

# Disorder Severity Classification in Tomato Based on Weight Cluster Loss and Convolutional Neural Network

**Babatunde Sunday**

sundaybabatunde53@gmail.com

Ahmadu Bello University

**Sahabi A. YUSUF**

Ahmadu Bello University

**Mohammed Abdullahi**

Ahmadu Bello University

**Yazeed Abdullahi Masha**

Ahmadu Bello University

**Alice Sikemi Matemilola**

Ahmadu Bello University

---

## Research Article

**Keywords:** Tomato Disorder Severity Classification, Convolutional Neural Networks (CNN), Inter-Class Similarities, Overfitting, Weight Cluster Loss

**Posted Date:** May 20th, 2025

**DOI:** <https://doi.org/10.21203/rs.3.rs-6641648/v1>

**License:**   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

**Additional Declarations:** No competing interests reported.

---

# Disorder Severity Classification in Tomato Based on Weight Cluster Loss and Convolutional Neural Network

Babatunde Sunday<sup>1\*</sup>, Sahabi A. YUSUF<sup>2</sup>, Mohammed Abdullahi<sup>1,2,+</sup>, Yazeed Abdullahi Masha<sup>1,2,3,+</sup>, and Alice Sikemi Matemilola<sup>3,+</sup>

<sup>1</sup> Computer Science Department, Ahmadu Bello University, Zaria-Nigeria

<sup>2</sup> Computer Science Department, Ahmadu Bello University, Zaria-Nigeria

<sup>1,2,+</sup> Computer Science Department, Ahmadu Bello University, Zaria-Nigeria

<sup>1,2,3,+</sup> Computer Science Department, Ahmadu Bello University, Zaria-Nigeria

<sup>3,+</sup> Computer Science Department, Ahmadu Bello University, Zaria-Nigeria

\* sundaybabatunde53@gmail.com

<sup>2</sup> sahabiali@yahoo.com

<sup>1,2,+</sup> abdullahilwafu@abu.edu.ng

<sup>1,2,3,+</sup> yazy4real@gmail.com

<sup>3,+</sup> matemilolao@gmail.com

## ABSTRACT

Agricultural productivity is a crucial determinant of economic stability. Within the agricultural sector, particularly in tomato production, the impact of plant diseases and pests poses a significant challenge. Detecting the severity of disorders in tomato plants is essential for addressing these challenges. Achieving accurate and rapid detection is imperative for developing early treatment strategies, ultimately minimizing economic losses. While various researchers have explored solutions using convolutional neural network (CNN) models to identify and classify disease severity in tomatoes, the limited availability of training data has led to overfitting issues and inter-class similarity, resulting in suboptimal performance measures. To address the overfitting problem arising from insufficient data, this research proposes a deep transfer-based framework. Three CNN models i.e., AlexNet, SqueezeNet, and InceptionV3 are employed to classify disease severity in tomato plants, specifically targeting tomato late blight, tomato early blight, tomato leaf mold, tomato bacteria spot, and healthy tomato leaves using the PlantVillage dataset. The study incorporates a weighted-cluster loss function to mitigate inter-class similarities. Computational accuracy serves as the performance metric. Following experimentation, InceptionV3 demonstrated the highest classification accuracy at 93.66%, surpassing AlexNet (83.03%) and SqueezeNet (80.09%). Consequently, the proposed system functions as a decision support tool for farmers, aiding in the identification of disorder severity in tomato plant leaves.

**Keywords:** Tomato Disorder Severity Classification; Convolutional Neural Networks (CNN); Inter-Class Similarities; Overfitting; Weight Cluster Loss

## INTRODUCTION

Plants play a pivotal role in sustaining human life, with crop plant production, fishing, livestock, and forestry comprising the primary agricultural sectors in Nigeria [13]. Crop plant production dominates, contributing

approximately 87.6% to the sector's total output, while livestock, fishing, and forestry follow with 8.1%, 3.2%, and 1.1%, respectively [13]. Nevertheless, crop plant production has been reported to be most frequently susceptible to diseases and pests due to factors such as high-density cultivation, monoculture practices, genetic uniformity, intensive farming, global trade, climate change, and the lack of natural predators [8]. In addressing the disease and pest challenges of crop production, smart farming emerges as a critical solution [19, 14].

Crop plant diseases have continued to remain at the forefront of the challenges to effective production (Kaur et al., 2023). Specifically, the tomato plant has been studied over the years to be among the crop plants most susceptible to various diseases, posing threats to farm yields [6]. According to the Food and Agricultural Organization of the United Nations Statistics (FAOSTAT), global tomato production in 2017 reached 170.8 million tons, emphasizing the importance of this crop [2].

Traditionally, the early detection of disorders in tomato plants relied on field observation through unaided human eyes [2, 11]. This method heavily depended on the agricultural expertise accumulated over years of farming, consultation of agricultural reference materials, acquisition of images of disease specimens, and recourse to internet searches for information, or the involvement of experts for a comprehensive review of disease symptoms [2]. The inherent challenges of this approach, compounded by the multitude of cultivated plants and their associated psychopathological issues, make conclusions and treatment strategies unreliable, inconsistent, and prone to errors [10].

