UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

6CS012/HJ1: Artificial Intelligence and Machine Learning

Development of a CNN-Based Model for Multi-Class Insect Classification

Full Name : Diparshan Baral

University ID : 2358244

Module Leader : Siman Giri

University email : D.Baral@wlv.ac.uk

Date of submission: 2025-05-20

Abstract

This report describes the development and testing of a deep learning based multi-class insect classification for agricultural pest management and ecologically focused applications. We aimed to develop and compare CNN architectures learned from scratch and their resultant performance and whether architecturally more complicated networks result in greater classification accuracy. We also tested the influence of various optimizers, namely Adam and SGD on convergence rates and resultant model performance. A deeper CNN architecture incorporating batch normalization and dropout regularization techniques was implemented but failed to surpass the baseline CNN due to underfitting, likely caused by insufficient data and improper hyperparameter settings. To overcome these limits, transfer learning with MobileNetV2 was used, resulting in higher performance with a weighted F1-score of 0.9587 over a test set of nine insect species. The purpose of this study is to demonstrate the importance of balancing model complexity and datasets, as well as the effectiveness of pre-trained architecture in small-scale picture classification problems. Its utilization is proven to greatly outperform training from scratch on small datasets, and it recommends future study paths such as increased data augmentation, hyperparameter tweaking, and an inquiry into employing different pre-trained architectures.

Table of Contents

Table of Figures

1. Introduction

Accurate and automatic insect species identification is critical for modern agricultural management and ecological monitoring networks. Manually identifying insects is labor-intensive and time-consuming, and it is prone to human error in field situations where species mimic one another, making decision making difficult. Deep learning using Convolutional Neural Networks (CNNs) has proven to be extremely effective in handling complex image categorization issues across a wide range of fields.

This project will develop an efficient CNN-based classifier for identifying insect species for crop defense utilizing both from-scratch training and transfer learning approaches. The project is based on current deep learning architectures and includes ideas such as feature extraction, regularization with dropout and batch normalization, and optimizer comparison. The research scope covers a large experiment space with varying model complexity, training efficiency comparison, and generalizability performance evaluation.

This work advances practical approaches for smart agriculture and biodiversity monitoring by addressing the difficulty of insect classification using deep learning. This work also discusses real-world limits encountered during training and suggests potential solutions for overcoming them in comparable situations.

2. Dataset

The data set used in this study consists of images of insects belonging to nine distinct classes. The dataset was custom and provided by the module leader. Each class represents a commonly observed agricultural pest or ecologically significant insect species.

Dataset Characteristics:

- Total Images : 2,616
- Training Set : 2,232 images (~85%)
- Test Set : 384 images (~15%)

- Image Resolution : Resized to 224×224 pixels to match input requirements of standard CNN architectures.
- Image Format : JPEG format, grouped into class-specific directories.

2.1. Class Distribution

The dataset is relatively balanced, with each class containing approximately 200–300 images:

- Training Set Class Counts: Aphids (266), Mosquito (295), Armyworm (223), Mites (254), Stem Borer (181), Beetle (291), Sawfly (200), Grasshopper (277), Bollworm (245).
- Test Set Class Counts: Aphids (44), Mosquito (50), Armyworm (43), Mites (42), Stem Borer (36), Beetle (50), Sawfly (37), Grasshopper (46), Bollworm (36).

