Insect and Plant Image Classification Using Convolutional Neural Networks

Section 1: Problem Definition & Motivation

The goal of this project is to develop a machine learning model that can classify images of insects and plants into predefined categories. Specifically, the system determines whether an insect is present in a given image and identifies the insect species when present. This classification task supports agricultural monitoring by providing automated detection and categorization of pests that affect crops.

Accurate identification of pests plays an essential role in modern agriculture. Different insects require different intervention strategies, and misidentification can result in unnecessary or excessive pesticide use. A reliable classification tool could help reduce pesticide application, support environmentally sustainable crop management, and improve yield protection. Additionally, such a system could aid in tracking the spread of invasive insect species, enabling earlier intervention and helping mitigate ecological and economic impacts.

The primary objective of this project is to develop a high-performing classifier that achieves an accuracy of at least 90% on the test set. Secondary goals include demonstrating strong generalization across multiple insect species and validating that the model performs consistently across training configurations.

Compared to my earlier midterm work, which involved simpler models, this project advances into deep learning using convolutional neural networks (CNNs) and a complex image dataset. The shift to CNNs required more advanced preprocessing, greater computational considerations, and more sophisticated model design, representing a significant step forward in methodological depth.

Section 2: Related Work & Background

To support the development of this model, I referenced several technical resources, including the torchvision documentation for model architectures and transformations, the scikit-learn documentation for evaluation metrics, and the course assignment involving CNN implementation. These resources collectively informed the design of the model, the training pipeline, and the evaluation framework used in this project.

Numerous image classification systems exist, including general-purpose CNNs and specialized pest identification models. However, many existing models are trained on narrow datasets or require extensive computational resources. They may struggle with field variability,

multiple insect species, or differences in imaging conditions. These limitations motivated the use of pretrained architectures and dataset combination to improve robustness.

The core algorithm underlying this project is a convolutional neural network. In particular, I employed a ResNet-18 architecture, which uses residual connections to mitigate vanishing gradients and improve training stability. This architecture is well-established for image classification tasks and offers a balance between accuracy and computational efficiency.

The unique contribution of this project lies in combining two separate datasets, one covering insects and one covering plant health, to create a practical, agriculture-focused classification tool. The model was trained and tuned specifically for this integrated use case, which is not directly addressed by either dataset individually.

Section 3: Methodology & Implementation

The dataset used in this project consists of two primary sources: AgroPest-12, which includes twelve insect categories, and the Agricultural Crops Image Classification dataset, which originally contained thirty plant classes. The plant classes were merged into a single “Healthy” class.

Images were preprocessed through resizing, normalization, and dataset stratification. The final dataset encompassed a diverse range of conditions, environments, and perspectives; however, the inclusion of images taken outside natural settings presented some classification challenges. To mitigate this, I augmented the images by flipping them, rotating them, moving them, adjusting their colors, and randomly erasing parts of them.

I selected a CNN-based approach because CNNs excel at recognizing spatial and hierarchical patterns in image data. Initially, I attempted to train a model from scratch, but training required more than 700 minutes and produced only approximately 20% accuracy with minimal improvement. This demonstrated that training from scratch was computationally impractical on my hardware. Consequently, I adopted ResNet-18 with pretrained ImageNet weights, which provided faster convergence and a stronger baseline performance.

Two training strategies were implemented using the ResNet-18 architecture. In the feature extraction approach, only the final layers of the model were trained with a learning rate of 0.001, allowing for fast convergence while preserving the pretrained features. In the fine-tuning approach, selected convolutional layers were also updated using a smaller learning rate of 0.0001, allowing for more nuanced adaptation of the learned features. Both models were trained for five epochs, which proved sufficient to achieve strong performance while keeping computational demands manageable

Training followed a standard supervised learning pipeline using cross-entropy loss and an Adam optimizer. The validation split consisted of 15% of the data, selected through stratified sampling to preserve class balance and reduce bias. Notably, the non-augmented subset of images was used for validation, allowing for a more realistic performance assessment.

Evaluation relied on accuracy as the primary metric, and results were compared to two baselines: random chance, which would yield roughly 8% accuracy for a thirteen-class problem, and the initial performance of the pretrained model before training, which achieved 62% accuracy for feature extraction and 78% for fine-tuning. Additional analyses, such as confusion matrices and class-specific metrics, were employed to gain a deeper understanding of the strengths and weaknesses across categories.

Section 4: Results & Analysis

The feature extraction approach achieved an accuracy of 83%, while fine-tuning achieved 95%, surpassing the project's target of 90%. These results indicate that even modest fine-tuning of a pretrained model enables strong performance on specialized agricultural datasets.

