Available online at website: https://jurnal.iaii.or.id/index.php/RESTI

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 9 No. 4 (2025) 840 - 849 e-ISSN: 2580-0760

# Open-Set Recognition for Potato Leaf Disease Identification Using OpenMax

Ike Verawati[1*], Mambaul Hisam[2], Yoga Pristyanto[3]
[1, 2]Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta, Sleman, Indonesia
[1]ikevarawati@amikom.ac.id, [2]mambaulhisam@students.amikom.ac.id, [3]yoga.pristyanto@amikom.ac.id

**Abstract**

*Traditional methods for identifying potato leaf diseases rely on manual visual inspection, which is prone to human error and inefficiencies. Although machine learning models have improved automation, conventional closed-set classifiers fail to recognize unknown diseases outside their training scope, limiting real-world applicability. This study addresses this gap by implementing Open-Set Recognition (OSR) using the OpenMax framework to classify known potato leaf diseases while effectively rejecting unknown pathologies. By leveraging the Xception architecture with dual learning schedulers (ReduceLROnPlateau and StepLR), we optimized the OpenMax parameters, including distance metrics (Euclidean, Eucos) and rejection thresholds. After rigorous tuning, the model achieved 86.8% accuracy and 86.4% F1-score under an openness score of 18.3%, with optimal performance using Euclidean distance and a 0.95 threshold. The results demonstrate robust discrimination between known classes (potato late blight, early blight, healthy leaves) and visually similar unknown classes (e.g., tomato diseases, healthy bell peppers). This study enhances AI-driven agricultural diagnostics by bridging the gap between closed-set precision and open-set practicality, offering a scalable solution for real-world disease identification where novel pathogens may emerge.*

**Keywords**: computer vision; Open-Set Recognition; OpenMax; potato leaf diseases; Xception

*How to Cite:* Ike Verawati, Mambaul Hisam, and Yoga Pristyanto, "Open-Set Recognition for Potato Leaf Disease Identification Using OpenMax", *J. RESTI (Rekayasa Sist. Teknol. Inf.)* , vol. 9, no. 4, pp. 840 - 849, Aug. 2025.
*Permalink*/DOI: https://doi.org/10.29207/resti.v9i4.6525

*Received*: April 9, 2025
*Accepted*: August 8, 2025
*Available Online*: August 18, 2025

*This is an open-access article under the CC BY 4.0 License*
*Published by Ikatan Ahli Informatika Indonesia*

## 1. Introduction

In conventional machine learning, the classes that are available at test time are constrained to the classes that were available at training time. In the context of a model that has been trained with a set of labels $K = \{c_1, c_2, c_3, \dots, c_N\}$, the output of the model at test time is constrained to the scope of that particular set of classes. This assumption, known as closed-set classification, posits that the labels and features available at test time are identical to those employed during training [1].

This issue has been widely recognized in real-world applications [1] - [6]. This limitation has spurred research into Open-Set Recognition (OSR), a more realistic approach for real-world applications [7]. This OSR approach is more realistic for application in the real world [1], for example, in the field of Intrusion Detection System (IDS) to detect zero-day attacks [8]. Additionally, the OSR approach finds application in traditional classification methods.

Conversely, manual identification of plant diseases is often time-consuming and difficult to standardize. Consequently, farmers frequently depend on personal experience or visual observation as a basis for diagnosis, a practice that is susceptible to errors, particularly in the context of novel diseases or symptoms that are not readily apparent [9]. The dearth of specialized training and education in phytopathology can further exacerbate the likelihood of misdiagnosis.