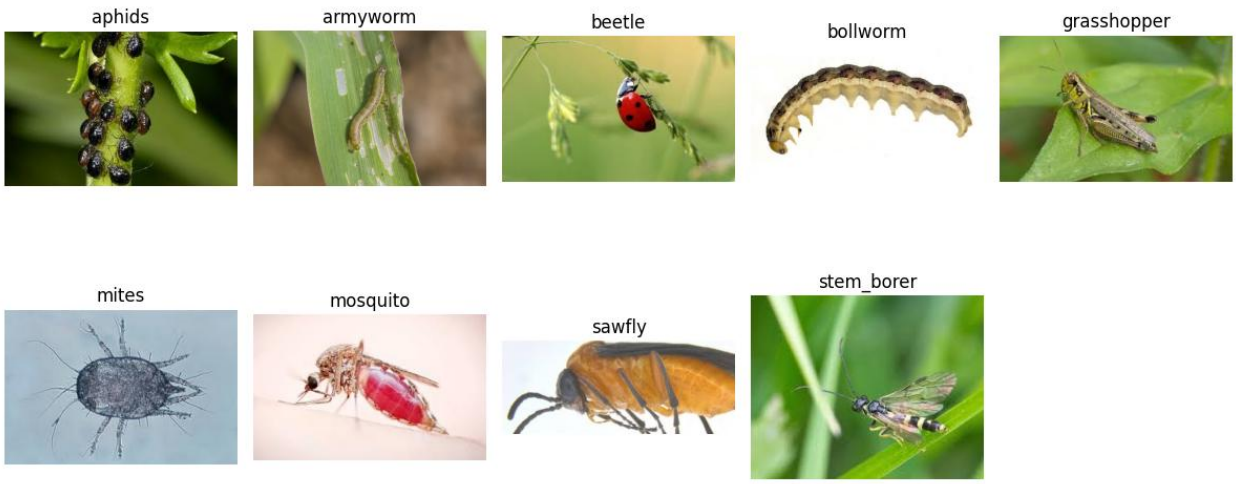

Figure 1: Insects visualization

2.2. Preprocessing and Data Cleaning

To ensure data quality and consistency:

- All images were verified for corruption using PIL's Image.verify() method; corrupted files were removed automatically.
- Visual inspection confirmed that images were representative of their respective classes and contained minimal noise or distortion.

- Dataset was split into training and validation sets using an 80/20 ratio, with a separate test set preserved for final evaluation.
- Pixel values were normalized to the range [0,1].
- Duplicate images were identified based on filename patterns (e.g., "Copy") and removed to prevent data leakage and overestimation of performance.

These preprocessing steps laid the foundation for reliable model training and unbiased evaluation.

Figure 2: Training Set Class Distribution

3. Methodology

We designed and evaluated three distinct approaches to classify insect species:

3.1. Baseline CNN Architecture

The baseline model consisted of a standard CNN with:

- Three convolutional blocks, each composed of a Conv2D layer followed by MaxPooling2D.

- Three fully connected layers.
- Softmax output layer for multi-class classification.

This architecture was selected for its simplicity and ability to extract meaningful spatial features without excessive computational demand.

3.2. Deeper CNN Architecture

An extended version of the baseline model was constructed to assess whether increased depth improves classification performance. It incorporated:

- Additional convolutional and pooling layers.
- Batch Normalization to stabilize training dynamics.
- Dropout regularization to mitigate overfitting.

Despite being more expressive, this model suffered from severe underfitting, indicating structural mismatch with the dataset size and characteristics.

3.3. Transfer Learning Using MobileNetV2

Given the limited size of the dataset, we adopted a transfer learning approach using MobileNetV2, a lightweight yet powerful architecture pre-trained on ImageNet. The methodology involved:

- Removing the top classification layers of MobileNetV2.
- Adding custom dense layers adapted to the target task.
- Freezing base layers during initial training to preserve learned representations.
- Fine-tuning selects top layers after feature extraction phase.

This approach leveraged rich feature hierarchies already learned from large-scale image data and proved essential in achieving high classification accuracy.

3.4. Model Compilation and Training Setup

All models were compiled using categorical crossentropy loss, appropriate for multi-class classification. Optimizers included:

- Adam with learning rate 5×10^{-4} for the baseline and deeper models.

- SGD with momentum (0.9) for comparative optimizer analysis.
- Fine-tuned Adam at 1×10^{-5} during the transfer learning phase.

Hyperparameters such as batch size (64), number of epochs (up to 50), and early stopping criteria were tuned to optimize training stability and convergence.