Numerous initiatives employing Artificial Intelligence (AI) have been recently undertaken as advancement from the use of these traditional methods [1]. These AI techniques aid farmers in accurately identifying and classifying pests and diseases, along with assessing the severity of symptoms [5]. For example, [9] highlight the utility of Computer-Aided Diagnosis (CAD) systems, enabling farmers to access expert knowledge inexpensively through computers or smartphones. [9] also underscore the remarkable growth in AI applications, particularly in Machine Learning (ML) and the emergence of Deep Learning (DL) models.

Lately, the progress in DL has opened avenues for a contemporary approach to precisely identify and assess the severity of tomato diseases. Within the realm of DL, object detection holds significant importance [14]. Object detection aims to precisely locate the region of interest in an image and determine the respective category to which each object belongs [16]. Identifying disorders in tomato plants stands as a primary challenge, demanding not only expertise in recognizing visual symptoms but also accurate classification of the severity of the tomato plant disease [6].

DL methods, as reported by [14], exhibit state-of-the-art performance compared to traditional ML approaches such as the Support Vector Machines (SVM), across various domains, including agriculture. François-Lavet et al., (2018) as well, emphasize the excellence of Convolutional Neural Networks (CNN) in image recognition tasks within the realm of DL, automatically learning features directly from input images, unlike traditional ML approaches that rely on handcrafted features based on prior knowledge. [7] describe a deep learning-based disease severity system, wherein a CNN model extracts high-level features directly from input images, followed by an output layer, typically utilizing softmax activation, to differentiate identified disease severities. For challenging real-world applications, [17] advocate for the use of CNN models for efficiency and effectiveness.

Additionally, the use of Transfer Learning (TL) methods such as ResNet, and InceptionV2 has become renowned in the detection and classification of tomato disease severity (Kumar et al., 2023). For example, [8] focused on developing an effective computer-aided disease detection system for tomato plants using a deep learning-based CNN technique, specifically the Modified InceptionResNet-V2 (MIR-V2). Their approach involves a TL method, training the model on both a public dataset and a self-collected dataset that includes

seven different tomato leaf disease classifications. Also, [3] developed a pipeline for automatic identification of tomato leaf diseases using three compact CNNs. The approach incorporates TL, feature merging, and a hybrid feature selection method, achieving high accuracy with a reduced feature set, demonstrating competitiveness compared to previous studies on tomato leaf disease classification.

Despite the strides made by various researchers in utilizing CNN models for identifying and classifying disease severity in tomatoes, challenges persist due to limited training data, resulting in overfitting and inter-class similarity issues [16]. These limitations result in comparatively low-performance measures for tomato disease severity classification. To mitigate these challenges, this research introduces a deep transfer-based framework employing three TL models i.e., AlexNet, SqueezeNet, and InceptionV. Additionally, a loss function i.e., weighted-cluster loss, which is used during the fine-tuning phase for tomato disorder severity classification is employed. In particular, the TL techniques aim to tackle the data scarcity issue, while the weighted-cluster loss function is designed to handle inter-class similarities by learning a distinct class center for each disease class. Generally, we focus on classifying disease severity in tomato plants, targeting late blight, early blight, leaf mold, bacteria spots, and healthy tomato leaves. In pursuit of our overarching objective, we specifically undertake the following actions:

1. Develop a comprehensive deep learning transfer-based framework to mitigate the challenges arising from overfitting due to a limited training dataset.
2. Incorporate the weighted-cluster loss function into the three TL models i.e., AlexNet, SqueezeNet, and InceptionV3, to effectively tackle issues related to inter-class similarities.
3. Execute the implementation of the enhanced TL models and evaluate their performance using benchmark datasets sourced from PlantVillage. Conduct a thorough assessment, focusing on classification accuracy, and compare the outcomes with recent existing work in the field to ascertain the effectiveness and advancements achieved by the proposed framework.

## RESULTS

For all experiments conducted, the Google Colab platform with Python was utilized. The experiments were carried out on a computer featuring an Intel(R) Core (TM) i5-3360M processor operating at a frequency of 2.80GHz and equipped with 8GB of RAM.

In this research, the average classification accuracy served as the performance evaluation metric. The proposed method's performance was directly compared with that of [16] by employing the three CNN models on the PlantVillage tomato leaf dataset. Notably, superior results are highlighted through bolding and underlining.

Considering the proposed method, Table 1 reveals that among the three CNN architectures, InceptionV3 outperformed both AlexNet and SqueezeNet, securing an impressive average accuracy score of 93.66%, while AlexNet and SqueezeNet achieved lower average accuracies of 83.03% and 80.19%, respectively. The InceptionV3 model is anticipated to outperform AlexNet and SqueezeNet due to its deeper and more complex architecture, allowing it to capture intricate features in the data. InceptionV3's utilization of inception modules, which incorporate multiple convolutional operations at different scales, enables it to effectively extract hierarchical features. This enhanced depth and feature extraction capability make InceptionV3 more adept at discerning subtle patterns and nuances within the PlantVillage dataset, potentially leading to superior

