CNN-Based Plant Disease Detection with Targeted Data Augmentation and Class Balancing

*Erhan KURNAZ*

*CCT COLLEGE*

*Dublin/Ireland*

[*2024534@student.cct.ie*](mailto:2024534@student.cct.ie)

***Abstract*— Plant diseases pose a major threat to global food security by reducing crop productivity and quality. Traditional methods for disease detection are time-consuming and often inaccurate, especially when scaling to large agricultural datasets. In this study, a deep learning-based image classification system using Convolutional Neural Networks (CNNs) was developed to detect and classify 23 different plant leaf conditions, including both healthy and diseased states. The initial baseline model achieved a test accuracy of 71%, but struggled with class imbalance and visual similarity between certain diseases. To address this, a targeted data augmentation strategy was applied to underrepresented classes, and the CNN architecture was refined for better generalization. After these enhancements, the final model achieved a test accuracy of 91%. Performance evaluations through accuracy/loss curves, confusion matrices, and classification reports confirmed significant improvements in both overall accuracy and minority class recognition. This work demonstrates the effectiveness of augmentation and architectural optimization in developing scalable, robust plant disease detection systems.**

**Keywords—Plant disease detection; Deep learning; Convolutional Neural Networks (CNN); Image classification; Data augmentation; Class imbalance; Precision agriculture**

1.Introduction

Plant diseases constitute one of the most significant threats to global food production and agricultural sustainability. These diseases not only result in substantial crop yield losses but also pose serious risks to food security, particularly in regions where agriculture is the primary livelihood source (FAO, 2020). According to recent estimates, up to 40% of global crop production is lost annually due to pests and diseases, costing the global economy billions of dollars and endangering the livelihoods of millions of smallholder farmers. The consequences are especially severe in developing nations, where limited access to diagnostic tools exacerbates the vulnerability of crops to disease outbreaks.

With the global population continuing to rise, the demand for food is increasing at an unprecedented rate, necessitating the adoption of efficient, scalable, and environmentally sustainable agricultural practices. Among the strategies that have gained traction in recent years is the integration of digital technologies, particularly those driven by Artificial Intelligence (AI), to enable smarter and more proactive crop monitoring. Early detection of plant diseases is critical not only for minimizing economic losses but also for reducing the overuse of chemical pesticides, which can have adverse environmental and health impacts.

Traditionally, plant disease detection has relied on manual inspection by experts—a method that is inherently time-consuming, labor-intensive, and unsuitable for large-scale agricultural operations (Miller, 2018).Moreover, the accuracy of manual diagnosis is often inconsistent, as it depends on human perception, prior knowledge, and access to expert consultation. These challenges highlight the urgent need for automated systems that can identify plant diseases reliably, rapidly, and at scale.

Recent advancements in AI, especially in Deep Learning (DL), offer promising alternatives. In particular, Convolutional Neural Networks (CNNs) have demonstrated exceptional performance in image classification tasks, including medical imaging, object recognition, and more recently, agricultural diagnostics. CNNs are capable of learning complex hierarchical features directly from raw image data, eliminating the need for hand-crafted feature extraction, which has traditionally been a bottleneck in conventional machine learning pipelines. This has led to a surge in research focusing on the application of CNN-based systems for automated plant disease detection from leaf images.

Despite the promising performance of CNNs in various controlled experiments, several challenges remain. One of the most persistent issues in this domain is dataset imbalance—a situation where certain disease categories are underrepresented in training data, leading to biased model performance and poor generalization. Addressing this problem is essential for developing robust and fair models capable of detecting both common and rare diseases with high accuracy.

This research aims to develop a CNN-based image classification model capable of accurately detecting plant diseases from labeled leaf images. A key focus of the study is addressing class imbalance in the dataset, which often leads to skewed predictive performance and limited real-world applicability. To mitigate this, data augmentation techniques—such as rotation, flipping, and brightness enhancement—will be employed to synthetically increase the diversity and volume of minority class samples, thereby improving the model’s generalizability.

