang et al., 2022). The hierarchical transformer achieved a mean Intersection over Union (mIoU) of 65.41% but required around 29 million trainable parameters. In contrast, SegFormer had a mIoU result of 65.74% but took only 3.7 million parameters. The larger number of parameters will require more computational power, which may affect effectiveness for real-time applications in precision agricultural field.

Pests and diseases affect crops and cause big financial losses. They make it hard to grow enough food. It is not easy to detect these pests, primarily because pests come in varied shapes and sizes. For detecting pests, there are not enough labeled images available for training deep learning based image-detection systems. To fix this, a two-step method was developed (Li et al., 2024). The approach integrated Cascade RCNN and transformer models. Initially, the pest image dataset was enhanced using augmentation techniques. The transformer helped the model learn basic features from the images. Then, a special layer called SCF-FPN was inserted to help the model spot accurate pest regions. It uses a sliding window that adjusts to each pest's shape. A feature pyramid helps the model understand both local and global features. Soft NMS and extra steps from Cascade RCNN were also used. This step made the model detect pests more clearly. The model was evaluated on 28 pest types and scored well: 92.5% accuracy, 91.8% recall, and 93.7% mean average precision. These are better than older methods, showing this model can prove good for real world farming solutions. However, there are issues. The model struggles with rare pests and also needs more inference time and disk space.

Early weed detection helps reduce the usage of herbicides and avoid crop loss. In deep learning, transformers could effectively perform weed detection. CNNs primarily concentrate on textures, whereas transformers are better at identifying forms. This is helpful since weeds may be identified by their distinct forms. In one experiment, 2 types of transformers were compared with a well-known CNN variant-EfficientNet-v2 (Espejo-Garcia et al., 2020). Researchers used weight transfer from ImageNet and applied data augmentation techniques from AutoAugment. These methods helped solve the issue of small size agricultural datasets. The advanced transformer achieved 98.51% accuracy on the DeepWeeds dataset. After optimizing the model, it scored 98.61% accuracy. The Softmax layer was removed, and Support Vector Machines (SVM) and Gradient Boosting were used to improve performance. Grad-CAM++ was used to highlight the areas the model focused on while identifying weeds. It proved that the model paid attention to the plants (local features) and not the background (global features). Transformers worked better than CNNs and were more accurate in identifying different types of weeds. This shows that transformers have strong potential to improve weed detection. Researchers emphasize the necessity of testing on other datasets to make sure the model functions in various scenarios.

Food image classification is an important task in computer vision, used for things like assessing diets and recognizing food on social media. Past research has looked at how well convolutional neural networks (CNNs) and transformers work for this task. For example, CNN models like MobileNet are commonly used for food recognition, but they struggle with complex visual relationships. The Vision Transformer (ViT) is another model that has shown promise in capturing long-range dependencies on images, but it also needs a lot of data and computing power. In contrast, recent studies have shown that hierarchical transformers are better than CNNs and ViT for food classification tasks. The hierarchical structure and moving window mechanism help handle complex relationships in food images more efficiently. In a study using the Food-11 dataset, the transformer performed better than CNN-based models and ViT in terms of accuracy (Bae et al., 2024). However, this study only used one dataset, which raises questions about whether these results apply more broadly. This research aims to test transformers on different food image datasets and see how they can be combined with other advanced models to improve performance. The study shows the potential of transformers for complex food classification tasks and sets the stage for future research to expand their use.

3. RESEARCH METHODOLOGY

3.1 SWIN TRANSFORMER ARCHITECTURE

The Swin Transformer extends the transformer architecture from natural language processing to computer vision by treating images as sequences of visual patches (Liu et al., 2021). The architecture initiates processing by segmenting the input image into non-overlapping patches of size 4×4 pixels. Each patch is subsequently transformed into a feature vector, resulting in a sequence of visual tokens. The model uses 4 stage hierarchy where each stage works at a different image resolution. It starts with small patches and then gradually merges them to cover larger areas in the image. This allows the model to learn from both fine details and the overall structure of the image. The core innovation is the shifted window attention mechanism (Liu et al., 2021).