The initial research was conducted by colleagues, including Rashid et al. The study proposes a novel multi-level deep learning model, designated as PDDCNN, which was developed for the precise recognition of potato leaf diseases. The research utilized a dataset amassed in the Pakistan region, designated as the Potato Leaf Disease (PLD). The researchers noted that the PlantVillage dataset might not accurately represent diseases in various regions due to the influence of environmental factors. The dataset was segmented into three distinct portions: 80% was

allocated for training, 10% for validation, and 10% for testing. The researchers also performed an additional technique of segmentation using *You Only Look Once (YOLO)*. The PDDCNN model, which was trained and tested on the PLD dataset, achieved a high accuracy of 99.75%, thereby demonstrating its effectiveness and efficiency in detecting potato leaf diseases [10].

Building on the PlantVillage dataset, Mahum et al. integrated three additional datasets, a total of three additional datasets from disparate sources. The amalgamation of these four datasets resulted in the identification of five distinct classes. The data underwent a normalization process, resulting in a dimension of 128 × 128 pixels. This research article also introduces a new deep learning framework for automatic potato leaf disease detection and classification. Utilizing the *Efficient DenseNet* architecture, the research obtained an accuracy of 97.2% [11].

In the same year, Nazir et al. developed a method based on the EfficientNetV2 architecture to address critical issues of potato leaf disease detection. This disease has a significant impact on global potato crop production. The researchers introduced a novel framework, dubbed EfficientPNet, which was designed to enhance the efficiency and accuracy of the detection process. The dataset employed was the PlantVillage dataset, which was subdivided into two segments: 80% for training and the residual portion for testing following augmentation. The modifications made to the EfficientNetV2 architecture yielded an accuracy of 98.12%, as reported in [12].

Tambe et al. further advanced this line of research. The PlantVillage dataset was segmented into three distinct segments: 70% was allocated for training, 15% for validation, and 15% for testing. Subsequently, the dataset was resized to 256 × 256 pixels and augmented with additional data. The study's architecture design yielded an output accuracy of approximately 99.18%. Their work underscores the transformative potential of deep learning in the realm of agricultural technology, with implications for enhancing crop yields, mitigating financial losses, and ensuring food security through effective disease management in potato farming [13].

To conclude the year 2023 in this literature study, it comes from Sunyoto et al., who improved the model's performance by using the Xception architecture combined with *ReduceLROnPlateau* which adjusts the learning rate dynamically during training. The dataset used is PlantVillage, which is resized to 150 × 150 pixels in 3 classes. To address an imbalanced dataset, they perform data augmentation. The research obtained an accuracy result of 97.22% [14].

On the other hand, the term Open-Set Recognition (OSR) was introduced by Scheirer et al. in an article entitled *"Toward Open Set Recognition"* [7]. The content of the article discusses the problems in traditional machine learning where the model assumes that the knowledge of the class is complete. The model does not know that there is a possibility of knowledge about other classes, like a frog in a shell. Scheirer et al. also introduces their framework—1-vs-Set Machine. The framework they proposed was improved upon Support Vector Machines (SVMs) for open-set scenarios.

Following the introduction of OSR by Scheirer et al., Bandele and Boult present their framework—OpenMax. OpenMax is a pioneering framework in the development of OSR, employing a deep learning approach that utilizes Extreme Value Theory (EVT), as exemplified by the Weibull distribution [15] (Yang et al.), which was developed by Bendale and Boult in 2016 in an article entitled *"Towards Open Set Deep Networks"* [16]. The Weibull distribution is derived by calculating the Mean Activation Vector (MAV) of each class in the second-to-last layer. The result or score of the Weibull distribution for each class is then used to calculate the pseudo-activation on new samples [1].

A multitude of methodologies can be employed to address this issue, with each researcher employing a distinct approach. One such approach, as illustrated by [17], involves the integration of OpenMax with a Generative Adversarial Network (GAN) framework [18]. Alternatively, [16] and [19] have employed EVT as the foundation for decision-making processes. Additionally, [20] and [21] employ a reciprocal point learning approach.

Before exploring OSR in depth, it is essential to establish the foundational principles of closed-set classification. In this paradigm, all training samples are assumed to belong to a predefined set of classes [17]. The model's predictive scope is strictly confined to this set *K*, both during the training and testing phases. This constraint defines closed-set classification, a framework inherent to traditional machine learning systems.