By exploring the practical application of CNNs within the context of agriculture, this study contributes to the evolving field of precision farming, which leverages data-driven technologies to enhance crop health monitoring and yield optimization. The proposed image-based disease detection system is designed to be non-invasive, scalable, and adaptable for real-world deployment, offering a cost-effective and sustainable tool for early disease identification and intervention. In addition to technical performance, the research also emphasizes the importance of designing models that are lightweight and deployable in low-resource settings such as rural farms with limited connectivity or processing power.

2.RELATED WORK

Recent advances in Deep Learning (DL), particularly Convolutional Neural Networks (CNNs), have significantly improved the accuracy of plant disease detection using leaf images. Earlier approaches relied on handcrafted features such as shape, color, and texture, often combined with machine learning classifiers like Support Vector Machines (SVM) and Random Forests (RF). However, these techniques were limited in terms of scalability, generalizability, and feature dependency.

CNNs overcame these limitations by automatically learning hierarchical feature representations from raw images. One of the earliest notable studies, Mohanty et al. (al., 2016), applied AlexNet and GoogleNet architectures on the PlantVillage dataset, achieving 99.34% accuracy across 38 classes. Similarly, Ferentinos et al. (Ferentinos, 2018) experimented with AlexNetOWTBn and VGG networks, evaluating performance over 58 classes and achieving up to 99.53% accuracy—one of the highest reported in the literature.

Transfer learning has also been widely adopted. Aravind et al. employed six pretrained CNN models (including VGG, ResNet, DenseNet, and GoogleNet) and reported the best performance (97.3%) using GoogleNet. Chen et al. implemented the INC-VGGN model and achieved 92% accuracy, while Li et al. combined CNN-extracted features with SVM and RF classifiers, reaching 94% accuracy.

Several other studies have contributed additional methodological innovations. For instance, Ramcharan et al. applied pretrained models to cassava disease detection and achieved 99.09% accuracy. Oyewola et al. (O. Oyewola, 2021) used a Deep Residual Neural Network (DRNN) architecture for cassava disease classification, outperforming traditional CNNs with 96.75% accuracy. Fuentes et al. introduced an object detection perspective, employing Faster R-CNN to detect both diseases and pests in tomato plants, achieving a precision of 85.98%.

Some studies addressed the data limitation problem by generating synthetic images. Abbas et al. used Conditional GANs to produce synthetic tomato leaf images and improved model robustness. Others, such as Jadhav et al., proposed histogram transformation techniques to enhance image quality and enrich training data. Class imbalance issues have also been approached through various data

|  |  |  |  |
| --- | --- | --- | --- |
| Author | Model | DataSet / Class | Model Performance |
| Mohanty et al. | AlexNet, GoogleNet | PlantVillage/38 Class | 99.34 |
| Ferentinos et al | AlexNetOWTBn,VGG | PlantVillage/54 Class | 99.53 |
| Oyewola et al. | DRNN | Cassava/5 Class | 99.75 |
| Fuentes et al. | Faster R-CNN | 9 Class | 85.98 |
| Ramcharan et al. | Pretrained CNN | PlantVillage | 99.09 |
|  |  |  |  |

augmentation strategies.

Table 1 Summarizes the key findings of selected CNN-based studies on plant disease classification

3.Methodology  
This section outlines the methodological approach adopted to develop a CNN-based image classification model for plant disease detection. The overall workflow comprises five major components: data acquisition and preprocessing, class balancing through data augmentation, model development and training, evaluation, and deployment. Due to the volume of image data and the need for scalable processing, the framework was implemented using Apache Spark with PySpark integration.

3.1 Dataset and Computational Environment

The dataset used in this study includes **2,230 RGB images** of both diseased and healthy plant leaves, categorized into **23 classes** representing different plant species and disease conditions (e.g., apple, tomato, potato, pepper). Each class corresponds to a unique plant-disease pair, such as *Apple Black Rot* or *Tomato Leaf Mold*. The dataset was hierarchically organized in a directory structure, where each folder represented a distinct class label.

