Deep Learning-Driven Pest Detection and Classification with Instant SMS Alerts for Precision Agriculture

Abstract

Pest detection in agriculture is crucial for timely intervention and crop protection, yet it often suffers from limited labeled data and the need for real-time processing. This work investigates the effectiveness of Few-Shot Learning (FSL) as an alternative to traditional deep learning models in scenarios where annotated data is scarce. FSL techniques are evaluated in comparison to conventional models to determine their adaptability, accuracy, and computational efficiency under constrained data conditions.

For large-scale datasets, Vision Transformers (ViT) are employed to leverage their strong representation capabilities. Additionally, Graph Neural Networks (GNNs) are explored to convert image data into graph structures, allowing models to capture spatial and relational patterns among visual elements. Self-Supervised Learning techniques are incorporated to enhance model performance by learning feature representations from unlabeled data, further reducing dependency on manual annotation.

To ensure practical deployment in agricultural fields, the system integrates TensorFlow Lite for lightweight, high-performance inference on mobile and edge devices. OpenCV is utilized to process real-time video streams, enabling the system to analyze continuous input directly from camera feeds and deliver instant pest detection results.

This study presents a comprehensive framework that combines FSL, ViT, GNNs, SSL and real-time video processing to evaluate the effectiveness of FSL under realistic constraints. The goal is to assess whether FSL-based models can outperform or match traditional deep learning approaches in low-data environments while maintaining inference speed and accuracy suitable for field applications. The results aim to guide future development of AI-powered agricultural tools that are both efficient and accessible, even in resource-limited scenarios.

Introduction

**Introduction**

Agricultural productivity is under constant threat from pests, which can devastate crops and significantly reduce yields if not detected and controlled promptly. Traditional pest identification techniques, which rely heavily on visual inspection by human experts, are not only labor-intensive and time-consuming but also prone to errors due to subjective judgment and environmental variability. With the increasing global demand for food and the limited availability of skilled labor in rural areas, there is a pressing need for automated, intelligent pest detection systems that can operate accurately, efficiently, and in real time.

Recent advances in computer vision and deep learning have revolutionized pest detection by enabling automated systems to recognize and classify pest species from images [[1]](https://doi.org/10.1016/j.compag.2019.104906). However, these systems typically require large, annotated datasets for training—something that is often unavailable in real-world agricultural settings due to the high cost and effort involved in data labeling. In such scenarios, Few-Shot Learning (FSL) [[21]](https://doi.org/10.1016/j.engappai.2023.107078) [[22]](https://doi.org/10.1016/j.cropro.2024.106993) emerges as a powerful alternative, allowing models to learn from a very limited number of labeled examples. FSL mimics the human ability to recognize new categories with minimal supervision, making it highly suitable for pest identification in diverse and low-resource environments.

For scenarios where larger datasets are available, the use of Vision Transformers (ViT) [[19]](https://doi.org/10.1016/j.compbiomed.2024.108584) [[20]](https://doi.org/10.1016/j.engappai.2023.107228) offers state-of-the-art performance by capturing long-range dependencies and learning complex representations. In addition, Graph Neural Networks (GNNs) [[5]](https://doi.org/10.1016/j.compind.2019.02.003) are employed to model relationships between visual features by converting image data into structured graphs. This facilitates more structured reasoning and enhances the model’s ability to distinguish between pest species with similar appearances.

To further enhance performance under limited data conditions, Self-Supervised Learning (SSL) techniques [[13]](https://doi.org/10.1016/j.procs.2022.12.396) [[17]](https://doi.org/10.1016/j.compbiomed.2023.106643) are applied to pre-train the model using unlabeled images, allowing it to learn meaningful visual representations without requiring explicit labels. This approach significantly improves generalization and reduces the risk of overfitting when fine-tuned on small labeled datasets.