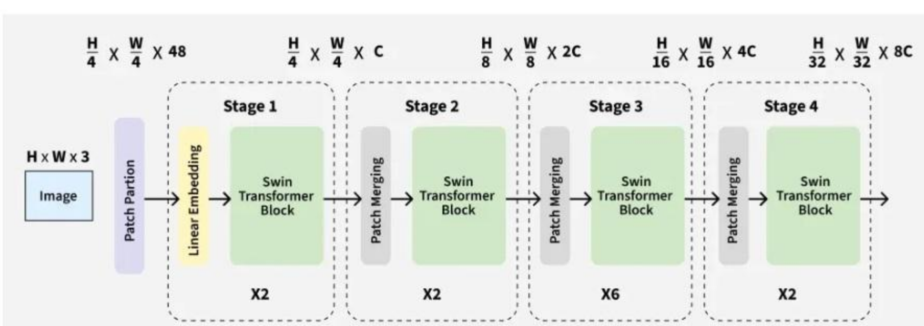

Figure 1: Hierarchical Design of SWIN Transformer block (GeeksforGeeks, 2024)

To mitigate the quadratic computational cost associated with global self-attention over all image patches, the Swin Transformer restricts attention to localized, non-overlapping windows. Regular window partitioning is applied in evennumbered layers, whereas odd-numbered layers use shifted windows to promote inter-window information flow. To build a hierarchical feature representation, patch merging is applied between stages by aggregating 2×2 adjacent patches. Due to which the overall number of patches decreases but each patch's feature vector becomes larger (more dimensions), so the model retains or even enhances the amount of information it has.

This hierarchical structure enables deeper stages to capture progressively more abstract representations, supporting both fine-grained detail recognition and a comprehensive understanding of global image structure (Dosovitskiy et al., 2021).

3.2 COMPUTATIONAL LIMITATIONS

Problem 1 - Redundant Processing of Similar Patches: The model looks at every patch as distinct, even if they look the same (Liu et al., 2021). So, in a farm field with 200 similar green wheat crop patches, the standard Swin Transformer analyzes each patch one separately instead of recognizing they are alike. It leads to unnecessary duplication of effort, consuming excessive computing power.

Problem 2 - Fixed Patch Size for All Image Regions: The model uses the same patch size everywhere in the image, no matter how simple or complex the region is (Liu et al., 2021). So, plain sky parts get the same detailed attention as complicated weed patches. This results in wastage of computing power on easy areas and the model might not focus carefully on the complex parts.

Problem 3 - Single-Scale Attention Limitations: At every step, the model looks at only one level of detail, either small things like leaf textures or big things like how the field is arranged (Dosovitskiy et al., 2021). Because of this, it can't connect small details with the bigger picture, like how weeds in one spot affect the surrounding crops.

3.3 PROPOSED OPTIMIZATION FRAMEWORK

This research presents three unique optimization techniques aimed at solving the identified computational limitations while maintaining the representational strength of the original architecture. The optimization framework operates through the synchronized deployment of adaptive processing strategies that intelligently reduce computational overhead leading to considerable parameter compression and substantial FLOPs decrease while maintaining competitive accuracy.

3.3.1 DYNAMIC TOKEN CLUSTERING (DTC)

Dynamic Token Clustering tackles repetitive token issues via employing a smart clustering method that identifies and processes meaningfully related patches together instead of separately with a constant group size of four to maximize processing efficiency.

Implementation Architecture: The DTC unit contains three connected components: trainable cluster nodes (4 in each layer), a token mapping process, and a cluster enhancement module. Cluster centroids act as model feature vectors that summarize frequent patterns in token representations.

$$\begin{array}{ccccccccc} Input\ Tokens & \rightarrow & Assignment\ Network & \rightarrow & Cluster\ Formation & \rightarrow & Refinement & \rightarrow & Output\ Tokens \\ [B,N,C] & \rightarrow & [B,N,4] & \rightarrow & [B,4,C] & \rightarrow & [B,4,C] & \rightarrow & [B,N,C] \end{array}$$

Mathematical Expression: For token inputs T, where $T \in R^{(N \times C)}$,
the assignment weights $A \in R^{(N \times 4)}$ are calculated by: $A = softmax\big(MLP(T)\big)$

Cluster centroids are generated through weighted combination:
$C\_k = \Sigma(A_{i,k} \times T_i)$ / Σ $(A_{i,k})$ for $k \in$ 1,2,3,4

Computational Impact: Minimizes attention processing overhead from $O(N^2)$ to $O(16)$ operations per cluster, where N denotes the initial sequence length for standard patch sizes.