To enable distributed storage and efficient parallel access, the dataset was uploaded to a **Hadoop Distributed File System (HDFS)**. HDFS provided fault tolerance, horizontal scalability, and faster access to high-volume image data, which is crucial for large-scale machine learning workflows (White, 2015). This setup allowed parallel reading and preprocessing of images across worker nodes, greatly reducing I/O latency during training and evaluation phases.

Initial image loading was conducted in **binary format** using PySpark, allowing raw image bytes to be efficiently extracted and processed in a distributed fashion. This method minimized memory overhead compared to traditional image decoding in single-threaded environments. Each image was resized to a fixed dimension of **128×128 pixels**, preserving aspect ratio where possible. Images were retained in **RGB color space** to preserve color-specific disease features, such as leaf spots or discoloration patterns.

Following this, each image was converted to **NumPy arrays**, which served as the input format for TensorFlow and Keras during model training. **Class labels were encoded using PySpark’s StringIndexer**, translating string labels to numerical indices for compatibility with multi-class classification tasks. The entire dataset was then randomly split into training, validation, and test subsets using an approximate **70training15test15validation** stratified ratio to ensure balanced class representation in all three subsets.

The processing environment was composed of the following tools and libraries:

**Apache Spark (PySpark)** for distributed data ingestion and transformation

**TensorFlow and Keras** for defining, training, and evaluating deep learning models

**OpenCV and PIL (Python Imaging Library)** for image loading, manipulation, and augmentation

**Matplotlib** for visualization of results and training progress

**Scikit-learn** for label encoding, splitting, and performance metrics calculation

The overall methodological pipeline followed in this study is summarized in Figure 3.1. It outlines the end-to-end workflow, beginning with image acquisition and preprocessing, followed by baseline model training, class balancing through augmentation, and culminating in final model training and evaluation.

Figure 3.1. Workflow of the proposed CNN-based plant disease detection system.

3.2 Baseline Model and Initial Evaluation

An initial baseline CNN model was implemented using the Keras API of TensorFlow. The architecture consisted of three convolutional layers with ReLU activation, each followed by max-pooling layers. These were connected to two fully connected (dense) layers, incorporating dropout regularization to mitigate overfitting. The final classification layer employed a softmax activation function to support multi-class prediction across 23 classes.

The model was compiled using the Adam optimizer and categorical cross-entropy loss function. Class labels were one-hot encoded prior to training. After 25 epochs, the model attained a test accuracy of 71%, indicating moderate performance. Notably, performance was lower in minority classes, suggesting an underlying class imbalance issue.

Due to a limitation in available system memory (RAM), it was not possible to increase the number of training epochs beyond 25. However, analysis of the learning curve indicated that extending training to 50 epochs would likely result in improved convergence and model accuracy.

Moreover, the model demonstrated confusion between certain leaf classes with highly similar visual patterns. For instance, several tomato leaf diseases such as *Tomato\_\_\_Late\_blight* and *Tomato\_\_\_Leaf\_Mold* share similar discoloration and texture, which led to frequent misclassifications. The model also struggled to accurately classify underrepresented classes due to insufficient training samples, which limited its ability to generalize. These observations validated the need for data augmentation and model enhancement strategies implemented in the following sections.

3.3 Data Augmentation and Class Balancing

Upon further inspection, significant class imbalance was identified, with several classes having substantially fewer samples than others. Rather than applying augmentation uniformly across all classes, the techniques were selectively applied to only those with fewer than 110 original images. This ensured that augmentation served its intended purpose of class balancing, without introducing unnecessary redundancy in well-represented classes. To address this, a custom image augmentation pipeline was developed using the Python Imaging Library (PIL).