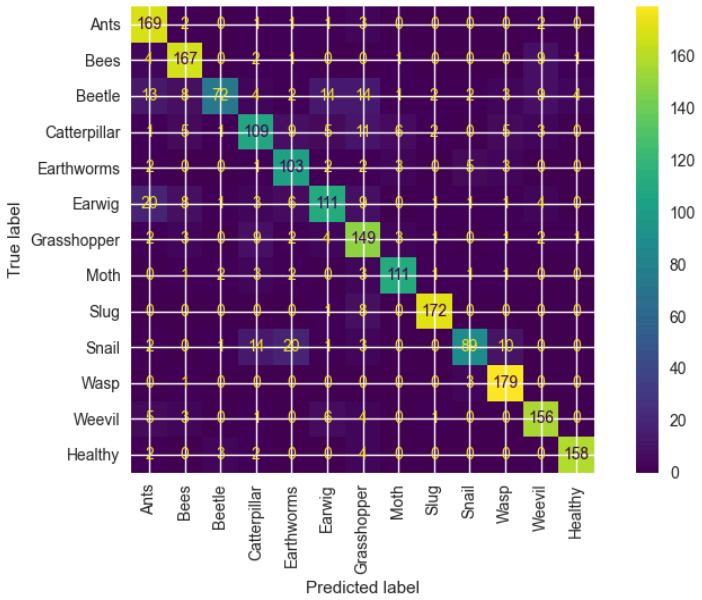
Feature Extraction

Class	Precision	Recall	F1-Score	Support
Ants	0.77	0.94	0.85	179
Bees	0.84	0.9	0.87	185
Beetle	0.9	0.49	0.63	148
Catterpillar	0.73	0.69	0.71	157
Earthworms	0.71	0.85	0.77	121
Earwig	0.77	0.67	0.72	165
Grasshopper	0.71	0.84	0.77	177
Moth	0.89	0.89	0.89	125
Slug	0.96	0.95	0.95	181
Snail	0.88	0.64	0.74	140
Wasp	0.88	0.98	0.93	183
Weevil	0.84	0.89	0.86	176
Healthy	0.96	0.93	0.95	169
Accuracy	—	—	0.83	2106
Macro Avg	0.83	0.82	0.82	2106
Weighted Avg	0.84	0.83	0.82	2106

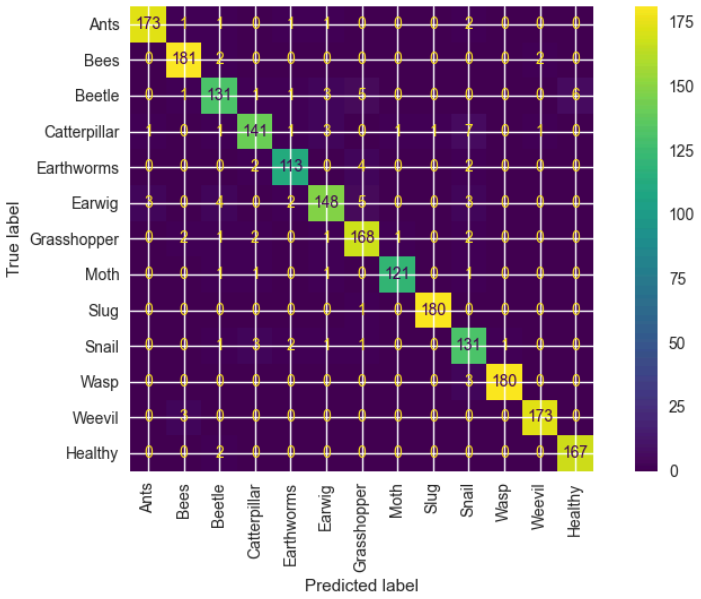
Fine-tuning

Class	Precision	Recall	F1-Score	Support
Ants	0.98	0.97	0.97	179
Bees	0.96	0.98	0.97	185
Beetle	0.91	0.89	0.9	148
Catterpillar	0.94	0.9	0.92	157
Earthworms	0.94	0.93	0.94	121
Earwig	0.94	0.9	0.92	165
Grasshopper	0.91	0.95	0.93	177
Moth	0.98	0.97	0.98	125
Slug	0.99	0.99	0.99	181
Snail	0.87	0.94	0.9	140
Wasp	0.99	0.98	0.99	183
Weevil	0.98	0.98	0.98	176
Healthy	0.97	0.99	0.98	169
Accuracy	—	—	0.95	2106
Macro Avg	0.95	0.95	0.95	2106
Weighted Avg	0.95	0.95	0.95	2106

Feature Extraction



Fine-tuning



Both training strategies significantly outperformed random chance and the untrained pretrained model baselines. Feature extraction improved accuracy by more than 20% compared to the baseline pretrained model, while fine-tuning improved accuracy by nearly 20 percentage points beyond that.

Model performance varied by class. Snails and earthworms were misclassified more frequently, likely due to images showing these species in unnatural environments such as sidewalks. These atypical contexts may have encouraged the model to rely on background cues rather than insect morphology. The beetle class also showed weaker recall relative to precision, suggesting that the model may have overfit to a particular subset of beetle images. This inconsistency indicates a need for more diverse beetle exemplars in the dataset.

Section 5: Conclusions & Future Work

This project demonstrates that convolutional neural networks, especially pretrained architectures like ResNet-18, are highly effective for insect classification tasks. Fine-tuning significantly improves accuracy, and combining multiple datasets can create a robust classification tool for agricultural applications.

Several constraints affected this project. The datasets included many staged or unnatural images, reducing ecological realism. Training time and computational limitations prevented

training from scratch or experimenting with larger models. Additionally, the healthy plant class was broad and visually diverse, making classification more challenging.

Future improvements could include isolating insects from their backgrounds and compositing them onto images of actual crops to create a more realistic dataset. Additional exploration of alternative pretrained architectures, such as EfficientNet or Vision Transformers, may also improve performance. Expanding the dataset to include more natural field images would further strengthen generalization.

Throughout this project, I gained deeper insight into the complexities of CNNs, the importance of dataset quality, and the trade-offs involved in training larger models. The experience reinforced how data selection and preprocessing can significantly influence model performance, just as much as the model architecture itself.