The proposed system also emphasizes real-time detection and deployment efficiency. By integrating TensorFlow Lite, the model is optimized for on-device inference, enabling deployment on smartphones and edge devices without sacrificing accuracy. OpenCV is utilized to process real-time video feeds from field cameras or drones, allowing continuous monitoring and instantaneous analysis. To facilitate rapid response, the system includes SMS notification integration, which automatically alerts farmers when pests are detected, enabling timely intervention and preventing widespread crop damage.

This research proposes a practical, field-ready pest detection system that combines deep learning with real-time tools. It tackles the challenges of limited labeled data and real-time use, offering a scalable and interpretable solution for precision agriculture. With explainable models and lightweight deployment, it's ideal for rural and resource-limited settings, supporting smarter and more sustainable pest management.

Methodology

#### **Software and Hardware Specifications**

This study uses Windows 11 Home Single Language as the operating system, Jupyter Notebook as the coding environment, and Python 3.12 64-bit for developing and executing machine learning models. The hardware requirements include an Intel Core i5–1204P, 16GB RAM, and 512GB ROM.

#### **Data Acquisition**

The primary dataset used in this project is sourced from Kaggle (<https://www.kaggle.com/simranvolunesia/pest-dataset>) and includes images of nine major agricultural pests: aphids, armyworm, beetle, bollworm, grasshopper, mites, mosquito, sawfly, and stem borer. These full-view images form the foundation for classification and explainability tasks. A secondary dataset from the same source contains half-cropped images of these pests from different angles, aiding in Few-Shot Learning (FSL) and explainability evaluation [[22]](https://doi.org/10.1016/j.cropro.2024.106993).  
 Additionally, a third dataset comprising Malaysian rice pest images includes nine classes with 300 training images per class (2,700 total), adding diversity in shape, color, and orientation, which is essential for training robust and interpretable models [[22]](https://doi.org/10.1016/j.cropro.2024.106993) [[25]](https://doi.org/10.1016/j.compag.2024.108812).

#### **Data Pre-Processing**

Preprocessing ensures the input data is clean, consistent, and suited for learning high-level visual features. Raw color images are used to preserve natural patterns, vital for interpretability in models like Vision Transformers (ViT) [[19]](https://doi.org/10.1016/j.compbiomed.2024.108584) [[20]](https://doi.org/10.1016/j.engappai.2023.107228), GNNs [[5]](https://doi.org/10.1016/j.compind.2019.02.003), FSL [[22]](https://doi.org/10.1016/j.cropro.2024.106993), and MobileNetV2 [[13]](https://doi.org/10.1016/j.procs.2022.12.396).  
 All images are resized to 224×224 pixels for compatibility and computational efficiency. Pixel values are normalized to [0, 1] using a factor of 1./255 to stabilize and accelerate training.

The dataset is split as follows:

* **Training Set (70%)**: for model learning
* **Validation Set (10%)**: for hyperparameter tuning
* **Testing Set (20%)**: for final model evaluation

#### **Data Augmentation**

To improve generalization, the following augmentations are applied:

* **Rotation**: ±30° to simulate various orientations
* **Shifts**: width (±10%) and height (±20%) for displacement simulation
* **Shearing**: 20° for perspective variation
* **Zooming**: up to 80% for scale variability
* **Flipping**: horizontal and vertical to enhance orientation robustness
* **ZCA Whitening**: epsilon = 1e-6 to improve edge sensitivity
* **Fill Mode**: nearest neighbor to handle empty pixels

These techniques are essential for building resilience against real-world variability [[8]](https://doi.org/10.1016/j.compag.2022.107204) [[1]](https://doi.org/10.1016/j.compag.2019.104906) .

### **Few-Shot Learning (FSL)**

Few-Shot Learning (FSL) is a paradigm where models are trained to recognize new categories with only 1–5 labeled samples per class—mirroring human-like generalization [[21]](https://doi.org/10.1016/j.engappai.2023.107078) [[22]](https://doi.org/10.1016/j.cropro.2024.106993) [[23]](https://doi.org/10.1016/j.atech.2023.100307). Unlike conventional deep learning that requires thousands of labeled samples, FSL is especially effective in agriculture where acquiring annotated pest images is labor-intensive and costly [[24]](https://doi.org/10.1016/j.compag.2021.106055) [[25]](https://doi.org/10.1016/j.compag.2024.108812).