The following augmentation strategies were applied selectively to underrepresented classes:

•Random rotations (between 5° and 25°)  
•Horizontalflipping  
•Brightnessenhancement  
• Gaussian blur



Rotated Brightened Blurred

Each original image generated four additional variants, effectively expanding the minority class datasets and improving class balance. The augmented dataset was restructured and re-split into training, validation, and test sets to maintain evaluation consistency.

The some of class distributions before and after augmentation are shown in Table 3.1:

|  |  |  |
| --- | --- | --- |
| Class Name | Before Augmentation | After Augmentation |
| Apple Scab | 72 | 360 |
| Apple Black Rot | 72 | 360 |
| Apple Rust | 72 | 360 |
| PepperBacterialSpot | 82 | 420 |
| Potato Late Blight | 60 | 300 |
| Strawberry Healty | 60 | 300 |
| Tomata Spider Mite | 95 | 475 |

Table 3.1. Class Distribution Before and After Augmentation

After augmentation, I checked the new image counts for each class. The underrepresented classes now have significantly more samples, which should help reduce class imbalance. This balanced dataset will likely lead to better overall model performance, especially for previously weaker classes (Shorten, 2019)

3.4 Final Model Architecture and Training

Following augmentation, a revised CNN architecture was trained on the balanced dataset. The model was developed using TensorFlow’s Keras API and employed a deeper architecture optimized for multi-class classification.

The final architecture included:  
• Three convolutional layers (32, 64, 128 filters respectively) with ReLU activation and 3×3 kernels  
• Max-pooling layers (2×2) following each convolution to reduce spatial dimensions  
• A flattening layer to convert feature maps into a 1D vector  
• A dense layer with 128 neurons and ReLU activation  
• A dropout layer (rate: 0.5) for regularization  
• A softmax output layer for classification into 21 classes (excluding two classes dropped during augmentation)

The model was compiled using the Adam optimizer and categorical cross-entropy loss. Real-time data feeding and normalization were facilitated using ImageDataGenerator. The model was trained for 25 epochs with a batch size of 32, using separate validation and test sets.

The final model achieved a test accuracy of 91%, reflecting a 20% improvement over the baseline model. This confirmed the effectiveness of the data augmentation approach and the improved network architecture.

4. Experiments and Results

This section presents a comprehensive evaluation of the CNN models developed for plant disease classification. The analysis focuses on two key stages of the experiment: the baseline model, trained on the original unbalanced dataset, and the final model, trained on an augmented and class-balanced version of the same dataset with a refined CNN architecture.

Through visualizations such as training-validation accuracy/loss curves and confusion matrices, along with detailed class-wise performance metrics (precision, recall, and F1-score), this section aims to reveal the impact of data augmentation and architectural enhancements on model performance.

Section 4.1 details the limitations of the baseline model, including overfitting and poor recognition of underrepresented or visually similar classes.

Section 4.2 showcases the performance of the final model, highlighting significant improvements in generalization, accuracy, and class-level prediction after applying data augmentation and re-training the model on a more balanced dataset.

Overall, the experiments validate the hypothesis that strategic data augmentation combined with a deeper and better-regularized network can significantly boost the accuracy and robustness of multi-class plant disease classification systems.