4. Experiments and Results

4.1. Baseline vs. Deeper Architecture

The baseline CNN achieved reasonable results, with a test accuracy of approximately 59%. Its validation accuracy reached ~50%, and it demonstrated stable convergence throughout training.

In contrast, the deeper CNN model, despite having additional filters and layers, performed poorly. During training:

- Validation accuracy remained around 12%, equivalent to random guessing.
- Weighted F1-score was only 0.0375, suggesting that the model did not learn discriminative features.
- Training curves showed little improvement beyond early epochs, regardless of optimizer choice.

This disparity indicates that increasing model depth without sufficient data does not necessarily improve performance and may lead to instability or underfitting.

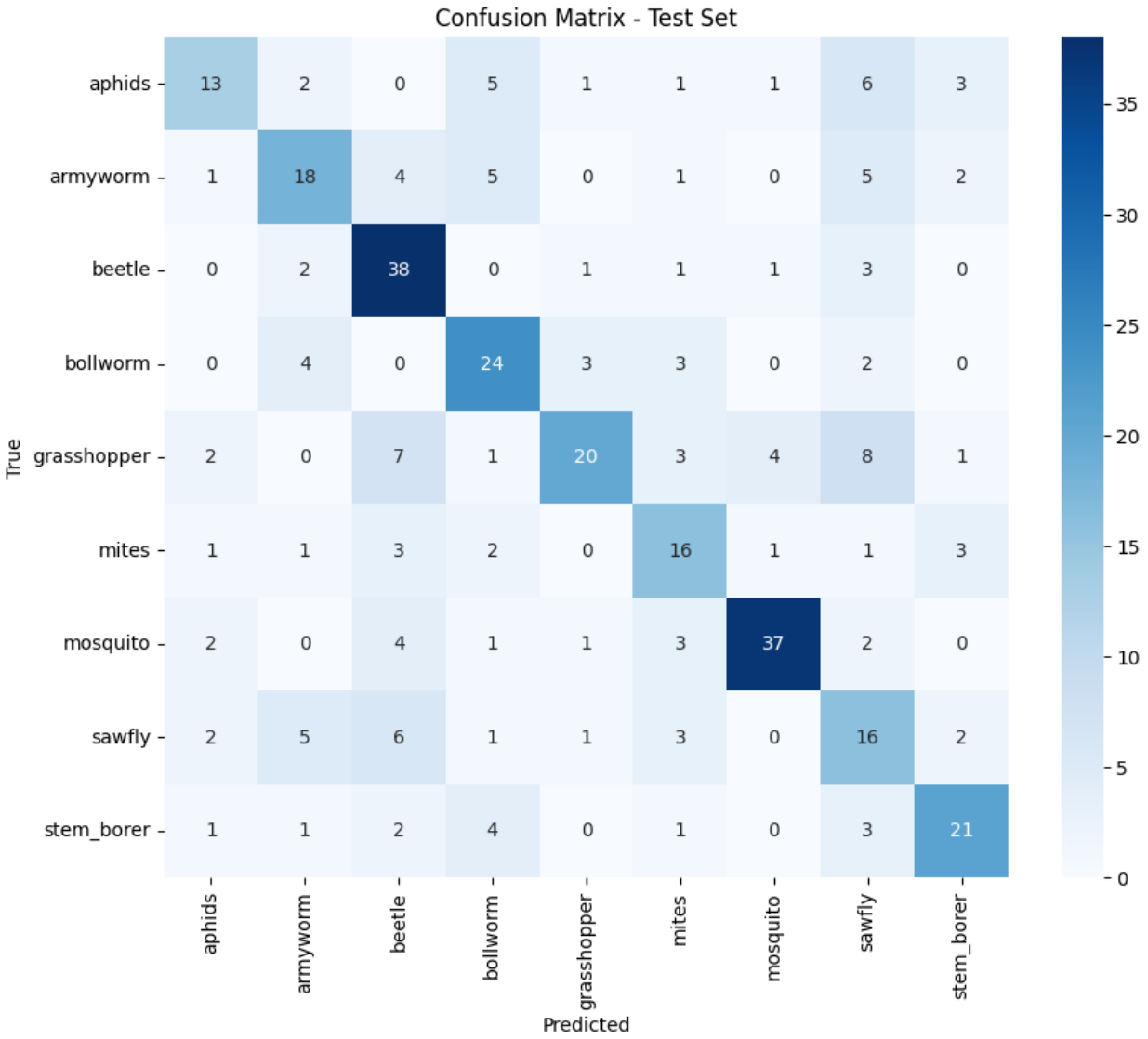

Figure 3: Baseline Model Confusion Matrix

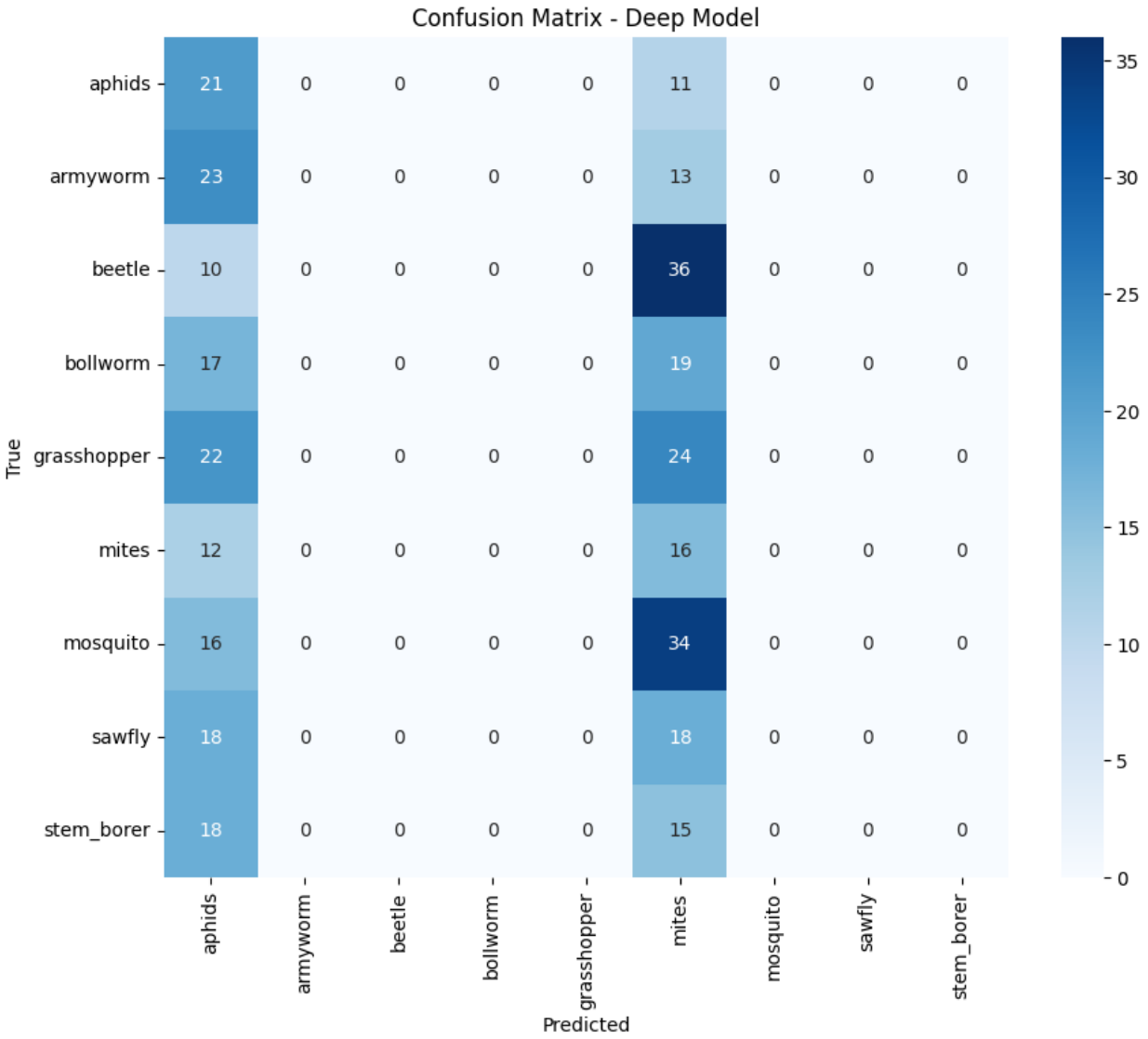

Figure 4: Deeper Architecture Confusion Matrix

4.2. Computational Efficiency

We analyzed computational cost by measuring training duration and resource utilization:

- Baseline CNN : Completed in approximately 140 seconds, showing rapid convergence and efficient GPU usage.
- Deeper CNN :
  - Trained for 50 epochs.
  - With Adam optimizer, it required 282.37 seconds.