#### **How FSL Works**

FSL models are trained using meta-learning or “learning to learn,” through episodes consisting of:

* A **support set**: a few labeled samples per class
* A **query set**: unlabeled examples the model must classify based on the support set

This setup enables the model to learn class-wise similarities and adapt to new categories [[21]](https://doi.org/10.1016/j.engappai.2023.107078) [[22]](https://doi.org/10.1016/j.cropro.2024.106993).

#### **FSL Approaches**

* **Metric-Based** (e.g., Siamese Networks, Prototypical Networks): These use similarity metrics to match queries with support examples.
* **Optimization-Based** (e.g., MAML): Learn models that quickly adapt to new tasks.
* **Memory-Augmented Networks**: Retain knowledge from prior tasks to boost recognition.

#### **Siamese Network**

Siamese Networks are effective metric-based models in FSL. Two subnetworks with shared weights extract features from input pairs (e.g., unknown pest image vs. known example).

* The **distance** between embeddings is calculated (Euclidean or cosine)
* **Contrastive Loss** guides training to minimize the distance for similar images and maximize it (above a margin m) for dissimilar pairs

This approach has shown success in various bio-agricultural tasks [[22]](https://doi.org/10.1016/j.cropro.2024.106993) [[25]](https://doi.org/10.1016/j.compag.2024.108812) [[30]](https://doi.org/10.1016/j.compag.2020.105828).

### **Vision Transformer (ViT)**

Vision Transformers (ViT) leverage transformer architecture for image classification by converting images into sequences of patch embeddings—enabling global context modeling [[11]](https://doi.org/10.1016/j.aej.2024.05.015) [[12]](https://doi.org/10.1016/j.procs.2023.01.209) [[14]](https://doi.org/10.1016/j.eswa.2024.124113).

* **Patch Embedding**: 16×16 patches flattened and projected into high-dimensional vectors
* **Transformer Encoder**: Applies self-attention layers to learn inter-patch relationships
* **Positional Encoding**: Injects spatial structure into the sequence
* **Self-Attention**: Enables the model to focus on key regions across the image

ViT requires large datasets to reach optimal performance but can be fine-tuned for pest classification using transfer learning [[19]](https://doi.org/10.1016/j.compbiomed.2024.108584) [[20]](https://doi.org/10.1016/j.engappai.2023.107228).

### **Graph Neural Networks (GNNs)**

GNNs model images as graphs, where:

* **Nodes** = image regions/features
* **Edges** = spatial/semantic relationships  
   These are well-suited to problems with structured, relational data like pest images in field settings [[5]](https://doi.org/10.1016/j.compind.2019.02.003) [[6]](https://doi.org/10.1016/j.ecoinf.2021.101460).

#### **GNN Mechanics**

* **Message Passing**: Nodes share and aggregate neighbor information
* **Aggregation Functions**: Mean, sum, or max operations used
* **Applications**:  
  + **Node classification**: Identify regions within pests
  + **Graph classification**: Identify entire pest species
  + **Edge prediction**: Understand pest parts or environment  
     Popular variants include GCN, GAT, and GraphSAGE [[6]](https://doi.org/10.1016/j.ecoinf.2021.101460) [[5]](https://doi.org/10.1016/j.compind.2019.02.003).

GNNs have been proven effective in domains requiring relational reasoning such as drug discovery, traffic prediction, and agriculture [[5]](https://doi.org/10.1016/j.compind.2019.02.003) [[6]](https://doi.org/10.1016/j.ecoinf.2021.101460).

### **Self-Supervised Learning (SSL) – MobileNetV2**

MobileNetV2 is a lightweight architecture optimized for edge deployment. While originally designed for supervised learning, it can be adapted for **self-supervised learning** to leverage unlabeled data [[13]](https://doi.org/10.1016/j.procs.2022.12.396) [[17]](https://doi.org/10.1016/j.compbiomed.2023.106643).