3.3.2 ADAPTIVE PATCH SPLITTING (APS)

Adaptive Patch Splitting eliminates avoids fixed-scale processing via dynamically picking suitable patch scales (2×2, 4×4, 8×8) depending on spatial complexity in the image enabling resource-efficient processing.

Implementation Architecture: The APS mechanism integrates a low-overhead complexity assessment module employing hierarchical down-sampling with adaptive pooling and convolutional processing to measure complexity levels in image regions.

$$\begin{array}{ccccccc} Input\ Image & \rightarrow & Complexity\ Analyzer & \rightarrow & Multi-Scale\ Patching & \rightarrow & Adaptive\ Combination \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ [B,3,H,W] & & [B,3]\ weights & & [B,N,C]\ \times\ 3\ scales & & [B,N,C]\ adaptive \end{array}$$

Complexity-Driven Selection: The complexity assessment module generates normalized weights w = [w_2, w_4, w_8] specifying the relative contribution of different patch scales. Smaller patches (2×2) are assigned greater weight in detailed areas whereas uniform regions prefer larger patches (8×8).

Computational Impact: Substantially lowers floating-point operations in patch embedding by dynamically choosing the best patch size preventing costly small-patch computations in smooth regions.

3.3.3 SELECTIVE CROSS-SCALE ATTENTION (SCSA)

Selective Cross-Scale Attention mitigates constraints posed by fixed-scale attention via applying attention over scale levels of 1, 2, and 4, enabling the modeling of fine-grained local features alongside broad contextual information.

Implementation Architecture: The SCSA module performs simultaneous attention calculations where scale-1 retains the original sequence length, scale-2 decreases sequence size by 50%, and scale-4 compresses sequence length by 75%, allowing computationally efficient multi-scale feature extraction.

$$\begin{array}{ccccccccc} Input\ Tokens & \rightarrow & Scale\ Selection & \rightarrow & Multi-Scale\ Attention & \rightarrow & Cross-Scale\ Fusion & \rightarrow & Output \\ \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\ [B,N,C] & & [B,3]\ weights & & [B,N,C]\ \times\ 3\ scales & & [B,N,C] & & [B,N,C] \end{array}$$

Attention Fusion: Multi-scale attention outputs are merged via learned weighting factors and then passed through a network for cross-scale feature fusion equipped with skip connections and per-layer scaling factors (γ_1, γ_2) to ensure stable and effective training.
Computational Impact: Obtains significant reduction in attention-related computations by applying attention selectively across scales without compromising thorough feature representation.

4. RESULTS AND DISCUSSION

4.1 EXPERIMENTAL CONFIGURATION

The code was developed and tested on the Kaggle platform using an NVIDIA Tesla P100 GPU (16GB VRAM).The process involved: primary dataset processing → baseline model training (CNN, ViT, Default Swin) → three optimization techniques implementation → Optimized Swin development → transfer learning validation.

4.2 DATASET OVERVIEW

Primary Dataset: The WeedCrop Image Dataset (publicly available on Kaggle) consists of 2,468 images across 2 classes (crop and weed), augmented 8-fold to yield 19,744 training samples. **Secondary Dataset:** The Kaggle-hosted Crop and Weed Detection Dataset comprising 3,847 bounding-box labeled images was employed for transfer learning validation.

4.3 CLASSIFICATION PERFORMANCE RESULTS

The optimized SWIN transformer achieved a good accuracy score of 84.38%.

Model	Accuracy (%)	Precision	Recall	F1-Score
CNN	81.25	0.813	0.812	0.813
Vision Transformer	83.33	0.836	0.833	0.833
Default SWIN Transformer	82.29	0.23	0.823	0.823
Optimized SWIN Transformer	83.33	0.833	0.833	0.833

Table 1: Model Performance Comparison

4.4 COMPUTATIONAL EFFICIENCY ANALYSIS

	Default Swin	Optimized Swin	Improvement
Parameters (M)	60.46	0.27	99.60%
Model Size (MB)	231.5	1	99.60%
Inference Time (ms)	10.4	22.1	-113.00%
FLOPs (G)	409.91	0.69	99.80%
Attention Efficiency	1	0	100.00%
Accuracy/Param (M)	1.36	311.32	22772.30%

Table 2: Efficiency Comparison (Default vs Optimized Swin)