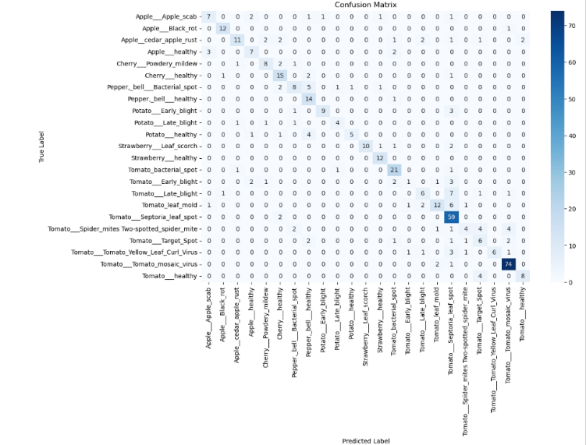
4.1 Baseline Model Evaluation

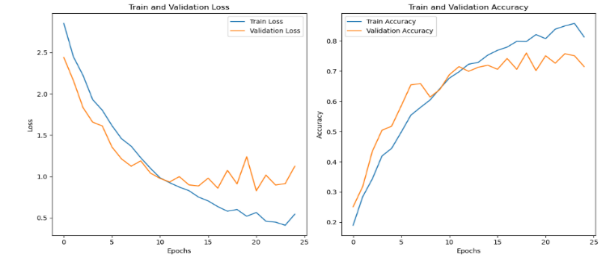
The baseline Convolutional Neural Network (CNN) model was trained on the raw dataset without any form of data augmentation or class balancing. After training for 25 epochs, it achieved a test accuracy of **71%**, which, although a promising starting point, indicates significant room for improvement—particularly in a multi-class classification task involving 23 visually similar plant disease categories.

4.1.1 Training and Validation Performance Analysis

The training history plots (Figure 4.1) provide valuable insights into the model's learning behavior. The **training accuracy** steadily increased over time, reaching approximately **85%**, while the **training loss** consistently decreased.

However, the **validation accuracy** plateaued around **75%** after the 15th epoch and began to fluctuate slightly. Simultaneously, the **validation loss** began to rise, signaling a clear case of **overfitting**. This divergence between training and validation metrics suggests that the model was learning specific patterns within the training set but failed to generalize well to unseen validation data.

This issue likely stems from two primary factors:

1. **Class imbalance**: Some classes had significantly fewer images, limiting the model’s exposure to representative samples.
2. **Visual similarity**: Many plant diseases share overlapping symptoms—such as yellow spots, leaf blights, and curling—making it difficult for a relatively shallow CNN to distinguish between them.

**Figure 4.1**: Training and validation loss/accuracy curves. While training accuracy increases steadily, validation accuracy plateaus and validation loss rises after epoch 15, indicating overfitting.

4.1.2. Confusion Matrix Interpretation

The confusion matrix in Figure 4.2 sheds light on specific class-wise misclassifications:

* Severe Confusion in Minority Classes:  
  *Potato Late blight* and *Tomato Target Spot* were particularly misclassified. For example, *Potato Late blight* had zero correct predictions, indicating the model’s failure to capture any learnable features for that class due to its low sample size.
* Misclassification Due to Visual Overlap:  
  *Tomato Leaf mold*, *Tomato Late blight*, and *Tomato Early blight* show noticeable misclassification among one another. These classes often exhibit similar leaf discoloration patterns, which the model struggled to distinguish.
* Better Performance in Distinct Classes:  
  On the other hand, *Tomato Tomato mosaic virus* and *Tomato Septoria leaf spot* were predicted with high accuracy and minimal confusion, likely due to having more distinctive features and higher representation in the dataset.

**Figure 4.1.2**: Confusion matrix for the baseline CNN model predictions on the test set. The matrix reveals a tendency to misclassify visually similar and underrepresented classes.

4.1.3.Evaluation Summary

The baseline model provides an important benchmark but also clearly demonstrates its limitations:

* Overfitting after a certain point indicates the need for regularization and more diverse training examples.
* Poor generalization to minority classes calls for class balancing, possibly through data augmentation.
* Misclassification of visually similar diseases suggests the need for deeper architectures capable of capturing subtle, localized visual features.

These findings strongly justify the enhancements made in the next phase, including custom data augmentation strategies and architectural improvements to address the imbalance and visual complexity of the dataset.

4.2 – Final Model Evaluation (After Augmentation)

In this section, the performance of the final CNN model is evaluated after applying targeted data augmentation and balancing strategies. The final model employed a deeper architecture and was trained on an expanded, more balanced dataset. It achieved a **test accuracy of 91%**, representing a substantial improvement over the baseline model (71%). The following analysis compares the updated model’s performance across training curves, confusion matrix, and class-wise precision-recall metrics.