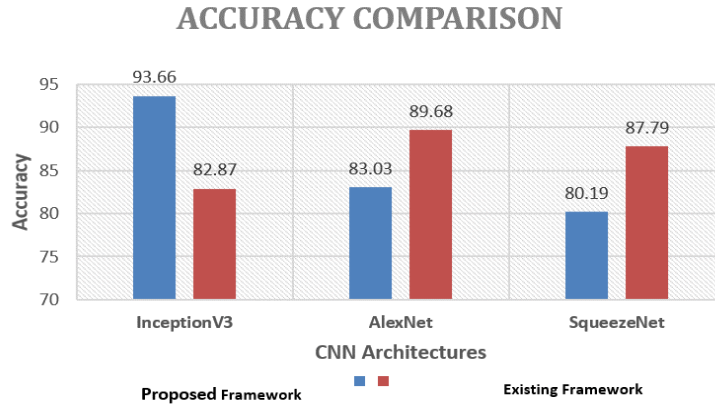
performance in disease severity assessment compared to the relatively shallower architectures of AlexNet and SqueezeNet.

The anticipated superiority of InceptionV3 over AlexNet and SqueezeNet is further supported by the incorporation of the Weighted Cluster Loss. The Weighted Cluster Loss introduces a novel approach to handling intra-class variation and inter-class similarity in the PlantVillage dataset, potentially addressing the challenges posed by imbalanced datasets. This additional loss function, combined with InceptionV3's intricate architecture, contributes to a more robust and discriminative learning process, enhancing the model's ability to accurately assess disease severity. As a result, InceptionV3, equipped with the Weighted Cluster Loss, is expected to surpass AlexNet and SqueezeNet in delivering precise and reliable evaluations of tomato leaf disease severity.

**Table 1:** Performance comparison based on average classification accuracy

| Methods              | CNN Architectures   |         |            |
|----------------------|---------------------|---------|------------|
|                      | InceptionV3         | AlexNet | SqueezeNet |
|                      | <b>Accuracy (%)</b> |         |            |
| Proposed Methods     | <b>93.66</b>        | 83.03   | 80.19      |
| (Verma et al., 2020) | 82.87               | 89.69   | 87.79      |

In the context of the CNN architectures examined in the study by Verma et al., AlexNet exhibited superior performance, attaining an average accuracy of 89.69%, outperforming InceptionV3 and SqueezeNet with scores of 82.87% and 87.79%, respectively. However, when considering the overall performance, encompassing both the work of Verma et al. and the proposed method, InceptionV3 from the proposed method demonstrated superior performance compared to its two competitors and the study by Verma et al., as visually depicted in Figure 1. Training, validation, and loss graphs best model using PlantVillage dataset is presened in Figure 2



**Figure 1:** Performance comparison based on accuracy between CNN architectures of Verma et al., (2020) and the proposed method.

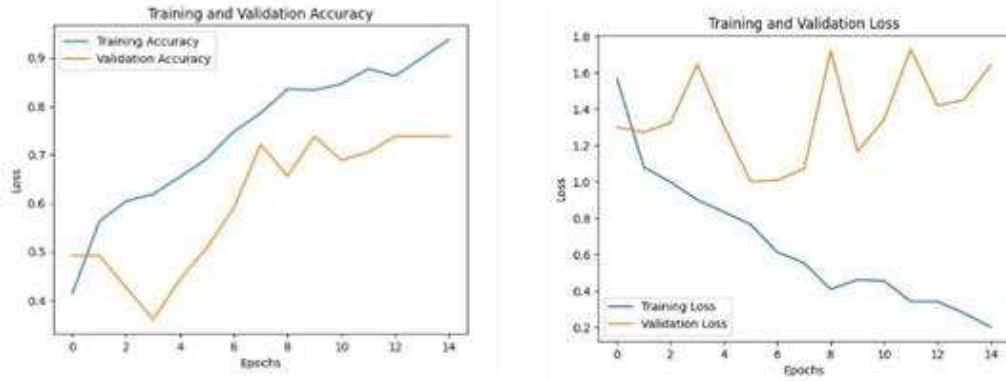


Figure 2: Training, validation, and loss graphs best model using PlantVillage dataset

The confusion matrix is a valuable tool for assessing the performance of a classification model. It provides a detailed breakdown of the model's predictions, allowing for the analysis of true positive, true negative, false positive, and false negative outcomes. This information is particularly useful for understanding the accuracy, of the model, which is a crucial metric in evaluating the effectiveness of disease severity classification.

Considering the proposed tomato severity classification, the confusion matrix can identify how well the model distinguishes between different severity levels. It allows researchers to assess if the model tends to misclassify certain severity levels and where the errors are concentrated. This insight aids in refining the model, adjusting hyperparameters, or addressing potential biases in the training data, ultimately improving the accuracy and reliability of severity predictions for different classes.

Table 2 illustrates the confusion matrix for the five classes of tomato disease, utilizing the proposed AlexNet architecture. Notably, in the five-class scenario encompassing one healthy class and four distinct unhealthy classes, the model accurately predicted 86% of healthy instances, with only 14% of healthy tomato leaf images misclassified as unhealthy.