#### **Pretext Tasks**

These tasks enable the model to learn without labels:

* **Contrastive Learning** (e.g., SimCLR, MoCo): Discriminate between similar/dissimilar pairs [[17]](https://doi.org/10.1016/j.compbiomed.2023.106643)
* **Rotation Prediction**: Predict image orientation
* **Colorization**: Predict color channels from grayscale
* **Inpainting**: Predict masked areas

#### **Backbone for SSL Frameworks**

MobileNetV2 can be used as the encoder in:

* **BYOL**: Learns without negative samples using Siamese architecture
* **MoCo**: Builds a momentum encoder for contrastive feature learning [[13]](https://doi.org/10.1016/j.procs.2022.12.396)

#### **Advantages**

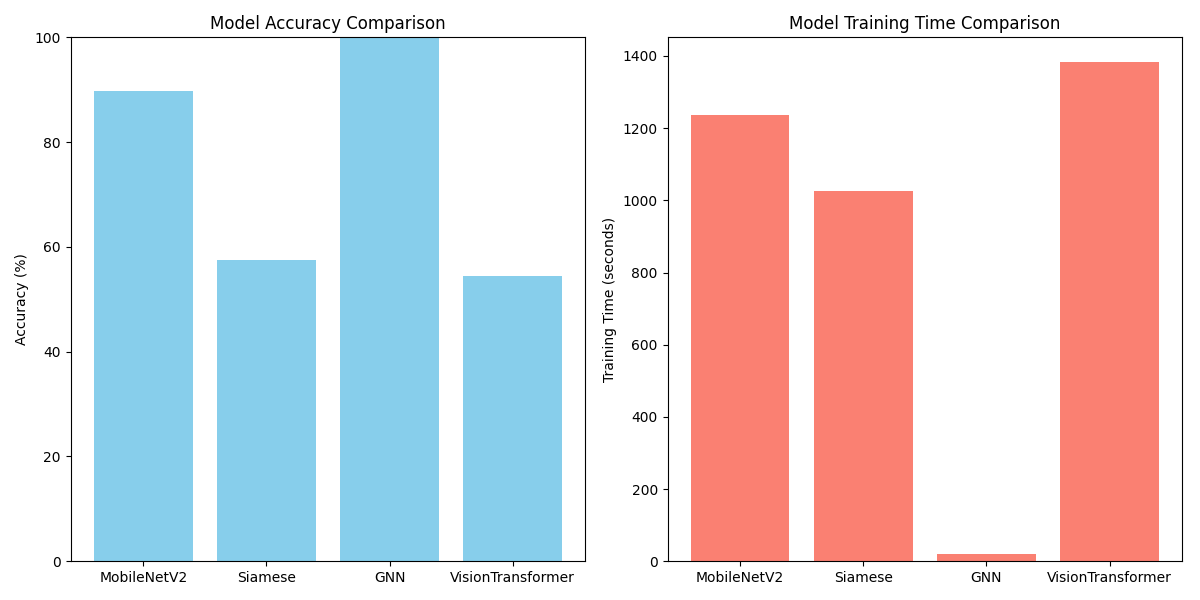
* **Efficiency**: Lightweight and fast, ideal for real-time pest detection
* **Transferability**: SSL-trained features generalize well
* **Scalability**: Effective on both small and large datasets

This makes MobileNetV2 ideal for real-time, rural deployment where labeled data is limited and computation is constrained [[13]](https://doi.org/10.1016/j.procs.2022.12.396) [[17]](https://doi.org/10.1016/j.compbiomed.2023.106643).