- With SGD optimizer, completed faster (70.22 seconds) but yielded comparable performance.

The deeper CNN incurred higher memory and processing overhead without corresponding gains in classification accuracy, highlighting inefficiencies in model design relative to data availability.

4.3. Training with Different Optimizers

Both Adam and SGD were tested on the deeper CNN model:

Adam Optimizer :

- Faster convergence initially.
- Early plateauing of accuracy and loss curves.
- Final test accuracy: ~15%

SGD with Momentum (0.9) :

- Slower convergence compared to Adam.
- No significant improvement in final performance.
- Final test accuracy: ~12%

While Adam showed marginally better behavior, neither optimizer could overcome the limitations imposed by the underlying architecture and dataset constraints.

4.4. Challenges in Training

Several challenges were encountered during model training:

- Underfitting in Deeper Models : Despite regularization, deeper CNNs failed to learn meaningful patterns.
- Overfitting in Baseline CNN : Observed in later epochs, mitigated via early stopping and learning rate reduction.
- Data Limitations : After removing duplicate images, the dataset became so small that the deeper model could not work properly. The dataset with duplicate images could not be used because it lacked in generalization(Overfitted).
- Hardware Constraints : Limited GPU access time in google colab.

Total training time for the deeper architecture exceeded expectations, further supporting the conclusion that model complexity must be carefully matched to data quantity and quality.

5. Fine-Tuning or Transfer Learning

We fine-tuned the MobileNetV2 architecture for our specific insect classification task. The base model was initially frozen, and custom dense layers were added on top to adapt to the nine-class classification problem. After an initial feature extraction phase, selective fine-tuning was applied to the last twenty layers of the base model.

Performance Evaluation:

Validation Accuracy : Reached ~94%

Test Accuracy : Achieved ~96%

Weighted F1-Score : 0.9587

These results represent a substantial improvement over models trained from scratch.

METRIC	BASELINE CNN	DEEPER CNN	TRANSFER LEARNING
Test Accuracy	~59%	~15%	~96%
F1-Score (Weighted)	0.6359	0.0375	0.9587
Training Stability	Stable	Unstable	Highly stable
Convergence Speed	Moderate	Slow	Fast

The transfer learning model exhibited exceptional performance, demonstrating that leveraging pre-trained weights can dramatically enhance classification accuracy even in scenarios with limited labeled data.