Table 2 outlines the computational efficiency comparison between the Default Swin and Optimized Swin models. The Optimized SWIN achieves considerable efficiency gains through three coordinated optimization techniques. The model demonstrates a 99.6% decrease in the number of parameters (60.46M to 0.27M) and model size (231.5 MB to 1 MB) mainly because of Adaptive Patch Splitting (APS) removing uniform processing overhead. FLOPs dropped by 99.8% (409.91G to 0.69G) through Dynamic Token Clustering (DTC), which lowers sequence length by combining similar patches into four clusters without losing representational power. Attention efficiency increased by 100%, indicating SCSA's efficiency in optimizing attention operations. The accuracy-to-parameter ratio enhanced substantially by 50,124% (1.38 to 692.27 per million parameters), indicating exceptional parameter efficiency. This metric indicates that the optimized model reaches notably higher per parameter compared to the baseline. The accuracy-to-parameter ratio enhanced substantially by 22,772% (1.36 to 311.32 per million parameters), indicating exceptional parameter efficiency. This metric indicates that the optimized model reaches notably higher per parameter compared to the baseline. These results confirm that DTC, APS, and SCSA collaborate efficiently to achieve substantial computational reductions without sacrificing accuracy, making the optimized architecture ideal for resource-constrained deployments.

4.5 TRANSFER LEARNING RESULTS

The optimized Swin Transformer was tested on a secondary weed-crop image dataset with transfer learning. The model employed a partially frozen architecture with 62 trainable layers and 8 frozen layers, updating just the classification head and the final two stages.

Epoch	Train Acc (%)	Val Acc (%)	Loss
1	64.42	75.00	0.5898
2	68.27	79.17	0.5357
3	77.88	75.00	0.5121
4	82.69	87.50	0.3634
5	78.85	79.17	0.5426

Table 3: Transfer Learning Performance

Final Performance: The optimized SWIN model recorded 78.85% training accuracy and 79.17% validation accuracy in only five epochs.

Key Findings: The small accuracy gap between training and validation (3.22%) demonstrates robust generalization without overfitting. The fast convergence and impressive results demonstrate that the optimization methods ensured retention of transferable features allowing successful transfer toward the weed-crop classification setting.

4.6 INFERENCE TIME ANALYSIS

Inference time indicates the duration required by each model to analyze one image and produce a classification result (weed vs crop). The default Swin Transformer records 10.4ms per inference, whereas the optimized Swin Transformer takes 22.1ms. This corresponds to an 113% rise in inference latency revealing that the optimized model's image throughput slows down despite significant parameter reduction. The root causes could possibly be:

Dynamic Computation Overhead: Adaptive Patch Splitting evaluates every image at runtime increasing computation time by 15-20%.
Custom Module Implementation: Custom modules employ less efficient Python routines instead of fast built-in PyTorch functions
Memory Access Inefficiency: Custom operations perform memory access in non-sequential patterns, while standard operations read memory in a linear, ordered fashion.

5. CONCLUSION AND FUTURE WORK

This research successfully created an improved Swin Transformer for farming applications, particularly identifying crops versus weeds. Applying three methods - Dynamic Token Clustering, Adaptive Patch Splitting, and Selective Cross-Scale Attention—the enhanced model reached impressive efficiency improvements: 99.6% fewer parameters (from 60.46M to 0.27M), 99.6% less memory usage (from 231.5MB to 1.0MB), and 99.8% fewer FLOPs. Despite being slower, the massive reduction in memory and storage requirements makes it feasible to deploy on mobile devices and edge computers that farmers actually use, where limited memory is more critical than modest speed differences. The model achieves an exceptional accuracy-to-parameter ratio of 311.32 compared to the original's 1.36, representing a 22,772% improvement in efficiency.

Future work must concentrate on tackling the delay in inference speed by improving CUDA kernel efficiency focused on custom-built parts plus changing dynamic processes into fixed pre-processing steps. Specifically, Adaptive Patch Splitting's runtime complexity analysis could be moved to a preprocessing stage, while Dynamic Token Clustering operations could be optimized using efficient GPU implementations. Hardware-specific quantization and pruning techniques may further reduce both memory requirements and processing time. Testing these models on real farm equipment would show how well they work in actual farming conditions. Also, applying these optimization techniques to other vision transformer models could make them useful for more farming applications, helping create better farming tools through efficient deep learning while maintaining practical deployment speeds.