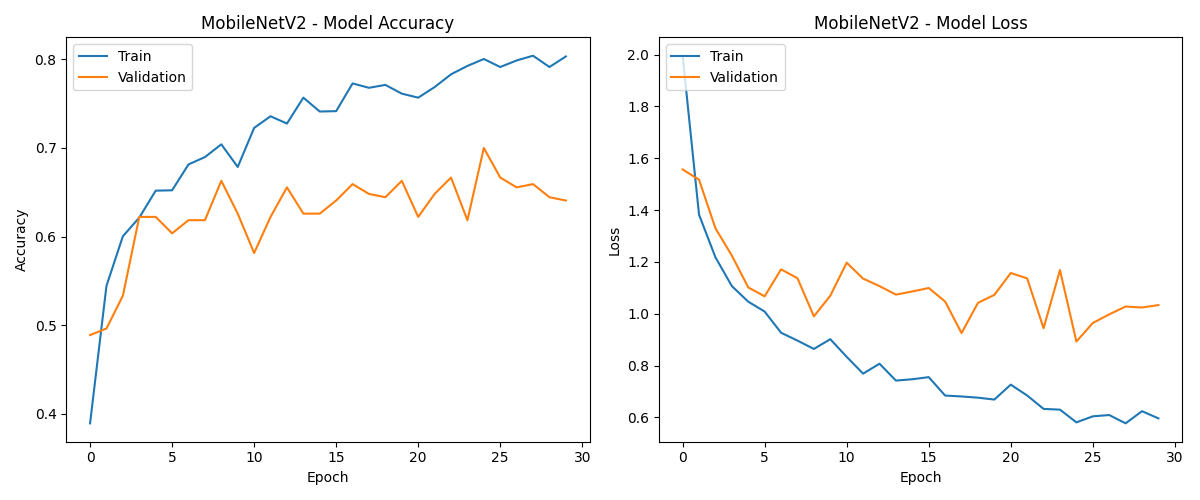
Result and discussion

Result of 4 different models.

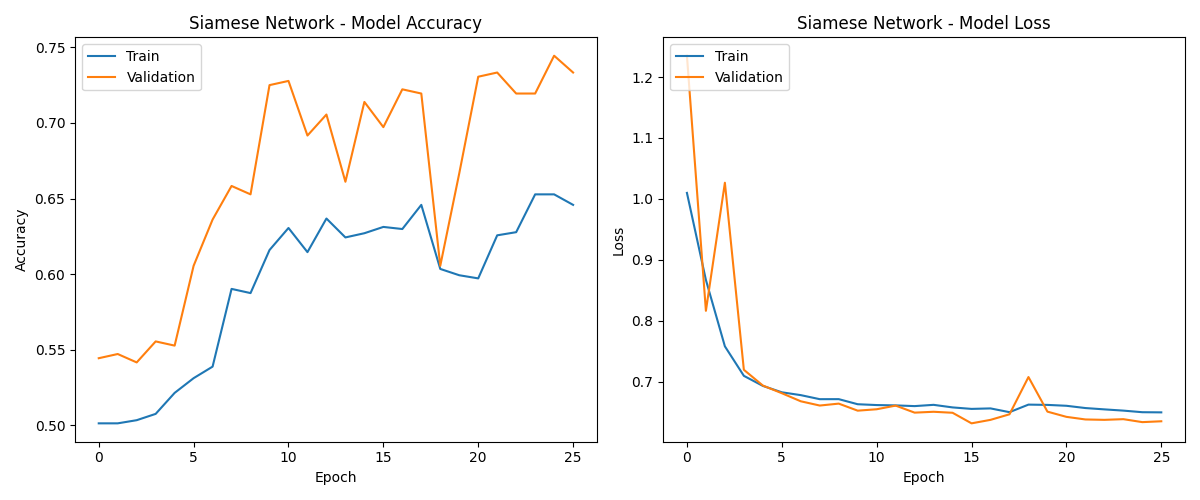
| S.no | Model | Epochs | Training Time | Accuracy |
| --- | --- | --- | --- | --- |
| 1. | MobileNetV2 | 30 | 1253.58s | 89.78% |
| 2. | Siamese FSL | 30 | 1026.8s | 57.66% |
| 3. | GNN | 30 | 19.43s | 100% |
| 4. | VisionTransformer | 30 | 1382.53s | 54.44% |