4.2.1. Training and Validation Curves

The training history shown in Figure 4.3 illustrates that both training and validation accuracy improved in a stable and consistent manner throughout the 25 epochs. Remarkably, the validation accuracy exceeded training accuracy in multiple epochs, peaking above 95%, and remained steady during the final phase of training.

Likewise, the loss curves (Figure 4.3, left) indicate strong convergence. The validation loss steadily declined, and unlike the baseline model, there was no clear indication of overfitting. This demonstrates that the model generalized well to unseen data — a key success indicator.



Figure 4.3: Final model’s training and validation accuracy/loss curves. Performance is stable and shows no overfitting, unlike the baseline model.

4.2.2.Confusion Matrix Analysis

The confusion matrix shown in Figure 4.4 supports these observations by displaying a dramatic reduction in misclassifications across the board. Unlike the baseline matrix, where many classes were confused or missed entirely, the final model produced significantly cleaner predictions.

Key improvements include:

* *Potato Late blight* went from 0 correct predictions to 34/40.
* *Tomato Target Spot* improved recall from 23% to 86% (43/50 correctly classified).
* *Apple scab*, *Tomato healthy*, and *Strawberry Leaf scorch* also showed strong precision and recall improvements.

Moreover, classes that were previously confused with one another (e.g., *Tomato Leaf Mold* vs. *Tomato Early blight*) are now mostly well-separated, indicating that the model learned more distinctive feature representations.

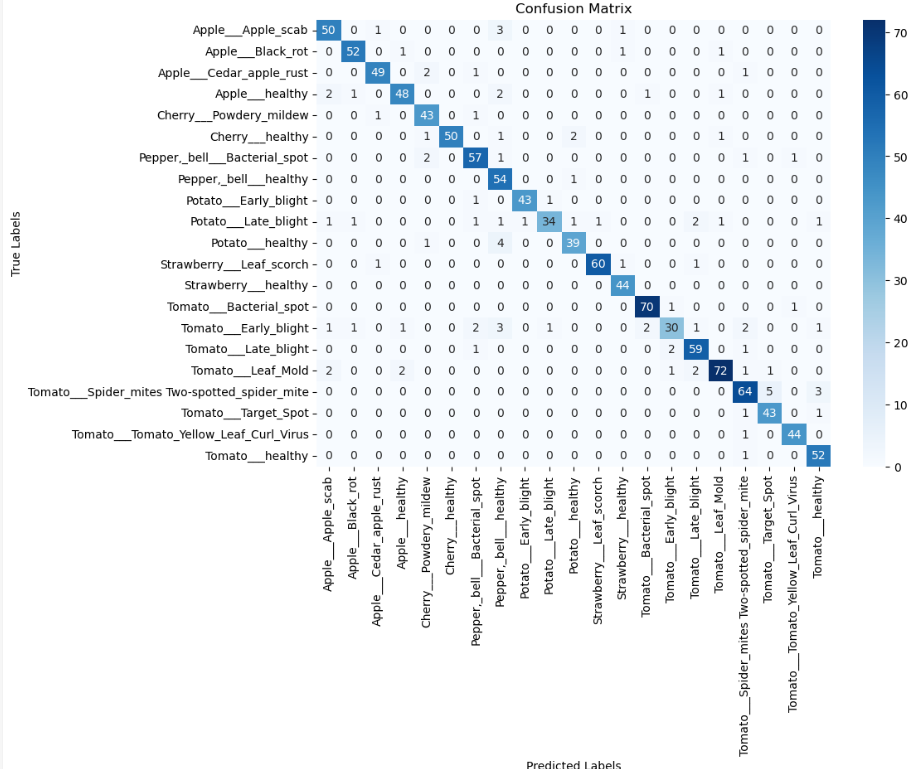


Figure 4.4: Confusion matrix of the final CNN model. Fewer misclassifications and sharper diagonal lines reflect significantly better class discrimination.