**Table 2:** Confusion Matrix for the healthy and four unhealthy classes based on AlexNet architecture

|                              | Predicted class              |                            |                           |                         |                       |
|------------------------------|------------------------------|----------------------------|---------------------------|-------------------------|-----------------------|
|                              | <i>Tomato_Bacterial_spot</i> | <i>Tomato_Early_blight</i> | <i>Tomato_Late_blight</i> | <i>Tomato_Leaf_Mold</i> | <i>Tomato_healthy</i> |
| <i>Tomato_Bacterial_spot</i> | <b>0.88</b>                  | 0.00                       | 0.10                      | 0.00                    | 0.02                  |
| <i>Tomato_Early_blight</i>   | 0.02                         | <b>0.91</b>                | 0.00                      | 0.07                    | 0.00                  |
| <i>Tomato_Late_blight</i>    | 0.06                         | 0.00                       | <b>0.89</b>               | 0.00                    | 0.05                  |
| <i>Tomato_Leaf_Mold</i>      | 0.03                         | 0.07                       | 0.07                      | <b>0.83</b>             | 0.00                  |
| <i>Tomato_healthy</i>        | 0.06                         | 0.05                       | 0.00                      | 0.03                    | <b>0.86</b>           |

Examining the SqueezeNet architecture, Table 3 displays the confusion matrix for the healthy class and four unhealthy classes. In the context of the five-class issue involving one healthy class and four distinct unhealthy

classes, the model accurately predicted 91% of healthy instances, with only 9% of healthy tomato leaf images misclassified as unhealthy.

**Table 3:** Confusion Matrix for the healthy and four unhealthy classes based on SqueezeNet architecture

|            |                              | Predicted class              |                            |                           |                         |                       |
|------------|------------------------------|------------------------------|----------------------------|---------------------------|-------------------------|-----------------------|
|            |                              | <i>Tomato_Bacterial_spot</i> | <i>Tomato_Early_blight</i> | <i>Tomato_Late_blight</i> | <i>Tomato_Leaf_Mold</i> | <i>Tomato_healthy</i> |
| True class | <i>Tomato_Bacterial_spot</i> | <b>0.89</b>                  | 0.00                       | 0.06                      | 0.00                    | 0.05                  |
|            | <i>Tomato_Early_blight</i>   | 0.00                         | <b>0.92</b>                | 0.04                      | 0.04                    | 0.00                  |
|            | <i>Tomato_Late_blight</i>    | 0.02                         | 0.08                       | <b>0.90</b>               | 0.00                    | 0.05                  |
|            | <i>Tomato_Leaf_Mold</i>      | 0.03                         | 0.00                       | 0.07                      | <b>0.90</b>             | 0.00                  |
|            | <i>Tomato_healthy</i>        | 0.00                         | 0.06                       | 0.01                      | 0.02                    | <b>0.91</b>           |

Analyzing the InceptionV3 architecture, Table 4 showcases the confusion matrix for the healthy class and four unhealthy classes. Within the framework of the five-class scenario, encompassing one healthy class and four distinct unhealthy classes, the model achieved a precise prediction of 98% for healthy instances, with only a 2% misclassification rate for healthy tomato leaf images as unhealthy.

**Table 4:** Confusion Matrix for the healthy and four unhealthy classes based on InceptionV3 architecture

|            |                              | Predicted class              |                            |                           |                         |                       |
|------------|------------------------------|------------------------------|----------------------------|---------------------------|-------------------------|-----------------------|
|            |                              | <i>Tomato_Bacterial_spot</i> | <i>Tomato_Early_blight</i> | <i>Tomato_Late_blight</i> | <i>Tomato_Leaf_Mold</i> | <i>Tomato_healthy</i> |
| True class | <i>Tomato_Bacterial_spot</i> | <b>0.99</b>                  | 0.00                       | 0.01                      | 0.00                    | 0.00                  |
|            | <i>Tomato_Early_blight</i>   | 0.01                         | <b>0.97</b>                | 0.00                      | 0.02                    | 0.00                  |
|            | <i>Tomato_Late_blight</i>    | 0.00                         | 0.00                       | <b>0.10</b>               | 0.00                    | 0.00                  |
|            | <i>Tomato_Leaf_Mold</i>      | 0.00                         | 0.00                       | 0.00                      | <b>0.10</b>             | 0.00                  |
|            | <i>Tomato_healthy</i>        | 0.00                         | 0.02                       | 0.00                      | 0.00                    | <b>0.98</b>           |

Based on the confusion matrices presented for the AlexNet, SqueezeNet, and InceptionV3 architectures, it is evident that each model exhibits varying degrees of accuracy in predicting healthy and unhealthy instances. The AlexNet architecture demonstrates robust performance with an 86% accuracy in identifying healthy instances, while SqueezeNet achieves a higher accuracy of 91%. However, InceptionV3 surpasses both, attaining an impressive accuracy of 98% in correctly predicting healthy instances.

In the context of misclassifications, the AlexNet model exhibits a 14% misclassification rate for healthy tomato leaf images as unhealthy, indicating some challenges in distinguishing between the classes. SqueezeNet performs slightly better, with a 9% misclassification rate. Remarkably, InceptionV3 demonstrates superior precision, with only a 2% misclassification rate for healthy tomato leaf images.

In conclusion, InceptionV3 emerges as the most effective architecture in accurately predicting healthy instances and minimizing misclassifications. The choice of the CNN architecture significantly influences the model's performance, and InceptionV3 stands out as a promising candidate for plant disease severity classification, showcasing its potential for practical applications in agriculture.