Model Comparison

MobileNetV2 training history



Siamese Network training history



Visual transformer training history



**References:**

[1] K.Thenmozhi, U. Srinivasulu Reddy, Crop pest classification based on deep convolutional neural network and transfer learning, <https://doi.org/10.1016/j.compag.2019.104906>

[2] Xi Cheng, Youhua Zhang, Yiqiong Chen, Yunzhi Wu, Yi Yue, Pest identification via deep residual learning in complex background, <https://doi.org/10.1016/j.compag.2017.08.005>

[3] Muhammad Attique Khan, Tallha Akram, Muhammad Sharif, Muhammad Awais, Kashif Javed, Hashim Ali, Tanzila Saba, CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features, <https://doi.org/10.1016/j.compag.2018.10.013>

[4] Chengjun Xie, Jie Zhang, Rui Li, Jinyan Li, Peilin Hong, Junfeng Xia, Peng Chen, Automatic classification for field crop insects via multiple-task sparse representation and multiple-kernel learning, <https://doi.org/10.1016/j.compag.2015.10.015>

[5] Solemane Coulibaly, Bernard Kamsu-Foguem, Dantouma Kamissoko, Daouda Traore, Deep neural networks with transfer learning in millet crop images, <https://doi.org/10.1016/j.compind.2019.02.003>

[6] Wenyong Li, Tengfei Zheng, Zhankui Yang, Ming Li, Chuanheng Sun, Xinting Yang, Classification and detection of insects from field images using deep learning for smart pest management: A systematic review, <https://doi.org/10.1016/j.ecoinf.2021.101460>

[7] Zhong-Qiu Zhao, Lin-Hai Ma, Yiu-ming Cheung, Xindong Wu**,** Yuanyan Tang**,** Chun Lung Chen, ApLeaf: An efficient android-based plant leaf identification system, <https://doi.org/10.1016/j.neucom.2014.02.077>

[8] Adhi Setiawan, Novanto Yudistira, Randy Cahya Wihandika, Large scale pest classification using efficient Convolutional Neural Network with augmentation and regularizers, <https://doi.org/10.1016/j.compag.2022.107204>

[9] Loris Nanni, Gianluca Maguolo, Fabio Pancino, Insect pest image detection and recognition based on bio-inspired methods, <https://doi.org/10.1016/j.ecoinf.2020.101089>

[10] Fuji Jian, Sara Doak, Digvir S. Jayas, Paul G. Fields, Noel D.G. White, Comparison of insect detection efficiency by different detection methods, <https://doi.org/10.1016/j.jspr.2016.07.008>

[11] Bin Yang, Binghan Zhang, Yilong Han, Boda Liu, Jiniming Hu, Yiming Jin, Vision transformer-based visual language understanding of the construction process, <https://doi.org/10.1016/j.aej.2024.05.015>

[12] Om Uparkar, Jyoti Bharti, R.K. Pateriya, Rajeev Kumar Gupta, Ashutosh Sharma, Vision Transformer Outperforms Deep Convolutional Neural Network-based Model in Classifying X-ray Images, <https://doi.org/10.1016/j.procs.2023.01.209>

[13] Aiusha V Hujon, Thoudam Doren Singh, Khwairakpam Amitab, Transfer Learning Based Neural Machine Translation of English-Khasi on Low-Resource Settings, <https://doi.org/10.1016/j.procs.2022.12.396>

[14] Xiaoyan Jiang, Shuihua Wang, Yudong Zhang, Vision transformer promotes cancer diagnosis: A comprehensive review, <https://doi.org/10.1016/j.eswa.2024.124113>

[15] Junyu Chen, Eric C. Frey, Yufan He, William P. Segars, Ye Li, Yong Du, TransMorph: Transformer for unsupervised medical image registration, <https://doi.org/10.1016/j.media.2022.102615>