To ascertain the nature of a model, Scheirer et al. introduced the concept of *openness* in a model as indicated by Equation 1 [7].

$$O = 1 - \sqrt{\frac{2 \times |C_{\mathrm{TR}}|}{|C_{\mathrm{TA}}| + |C_{\mathrm{TE}}|}} \tag{1}$$

$$O^{*} = 1 - \sqrt{\frac{2 \times |C_{\mathrm{TR}}|}{|C_{\mathrm{TR}}| + |C_{\mathrm{TE}}|}} \tag{2}$$

$C_{TR}$ is the subset of the training data class, $C_{TE}$ the subset of the testing data class and $C_{TA}$ is a subset of the target class to be classified.

The interpretation of this equation is as follows: If the result is zero, the model is classified as a closed set. The magnitude of the result is directly proportional to the degree of openness exhibited by the model. The result of Equation 1 is a fractional number in the interval [0, 1] [7]. Geng et al. contend that the rationality of

Equation 1 is questionable [1]. This is because it demonstrates the relation $C_{TA} \subseteq C_{TR} \subseteq C_{TE}$, which can yield the value $O < 0$. In Equation 2, the results of Geng et al. have been corrected by Equation 1. The utility of *openness* in identifying or providing information about the model is noteworthy.

A particular instance of the OSR approach is used to classify or identify potato leaf diseases. All these studies [10] - [14], still use the traditional approach—closed-set. Therefore, this research involves a novelty that focuses on solving the problem of closed-set classification in potato leaf diseases by using one of the many open-set recognition methods, namely OpenMax [16]. Agreeing with the statement in their study [13], deep learning approaches have the potential to reform technology in agriculture. However, the closed-set environment is not mature enough to act like "something else", which can be identified as a new disease.

## 2. Methods

As illustrated in Figure 1, the research methodology is initiated with the aggregation of data, followed by a preprocessing stage and subsequent modeling. After obtaining the Activation Vector (AV) from the closed-set modeling process—a prerequisite for OpenMax [16]. The AV was employed for fitting the Weibull distribution. The final step involves evaluation, which includes tuning the OpenMax parameters. Not only will OpenMax be evaluated, but closed-set modeling will also be evaluated.

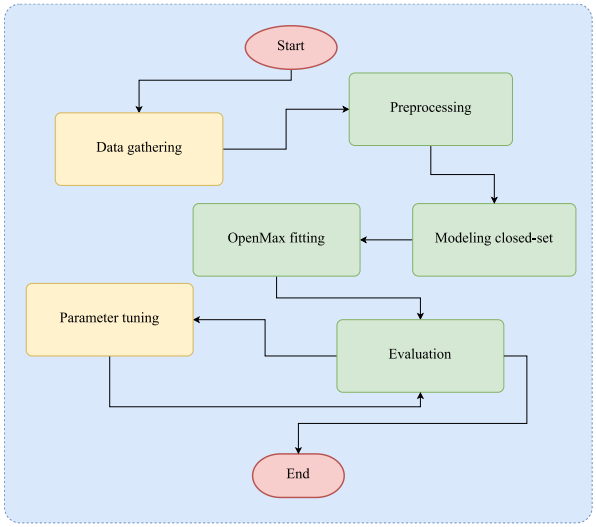



Figure 1. Overview of the overall research flow. Shapes with a yellow background indicate direct processes. Shapes with a green background indicate a collection of processes detailed in each subchapter.

Starting from data or dataset gathering, the dataset is obtained on the Kaggle[1] web platform. The original source of the dataset comes from the research of Mohanty et al. [22]. When it first appeared in 2016, the dataset was often used for research. for research. Especially research on the subject of plant disease identification in the field of agriculture. This dataset contains 15 classes, which contain a variety of leaf diseases from 3 plants. The 3 plants are: (i) potatoes, (ii) tomatoes, (iii) peppers; each of which has various diseases. However, out of these 15 classes, only a few classes will be used. The details of the classes to be used will be explained in the preprocessing stage.