## **Methods**

This section delineated the research methodology, outlining specific techniques and the overarching model architecture employed for the proposed disorder severity classification method. Additionally, it provided comprehensive details on the dataset characteristics, evaluation measures, and the fine-tuning process implemented for the model.

### **Proposed Method Architecture**

Within this section, we present the proposed deep transfer-learning framework and subsequently introduce the weighted-cluster loss. This study employs a publicly available pre-trained base model, utilizing transfer learning, rather than conducting direct pre-training. Specifically, the pre-trained models of AlexNet, SqueezeNet, and InceptionV3, accessible on the official PlantVillage database website, are utilized. These models were initially trained on the PlantVillage dataset, encompassing 6264 diseases, resulting in the acquisition of rich feature representations for disease factors by the network.

The pre-trained AlexNet, SqueezeNet, and InceptionV3 models, initially trained on the PlantVillage dataset for disease identification, undergo fine-tuning for disease recognition through the integration of the weighted-cluster loss. Preceding the fine-tuning stage, an additional fully connected layer is introduced to the model, with the first three stages of the pre-trained model kept frozen to optimize computational resources. The update of model parameters is facilitated through the application of the weighted-cluster loss at the output layer. The visual representation of the proposed method's architecture is depicted in Figure 3.

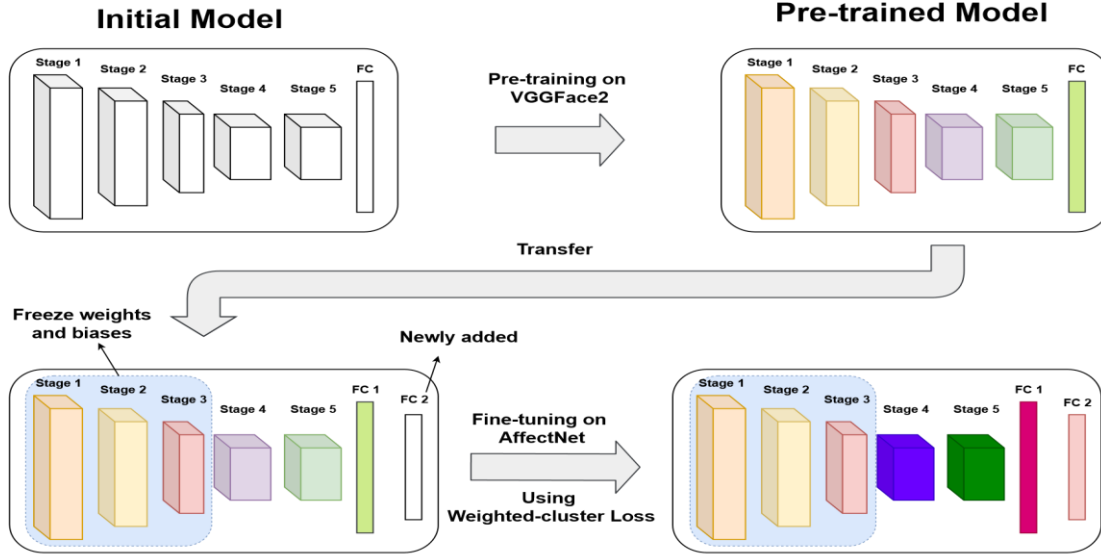


Figure 3: Overall architecture of the proposed method

### 1. Base Models

The investigation harnesses the computational prowess of three fundamental CNN architectures i.e., InceptionV3, ResNet, and SqueezeNet [16]. These models, pre-trained on the intricate PlantVillage dataset, serve as the bedrock for disease identification. Their nuanced network structures contribute to the overall resilience and efficacy of the proposed methodology, highlighting the technical finesse and adaptability of transfer learning in the realm of plant disease severity assessment.

### 2. Fine Tuning Process

In this phase, the study introduces an additional fully connected layer to the pre-trained models of AlexNet, SqueezeNet, and InceptionV3 (positioned before the output layer, as illustrated in Table 5). Furthermore, the output layer is modified to accommodate an eight-class classification. Concurrently, the weights and biases of Stages 1 to 3 in the pre-trained model are frozen, a common practice in transfer learning-based systems when computational resources are constrained.

**Table 5:** Detailed architectures of AlexNet, SqueezeNet, and InceptionV3 in the fine-tuning phase.

|                  | Output Size         | Kernel                       | Repeat |
|------------------|---------------------|------------------------------|--------|
| Stage 1 (Freeze) | $112 \times 112$    | Convolution ( $7 \times 7$ ) | 1      |
| Stage 2 (Freeze) | $56 \times 56$      | Convolution block            | 3      |
| Stage 3 (Freeze) | $28 \times 28$      | Convolution block            | 4      |
| Stage 4          | $14 \times 14$      | Convolution block            | 6      |
| Stage 5          | $7 \times 7$        | Convolution block            | 3      |
|                  | $1 \times 1$        | global average pooling       |        |
| Fully connected  | $(2048 \times 512)$ | Fully Connected              |        |
| Fully connected  | $(512 \times 5)$    | Fully Connected              |        |
| Output layer     |                     | 5-D Softmax                  |        |

Convolutional blocks (AlexNet, SqueezeNet, and InceptionV3 blocks) are depicted in parentheses, indicating the stacked numbers of these blocks. Table 6 further presents the details of the total number of parameters, layers, and sizes for each of the three CNN architectures after construction.