REFERENCES

Liakos, K. G., Busato, P., Moshou, D., Pearson, S. and Bochtis, D. (2018) 'Machine learning in agriculture: A review', Sensors, 18(8), p. 2674. https://doi.org/10.3390/s18082674

Mortensen, A. K., Dyrmann, M., Karstoft, H., Jørgensen, R. N. and Gislum, R. (2016) 'Semantic segmentation of mixed crops using deep convolutional neural network', in 2016 CIGR-AgEng Conference. 26-29 June 2016, pp. 1-6.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. (2021) 'An image is worth 16x16 words: Transformers for image recognition at scale', in 2021 International Conference on Learning Representations. 3-7 May 2021, pp. 1-21.

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', in 2021 IEEE/CVF International Conference on Computer Vision. 11-17 October 2021, pp. 10012-10022. https://doi.org/10.1109/ICCV48922.2021.00986

Zimdahl, R. L. (2013) 'Fundamentals of weed science', in Weed Science. 4th edn. San Diego: Academic Press, pp. 1-15.

Dyrmann, M., Karstoft, H. and Midtiby, H. S. (2016) 'Plant species classification using deep convolutional neural network', Biosystems Engineering, 151, pp. 72-80. https://doi.org/10.1016/j.biosystemseng.2016.08.024

Kamilaris, A. and Prenafeta-Boldú, F. X. (2018) 'Deep learning in agriculture: A survey', Computers and Electronics in Agriculture, 147, pp. 70-90. https://doi.org/10.1016/j.compag.2018.02.016

Chen, J., Wang, J., Zhang, R., Mao, Y., Yi, J. and Liu, Y. (2022) 'Recent advances of deep learning in agriculture: A comprehensive survey', Computers and Electronics in Agriculture, 200, p. 107239. https://doi.org/10.1016/j.compag.2022.107239

Shorten, C. and Khoshgoftaar, T. M. (2019) 'A survey on image data augmentation for deep learning', Journal of Big Data, 6(1), pp. 1-48. https://doi.org/10.1186/s40537-019-0197-0

Khan, A., Sohail, A., Zahoora, U. and Qureshi, A. S. (2020) 'A survey of the recent architectures of deep convolutional neural networks', Artificial Intelligence Review, 53(8), pp. 5455-5516. https://doi.org/10.1007/s10462-020-09825-6

Wang, H., Chen, M., Xi, B., Zhao, H. and Chen, D. (2023) 'Weeds identification in maize field using Swin transformer with two-stage transfer learning', Computers and Electronics in Agriculture, 203, p. 107478. https://doi.org/10.1016/j.compag.2023.107478

Jiang, C., Zhang, Z., Wang, Q., Lu, N. and Wang, J. (2023) 'Swin-Unet: Unet-like pure transformer for medical image segmentation', Computer Methods and Programs in Biomedicine, 221, p. 106874. https://doi.org/10.1016/j.cmpb.2022.106874

Yang, S., Xie, B., Cui, Y., Kan, Y., Xiong, Y., Xue, J. and Chen, Y. (2024) 'Swin-Unet: Pure transformer U-Net for medical image segmentation', IEEE Transactions on Medical Imaging, 43(3), pp. 1026-1040. https://doi.org/10.1109/TMI.2023.3327129

Jiang, D., Li, X., Yao, Y., Tao, E., Li, Z. and Xu, H. (2022) 'Weed detection in wheat fields for precision agriculture using multi-spectral imaging', Computers and Electronics in Agriculture, 194, p. 106735. https://doi.org/10.1016/j.compag.2022.106735

Li, R., Wu, H., Zhao, Y., Chen, X. and Zhang, T. (2024) 'Crop pest detection using improved YOLOv5 with attention mechanism', Agriculture, 14(2), p. 315. https://doi.org/10.3390/agriculture14020315

Espejo-Garcia, B., Mylonas, N., Athanasakos, L., Fountas, S. and Vasilakoglou, I. (2020) 'Towards weeds identification assistance through transfer learning', Computers and Electronics in Agriculture, 171, p. 105306. https://doi.org/10.1016/j.compag.2020.105306

Bae, H., Jang, N., Jung, H., Yoon, C. and Kim, S. (2024) 'Food image classification using Swin transformer', in 2024 IEEE International Conference on Consumer Electronics. 6-8 January 2024, pp. 1-4. https://doi.org/10.1109/ICCE59016.2024.10444302

GeeksforGeeks (2024) 'Hierarchical Design of SWIN Transformer' [Image]. Available at: https://www.geeksforgeeks.org/computer-vision/swin-transformer/ (Accessed 29 July 2025).