### *2.1 Preprocessing*

After we gather the data–PlantVillage, the next step is preprocessing. However, to perform preprocessing, it is necessary to know the problems in the dataset that will be used to obtain significant results. Figure illustrates the overall preprocessing process. Of all the preprocessing sub-points that will be discussed, Figure 2 is a detail of the overall preprocessing that researchers are going to do.

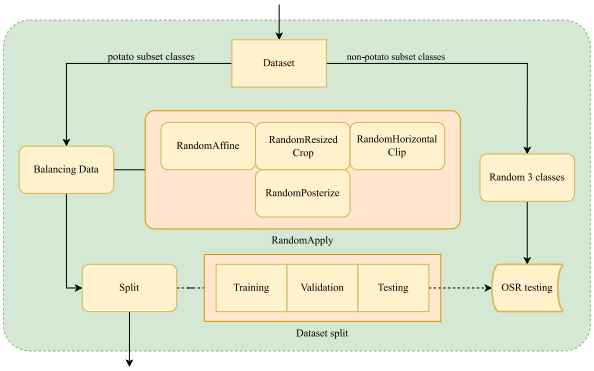



Figure 2. The preprocessing stage that will be performed is described in more detail.

Table 1. Preprocessing methodology.

| Configurations | Parameters |
|---|---|
| Class balancing (minor upscaling) | 100%[2] |
| Dataset splitting train/val/test | 70:20:10 |
| Image augmentations | Table 2 |

The main problem with this dataset is that the class to be used during training is imbalanced. To solve this problem, the researcher used the same approach as Barulina et al. [23]. However, there is a slight change in the amount of data to be segmented to balance 100% of the minority data by applying one of *RandomApply*, as shown in Figure 2.

To align with the requirements to test the OSR, Geng et al. introduces Unknown Unknown Classes (UUCs) derived from related plant domains to simulate real-world scenarios where novel diseases may emerge. Three classes were deliberately selected as UUCs: two from healthy leaf bell peppers and one from tomato diseases (early blight and late blight). These classes were chosen based on their pathological and visual similarity to potato leaf diseases, ensuring they represent plausible "unknown" or "other" threats within the same agricultural domain. As in Table 3. By

[1]https://www.kaggle.com/datasets/emmarex/plantdisease

[2]By randomly augmenting the minor classes in the *RandomApply* block (see Figure ).

integrating these classes, the study rigorously tests the OSR framework's ability to distinguish between known potato diseases and novel, yet visually analogous, pathologies.

Table 2. Image augmentation parameters.

| Augmentation types | | Parameters |
|---|---|---|
| *RandomResizedCrop* | | 250 |
| *RandomHorizontalClip* | | - |
| *RandomPerspective* | *distortion_scale* | 0.3 |
| | *p* | 0.3 |
| *RandomRotation* | | [0, 20] |
| *ColorJitter* | *brightness* | 0.5 |
| | *hue* | 0.3 |
| *Resize* | | 299 |
| *Normalize* | *mean* | [0.485, 0.456, 0.406] |
| | *std* | [0.229, 0.224, 0.225] |

Table 3. Details of the classes that will be used in this study.

| Plant types | Classes | Total |
|---|---|---|
| Potato | Late blight | 1000 |
| | Early blight | 1000 |
| | Healthy | 152 |
| Tomato | Late blight | 1909 |
| | Early blight | 1000 |
| Bell pepper | Healthy | 1478 |

The dataset was partitioned into three subsets following established guidelines [24]: 70% for training, 20% for testing, and 10% for validation. To evaluate OpenMax's robustness, the test set was augmented with UUCs, which were allocated a sample size equivalent to the Known Known Classes (KKCs) in the test set. This ensures a balanced evaluation of the model's ability to discriminate between known and novel classes.