[16] Kyungjin Cho, Jeeyoung Kim, Ki Duk Kim, Sungju Park, Junsik Kim, Jihye Yun, Yura Ahn, Sang Young Oh, Sang Min Lee, Joon Beom Seo, Namkug Kim, MuSiC-ViT: A multi-task Siamese convolutional vision transformer for differentiating change from no-change in follow-up chest radiographs, <https://doi.org/10.1016/j.media.2023.102894>

[17] Madhuri Gokhale, Sraban Kumar Mohanty, Aparajita Ojha, GeneViT: Gene Vision Transformer with Improved DeepInsight for cancer classification, <https://doi.org/10.1016/j.compbiomed.2023.106643>

[18] Elyas Asadi Shamsabadi, Chang Xu, Aravinda S. Rao, Tuan Nguyen, Tuan Ngo, D. António Semblano Gouveia Dias-da-Costa, Vision transformer-based autonomous crack detection on asphalt and concrete surfaces, <https://doi.org/10.1016/j.autcon.2022.104316>

[19] T. Saranya, C. Deisy, S. Sridevi, Efficient agricultural pest classification using vision transformer with hybrid pooled multihead attention, <https://doi.org/10.1016/j.compbiomed.2024.108584>

[20] Zhenzhe Hechen, Wei Huang, Le Yin, Wenjing Xie, Yixin Zhao, Dilated-Windows-based Vision Transformer with Efficient-Suppressive-self-attention for insect pests classification, <https://doi.org/10.1016/j.engappai.2023.107228>

[21] Yan Gao, Haijiang Li, Weiqi Fu, Few-shot learning for image-based bridge damage detection, <https://doi.org/10.1016/j.engappai.2023.107078>

[22] Yinshuo Zhang, Lei Chen, Yuan Yuan, Few-shot agricultural pest recognition based on multimodal masked autoencoder, <https://doi.org/10.1016/j.cropro.2024.106993>

[23] João Vitor de Andrade Porto, Arlinda Cantero Dorsa, Vanessa Aparecida de Moraes Weber, Karla Rejane de Andrade Porto, Hemerson Pistori, Usage of few-shot learning and meta-learning in agriculture: A literature review, <https://doi.org/10.1016/j.atech.2023.100307>

[24] Yang Li, Jiachen Yang, Meta-learning baselines and database for few-shot classification in agriculture, <https://doi.org/10.1016/j.compag.2021.106055>

[25] Masoud Rezaei, Dean Diepeveen, Hamid Laga, Michael G.K. Jones, Ferdous Sohel, Plant disease recognition in a low data scenario using few-shot learning, <https://doi.org/10.1016/j.compag.2024.108812>

[26] Chunshan Wang, Ji Zhou, Chunjiang Zhao**,** Jiuxi Li, Guifa Teng, Huarui Wu,Few-shot vegetable disease recognition model based on image text collaborative representation learning, <https://doi.org/10.1016/j.compag.2021.106098>

[27] Lucas M. Tassis, Renato A. Krohling, Few-shot learning for biotic stress classification of coffee leaves, <https://doi.org/10.1016/j.aiia.2022.04.001>

[28] Ben McEwen, Kaspar Soltero, Stefanie Gutschmidt, Andrew Bainbridge-Smith, James Atlas, Richard Green, Active few-shot learning for rare bioacoustic feature annotation, <https://doi.org/10.1016/j.ecoinf.2024.102734>

[29] Jinchao Pan, Limei Xia, Qiufeng Wu, Yixin Guo, Yiping Chen, Xiaole Tian, Automatic strawberry leaf scorch severity estimation via faster R-CNN and few-shot learning, <https://doi.org/10.1016/j.ecoinf.2022.101706>

[30] Fangming Zhong, Zhikui Chen, Yuchun Zhang, Feng Xia, Zero- and few-shot learning for diseases recognition of Citrus aurantium L. using conditional adversarial autoencoders, <https://doi.org/10.1016/j.compag.2020.105828>