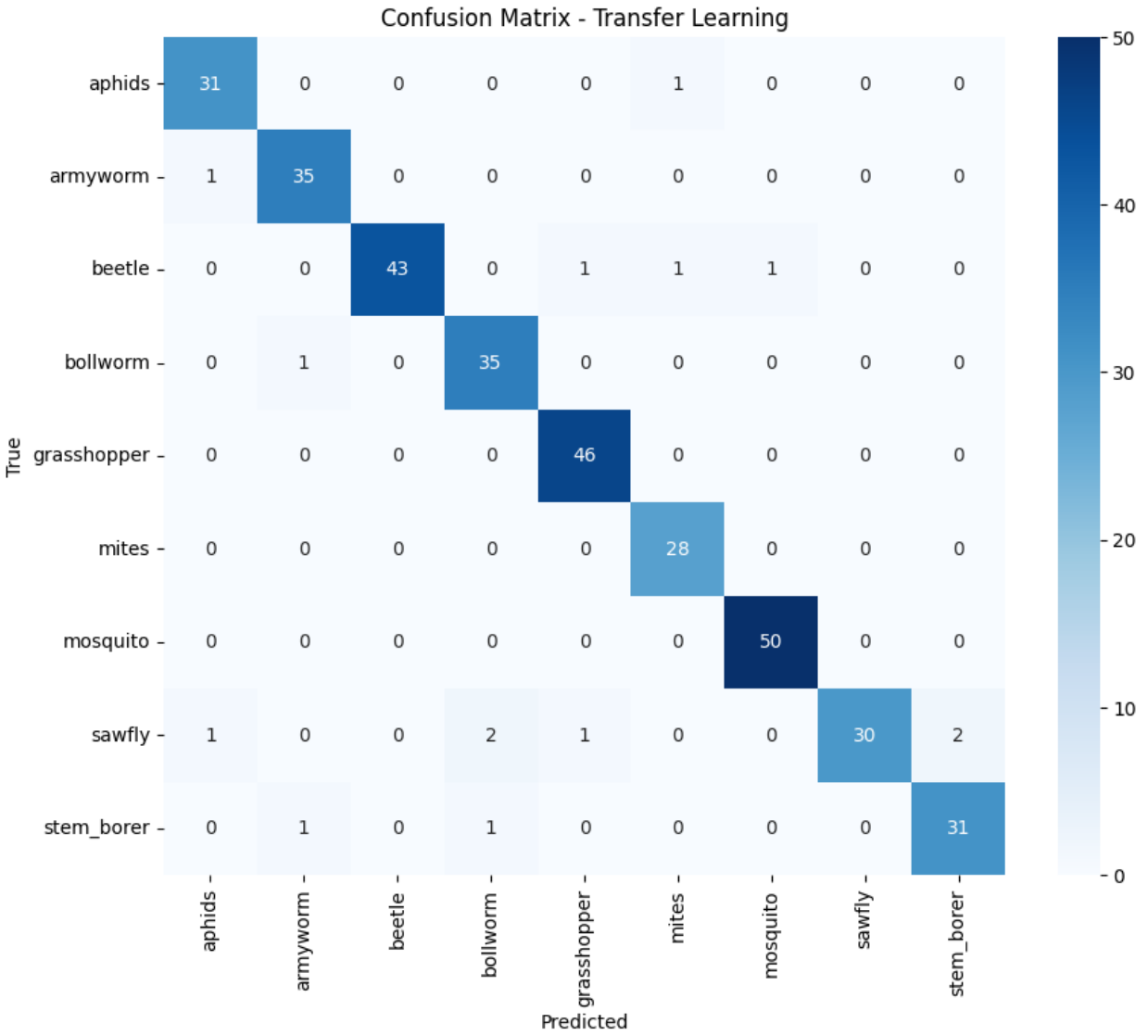

Figure 5: Transfer Learning Confusion Matrix

6. Conclusion and Future Work

This study addresses the CNN-based approaches to insect classification for agricultural and ecological purposes. Deeper models converged poorly while a standard CNN achieved medium accuracy using both Adam and SGD optimizers, indicating low-data regime problems. However, transfer learning with MobileNetV2 achieved high classification accuracy and confirmed its validity for small-scale image tasks.

The results demonstrate the importance of model choice in data-scarce settings and clearly establish pre-trained architectures' advantages. Deployment on handheld devices

and extension to detection and segmentation for complete pest tracking is seen as directions for future work using semi-supervised learning.