**Table 6:** Network specifications of the three CNN architectures

|            | Parameters (millions) | Size(MB) | Layers | Depth |
|------------|-----------------------|----------|--------|-------|
| SqueezeNet | 1.24                  | 4.6      | 68     | 18    |
| AlexNet    | 61                    | 227      | 25     | 8     |
| Inception  | 23.9                  | 89       | 316    | 48    |

### *Weighted Cluster Loss*

To overcome the constraints associated with existing loss functions, such as weighted-softmax loss [11] and center loss, this study adopts a loss function termed weighted-cluster loss [12]. The weighted cluster loss incorporates an additional term into the center loss, aiming to separate the centers of each class concurrently. This integration is anticipated to enable the model to effectively manage both high intra-variation and inter-class similarity within the PlantVillage dataset. The formulation of the weighted-cluster loss for the  $i^{th}$  training sample is expressed in Equation 1.

$$L_{weighted\_cluster} = \frac{1}{2} \sum_{i=1}^m W_{yi} \frac{\|x_i - c_{yi}\|_2^2}{(\sum_{j=1, j \neq i}^k \|c_j - c_{yi}\|_2^2) + \alpha} \quad (1)$$

where  $W_{yi}$  denotes the weights assigned to the class corresponding to the label  $y_i$ ,  $k$  indicates the number of emotion classes, and the constant  $\alpha$  is introduced to avoid the denominator reaching 0. In this study, we adopt a default value of  $\alpha = 1$  according to [12].

In Equation 1, the numerator imposes a penalty on the distance between the deep feature of the training sample (extracted from the last fully connected layer before the output layer) and its corresponding center. Meanwhile, the denominator penalizes the distance between the said center and all other class centers. Minimizing the

weighted-cluster loss results in the compaction of deep features for training samples belonging to the same class (i.e., forming a cluster) in the feature space, while simultaneously enlarging the distance between clusters of different classes. Additionally, the use of weight  $W_{yi}$  addresses the issue of imbalanced datasets by imposing less penalty on models for misclassifying samples from majority classes and a heavier penalty for misclassifying samples from minority classes. The overall loss function of PlantVillage training is given in Equation 2.

$$L = L_{\text{weighted-softmax}} + \lambda L_{\text{weighted-cluster}}$$

$$= - \sum_{i=1}^m W_{yi} \left( \log(\hat{p}_i) - \frac{\lambda}{2} \frac{\|X_i - c_{yi}\|_2^2}{\sum_{j=1, j \neq i}^k \|C_j - c_{yi}\|_2^2 + \alpha} \right) \quad (2)$$

where the parameter  $\lambda$  is employed to balance the two losses. The  $W_{yi}$  is further defined as shown in Equation 10:

$$W_{yi} = \frac{N_{\min}}{N_{yi}} \quad (3)$$

where  $N_{\min}$  represents the number of samples from the smallest class (disgust), and  $N_{yi}$  denotes the number of samples from the class corresponding to the label  $y_i$ . The combined use of joint weighted-cluster loss and weighted-softmax loss is directly applicable for training deep neural networks. The iteration-based update of class centers,  $c_{yi}$ , occurs during the training process, specifically within mini-batches, with a scalar  $\gamma$  controlling the learning rate of class centers (set to  $\gamma = 1$  in our experiments). The partial derivative of the weighted-cluster loss,  $L_{\text{weighted-cluster}}$ , concerning the sample's features,  $X_i$ , can be computed as in Equation 4:

$$\frac{\partial L_{\text{weighted-cluster}}}{\partial X_i} = W_{yi} \frac{X_i - c_{yi}}{(\sum_{l=1, l \neq i}^k \|C_l - c_{yi}\|_2^2) + \alpha} \quad (4)$$

The computation of the  $j^{\text{th}}$  class center's update can be determined using Equation 5.

$$\Delta c_j^t = \sum_{i=1}^m W_{yi} \frac{\delta(y_i=j)(X_i - c_j)}{(\sum_{l=1, l \neq i}^k \|C_l - c_{yi}\|_2^2) + \alpha} \quad (5)$$

where  $\delta(y_i = j) = 1$  if  $y_i = j$ , and  $\delta(y_i = j) = 0$  if  $y_i \neq j$ .

