UNIVERSITY PARTNER

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**

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

Date of submission: 2025-05-14

Abstract

In this report, we present the development and evaluation of a deep learning-based model for multi-class insect classification with applications in agricultural pest management and ecological research. The objective was to construct and compare convolutional neural network (CNN) architectures trained from scratch, analyze their performance, and evaluate whether increasing architectural complexity leads to improved classification accuracy. Furthermore, we explored the impact of various optimizers—namely Adam and SGD—on convergence speed and final 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 limitations, we employed transfer learning using MobileNetV2, which demonstrated superior performance, achieving a weighted F1-score of 0.9587 on a test set comprising nine distinct insect species. This study underscores the importance of matching model complexity with dataset size and highlights the efficacy of pre-trained models in small-scale image classification tasks. We conclude that transfer learning significantly outperforms training from scratch when working with limited datasets and propose future directions including advanced data augmentation, hyperparameter optimization, and exploration of alternative pre-trained architectures.

1. Introduction

Accurate and automated classification of insect species is crucial for modern agricultural practices and ecological monitoring systems. Manual identification is labor-intensive, time-consuming, and prone to error, especially in field conditions where visual similarity between species complicates decision-making. Deep learning, particularly Convolutional Neural Networks (CNNs), has shown remarkable success in solving complex image classification problems across diverse domains.

This project aims to develop a robust CNN-based classifier for identifying insect species relevant to crop protection, leveraging both from-scratch training and transfer learning strategies. The work builds upon established methodologies in deep learning, integrating concepts such as feature extraction, regularization via dropout and batch normalization, and optimizer comparison. The scope includes rigorous experimentation with varying model complexities, comparative analysis of training efficiency, and evaluation of generalization performance.

By addressing the challenge of insect classification through deep learning, this study contributes to scalable solutions for real-world deployment in smart agriculture and biodiversity tracking systems. It also provides insights into practical limitations encountered during model training and offers recommendations for overcoming them in similar contexts.

2. Dataset

The dataset used in this study consists of images of insects belonging to nine distinct classes, curated specifically for academic use within the context of this assignment. 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).

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.

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.

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.

6. Conclusion and Future Work

This study explores CNN-based approaches for insect classification in agricultural and ecological contexts. A baseline CNN achieved moderate accuracy, while deeper architectures failed to converge effectively under both Adam and SGD optimizers, highlighting limitations in low-data regimes. In contrast, transfer learning with MobileNetV2 achieved high classification accuracy, confirming its effectiveness for small-scale image tasks.

The results emphasize the importance of model selection when data is scarce and demonstrate the clear benefits of pre-trained architectures. Future work should explore semi-supervised learning, deployment on mobile platforms, and expansion to detection and segmentation for comprehensive pest monitoring.