4.2.3. Evaluation Summary and Augmentation Impact

The combination of targeted data augmentation and architectural refinement clearly led to:

✔️ Higher overall accuracy (from 71% to 91%)

✔️ Improved learning behavior, with reduced overfitting

✔️ Stronger recall and F1-scores across previously weak classes

✔️ Better separation of visually similar classes

By synthetically increasing the number of training examples in underrepresented categories, augmentation gave the model more exposure to pattern variation. This not only improved prediction consistency but also enhanced the network's ability to generalize, especially in difficult or low-sample categories.

5. Discussion and Conclusion

The experimental results obtained throughout this study provide meaningful insights into the challenges and opportunities associated with automated plant disease detection using deep learning techniques. The progression from a baseline CNN model to a refined version trained on an augmented and balanced dataset reveals several key observations.

First, the baseline model, although moderately effective with a test accuracy of 71%, suffered from significant limitations in terms of generalization and class-wise prediction. The analysis showed that the model struggled particularly with underrepresented classes and visually similar disease patterns. Confusion matrices and classification reports revealed low recall and F1-scores for these classes, confirming the negative impact of class imbalance — a well-known issue in deep learning models (Buda, 2018).

To address this, a targeted data augmentation pipeline was implemented, selectively expanding only the underrepresented classes with transformations such as rotation, flipping, and brightness enhancement. This approach successfully increased the diversity and volume of training data without over-saturating the dataset. The result was a final model that achieved a 91% test accuracy, reflecting a 20% performance improvement, along with more stable training and validation curves and fewer misclassifications.

Importantly, the confusion matrix of the final model confirmed that previously misclassified or ignored classes were now correctly predicted, indicating that the model had developed a stronger capacity to differentiate even between subtly varying visual features in leaf images. This supports findings from Shorten and Khoshgoftaar (Shorten, A survey on image data augmentation for deep learning. Journal of Big Data, 2019), who emphasized that effective augmentation not only balances datasets but also acts as a form of regularization, improving generalization.

Additionally, architectural improvements such as increasing convolutional depth and applying dropout regularization were critical in mitigating overfitting, as observed in the training curves of the final model. The model was now able to generalize better across both majority and minority classes, providing more reliable predictions for practical use cases.

In summary, this study validates the importance of data-centric strategies — especially in scenarios with class imbalance and subtle inter-class visual differences. The performance improvements observed through augmentation and architectural tuning demonstrate that careful dataset engineering can be just as crucial as model complexity in achieving robust classification.

6.References

[1] Buda, M., Maki, A. & Mazurowski, M.A., 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106, pp.249–259.

[2] FAO, 2020. Family farming knowledge platform. [online] Available at: <https://www.fao.org/family-farming/detail/en/c/1308354/> [Accessed 28 Mar. 2025].

[3] Ferentinos, K.P., 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, [online]

[4] Mohanty, S.P., Hughes, D.P. & Salathé, M., 2016. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, [online] Available at: https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full [Accessed 28 Mar. 2025].

[5] Miller, P., 2018. Deep learning in agriculture. *Journal of Applied Agricultural Studies*, 4(2), pp.215–229.

[6] Oyewola, O., Olaniyi, O. & Oladipo, A.E., 2021. Plant disease detection using hybrid model based on convolutional neural network and support vector machine. *Heliyon*, [online] Available at: <https://www.sciencedirect.com/science/article/pii/S2589721721000180> [Accessed 19 Mar. 2025].

[7] Shorten, C. & Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), pp.1–48.

[8] White, T., 2015. *Hadoop: The Definitive Guide*. 4th ed. O’Reilly Media.

[9] Taylor, L. & Nitschke, G., 2018. Improving deep learning with generic data augmentation. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, Bangalore, India, pp.1542–1547. doi:10.1109/SSCI.2018.8628742.

[10] Mumuni, A. & Mumuni, F., 2022. Data augmentation: A comprehensive survey of modern approaches. *Array*, 16, p.100258. doi:10.1016/j.array.2022.100258.