Subsequently, the class centers can be modified within each mini-batch utilizing a learning rate denoted as  $\gamma$ :

$$c_j^{t+1} = c_j^t - \gamma \Delta c_j^t \quad (6)$$

Algorithm 1: Outlines how the disorder severity model learns using the weighted cluster loss functions

---

**Algorithm 1** Learning algorithm of the disorder severity method with the weight cluster loss functions

---

**Input:** Training dataset  $\{x_i\}$ , mini-batch size  $m$ , total iterations  $T$ , learning rates  $\mu$  and  $\gamma$ , and hyper-parameters  $\lambda$

- 1: **Initialize:** Parameters of network layers  $W$ , parameters of weighted-softmax loss  $\theta$ , and parameters of weighted-cluster loss (i.e., centers)  $\{c_j | j = 1, 2, \dots, n\}$ .
- 2: **for**  $t = 1$  to  $T$  **do**
- 3:     Calculate the combined loss
- 4:      $L^t = L_{\text{Weighted-softmax}}^t + \lambda L_{\text{Weighted-cluster}}^t$
- 5:     Calculate for each  $i$  the backpropagation error
- 6:      $\frac{\partial L^t}{\partial x_i^t} = \frac{\partial L_{\text{Weighted-softmax}}^t}{\partial x_i^t} + \lambda \frac{\partial L_{\text{Weighted-cluster}}^t}{\partial x_i^t}$
- 7:     Update the weighted-softmax loss parameters  $\theta$
- 8:      $\theta^{t+1} = \theta^t - \mu \frac{\partial L_{\text{Weighted-softmax}}^t}{\partial \theta^t}$
- 9:     Update the weighted-cluster loss parameter  $c_j$  for each  $j$  as shown in Equation 12.
- 10:      $c_j^{t+1} = c_j^t - \gamma \Delta c_j^t$
- 11:     Update the network parameters  $W$
- 12:      $W^{t+1} = W^t - \mu \frac{\partial L^t}{\partial x_i^t} \frac{\partial x_i^t}{\partial W^t}$
- 13: **end for**

**Output:** Network layer parameters  $W$

---

### PlantVillage Tomato Leaf Dataset

To attain the proposed disorder severity method’s objective, data was sourced from the PlantVillage [15] repository. The PlantVillage tomato leaf dataset is a valuable resource in the field of plant pathology and computer vision for agriculture. It comprises a diverse collection of high-resolution images of tomato leaves affected by various diseases and disorders. These images are meticulously annotated to provide labels indicating the specific disease or condition present in each sample. The dataset encompasses a wide range of tomato plant issues, including fungal infections, bacterial diseases, viral infections, and non-infectious disorders [15].

Researchers leverage the PlantVillage tomato leaf dataset to develop and train convolutional neural networks, for automated disease identification and severity estimation in tomato plants [4, 5]. The dataset’s rich diversity enables the creation of robust models capable of recognizing and categorizing different diseases accurately. The assessment of the proposed method involved the utilization of the PlantVillage tomato leaf dataset. A comprehensive overview of the dataset is provided in Table 7.

**Table 7:** PlantVillage tomato leaf dataset classes and image counts

| <b>Classes</b>          | <b>Images count</b> |
|-------------------------|---------------------|
| Tomato Healthy          | 1591                |
| Tomato Early Blight     | 1000                |
| Tomato Late Blight Mold | 1910                |
| Tomato Tomato Leaf Mold | 952                 |
| Tomato Bacterial Spot   | 811                 |

### Evaluation Measure

To assess how well the proposed method performs, we utilized the accuracy evaluation metric as a key measure of its effectiveness. There are various evaluation metrics in the literature to measure the discriminative performance of the proposed method; however, accuracy was chosen due to its widely acknowledged significance in providing an overall assessment of the model's ability to correctly classify instances across all classes [12]. This metric is particularly valuable for its simplicity and ease of interpretation, making it a suitable choice for evaluating the overall effectiveness of the proposed approach [4]. It is represented thus:

$$\text{Model Accuracy} = \frac{\text{Counts of correct predictions}}{\text{Total count of predictions}} \quad (7)$$

### Hyperparameters Settings

During the model training process, the research employed the categorical cross-entropy loss function with SoftMax for multiclass datasets. The Rectified Linear Unit (ReLU) was utilized for back-propagation. The Stochastic Gradient Descent with Momentum (SGDM) was chosen for learning rate tuning. Additionally, dropout was implemented at network connections to mitigate the risk of overfitting by randomly deactivating neurons in the network. The selection of parameter settings and their respective values was conducted following the methodology outlined in the study by (Verma et al., 2020) to ensure a fair and comparable evaluation. Table 8 presents the parameters used and their corresponding values.

**Table 8:** Parameters settings and values used

|                  | <b>AlexNet</b> | <b>InceptionV3</b> | <b>SqueezeNet</b> |
|------------------|----------------|--------------------|-------------------|
| Optimizer Type   | SGDM           | SGDM               | SGDM              |
| Learning Rate    | 0.0010         | 0.0100             | 0.0001            |
| Momentum         | 0.9000         | 0.9000             | 0.9000            |
| Mini-Batch Size  | 20             | 20                 | 20                |
| L2Regularization | 1.0000e-04     | 1.0000e-04         | 1.0000e-04        |
| Max Epochs       | 15             | 15                 | 15                |

## REFERENCES

- [1] Abbasi, R., Martinez, P., & Ahmad, R. (2023). Crop diagnostic system: A robust disease detection and management system for leafy green crops grown in an aquaponics facility. *Artificial Intelligence in Agriculture*, 10, 1–12. <https://doi.org/10.1016/J.AIIA.2023.09.001>
- [2] Akano, O. I., Oderinde, F. O., & Omotayo, A. O. (2023). Agricultural yield, food nutrition and dietary energy supply in Nigeria: Evidence from nationally representative data. *Journal of Agriculture and Food Research*, 11(February), 100525. <https://doi.org/10.1016/j.jafr.2023.100525>
- [3] Attallah, O. (2023). Tomato Leaf Disease Classification via Compact Convolutional Neural Networks with Transfer Learning and Feature Selection. *Horticulturae*, 9(2). <https://doi.org/10.3390/horticulturae9020149>
- [4] Balas, V. E., Uluta, suluta, s, H., & Aslanta, s, V. A. (2023). Design of Efficient Methods for the Detection of Tomato Leaf Disease Utilizing Proposed Ensemble CNN Model. *Electronics* 2023, Vol. 12, Page 827, 12(4), 827. <https://doi.org/10.3390/ELECTRONICS12040827>
- [5] Baser, P., Saini, J. R., & Kotecha, K. (2022). TomConv: An Improved CNN Model for Diagnosis of Diseases in Tomato Plant Leaves. *Procedia Computer Science*, 218, 1825–1833. <https://doi.org/10.1016/j.procs.2023.01.160>
- [6] Cardellicchio, A., Solimani, F., Dimauro, G., Petrozza, A., Summerer, S., Cellini, F., & Renò, V. (2023). Detection of tomato plant phenotyping traits using YOLOv5-based single stage detectors. *Computers and Electronics in Agriculture*, 207, 107757. <https://doi.org/10.1016/J.COMPAG.2023.107757>.
- [7] Kao, I. H., Hsu, Y. W., Yang, Y. Z., Chen, Y. L., Lai, Y. H., & Perng, J. W. (2019). Determination of Lycopersicon maturity using convolutional autoencoders. *Scientia Horticulturae*, 256. <https://doi.org/10.1016/J.SCIENTA.2019.05.065>
- [8] Kaur, P., Harnal, S., Gautam, V., Singh, M. P., & Singh, S. P. (2023). A novel transfer deep learning method for detection and classification of plant leaf disease. *Journal of Ambient Intelligence and Humanized Computing*, 14(9), 12407–12424. <https://doi.org/10.1007/S12652-022-04331-9/METRICS>
- [9] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature* 2015 521:7553, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- [10] Marinello, F., Zou, X., Liu, Z., Zhu, X., Zhang, W., Qian, Y., Li, Y., Karunathilake, E. M. B. M., Le, A. T.,
- [11] Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, 10(1), 18–31. <https://doi.org/10.1109/taffc.2017.2740923>
- [12] Ngo, Q. T., & Yoon, S. (2020). Facial expression recognition based on weighted-cluster loss and deep transfer learning using a highly imbalanced dataset. *Sensors (Switzerland)*, 20(9). <https://doi.org/10.3390/s20092639>
- [13] Ofuya, T. I., Okunlola, A. I., & Mbata, G. N. (2023). A Review of Insect Pest Management in Vegetable Crop Production in Nigeria. *Insects* 2023, Vol. 14, Page 111, 14(2), 111. <https://doi.org/10.3390/INSECTS14020111>
- [14] Punithavathi, R., Rani, A. D. C., Sughashini, K. R., Kurangi, C., Nirmala, M., Ahmed, H. F. T., & Balamurugan, S. P. (2023). Computer Vision and Deep Learning-enabled Weed Detection Model for Precision Agriculture. *Computer Systems Science and Engineering*, 44(3), 2759–2774. <https://doi.org/10.32604/csse.2023.027647>
- [15] Thakur, P. S., Sheorey, T., & Ojha, A. (2023). VGG-ICNN: A Lightweight CNN model for crop disease identification. *Multimedia Tools and Applications*, 82(1), 497–520. <https://doi.org/10.1007/S11042-022-13144-Z/METRICS>

- [16] Verma, S., Chug, A., & Singh, A. P. (2020). Application of convolutional neural networks for evaluation of disease severity in tomato plant. *Journal of Discrete Mathematical Sciences and Cryptography*, 23(1), 273–282. <https://doi.org/10.1080/09720529.2020.1721890>
- [17] Walecki, R., Rudovic, O., Pavlovic, V., Schuller, B., & Pantic, M. (2017). Deep structured learning for facial action unit intensity estimation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*, 5709–5718. <https://doi.org/10.1109/CVPR.2017.605>

### **Author contributions statement**

Babatunde Sunday: Writing—original draft, Methodology, Formal analysis, Conceptualization. Sahabi Ali Yusuf: Writing – review & editing, Supervision, Investigation. Yazeed Masha Abdullahi: Writing – review & editing, Supervision. Mohammed Abdullahi: Writing –review & editing, Supervision, Investigation. Alice Sikemi Matemilola: Supervision, Investigation

### **Data Availability**

The dataset used in this study is publicly available and was obtained from the PlantVillage Tomato Leaf Disease dataset. The dataset contains images of healthy and diseased tomato plant leaves, categorized into multiple disease classes. It is maintained on Kaggle and originally sourced from the PlantVillage project.

### **Funding Statement**

The author(s) received no financial support for the research, authorship, and/or publication of this article.

### **Ethics in Publishing Statement**

I testify on behalf of all co-authors that our article submitted followed ethical principles in publishing.

### **Consent to Publish declaration**

All Author hereby consents to publication of this work

### **Consent to Participate declaration**

All authors hereby agreed to participate in publication of this work