Before training, data augmentation was applied exclusively to the training subset to enhance diversity while preserving the integrity of validation and test data. Techniques such as rotation, flipping, and scaling were employed (detailed in Table 4), with exceptions made only for minor class balancing to address dataset imbalances. This workflow ensures reproducibility and aligns with best practices in deep learning for agricultural applications.

### *2.2 Closed-set modeling*

As an extension of Convolutional Neural Networks (CNNs), OpenMax operates as an activation function that recalibrates class probabilities to account for unknown classes. However, it is critical to emphasize that OpenMax inherently relies on a closed-set classification model as its foundation. This requirement stems from its need to first establish well-defined decision boundaries for known classes before addressing open-set scenarios.

A pivotal component of OpenMax is the AV as dataset, which is derived from the penultimate layer of the CNN. The AV dataset serves as the basis for modeling the Weibull distribution, a statistical method used to estimate the probability of samples belonging to unknown classes. The subsequent section elaborates on the methodology for constructing the AV dataset, a process visually summarized in Figure 3.

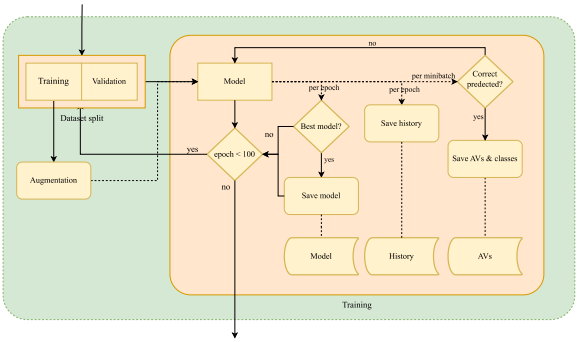



Figure 3. Closed-set modeling flow. A sharp rectangular field represents an object. An oval rectangular field indicates a process. A tile-like field with curves on the right and left sides indicates data stored on the filesystem. A diamond-like plane means a branch.

Following preprocessing, the data was fed into the selected model architecture. The study employed the Xception architecture, a choice informed by its demonstrated efficacy in prior work achieving 97.22% accuracy [14]. Model hyperparameters and configurations are detailed in Table 2. While multiple configurations were explored, this section focuses on the most salient design choices.

Table 4. Model configuration.

| Configurations | | Parameters | |
|---|---|---|---|
| Architecture | | Xception [47] | |
| Epoch | | 100 | |
| Batch Size | | 16 | |
| Loss Function | | *CrossEntropyLoss* | |
| Optimizer | SGD | *lr* | 0.01 |
| | | *momentum* | 0.9 |
| Scheduler | *ReduceLrOnPlateu* | *patience* | 4 |
| | *StepLr* | *step_size* | 5 |
| | | *gamma* | 0.1 |

The *ReduceLROnPlateau* scheduler was implemented with a patience value of 4, selected as the midpoint between the range (2–10) reported in prior studies [14]. For comparative analysis, a conventional *StepLR* scheduler was also tested to evaluate performance under fixed learning rate decay intervals.

A transfer learning approach was adopted, leveraging weights pre-trained on the ImageNet dataset. The classifier head was modified by appending two *Fully Connected (FC)* layers to the flattened output of the penultimate hidden layer. To address gaps in prior implementations [14], the FC layers were initialized using the Xavier uniform distribution for weights and zero-initialized biases, ensuring stable gradient propagation during training.

During inference, the model utilized pre-trained weights and biases to classify input data. For OpenMax modeling, the AVs from the penultimate layer were systematically stored for samples with correct predictions [16]. Additionally, training histories and the best-performing model weights from each epoch were archived to facilitate comprehensive performance analysis.

### *2.3 OpenMax fitting*

Following the extraction of AVs from correctly classified samples, the next phase involves fitting a Weibull distribution to these AVs using EVT. Prior to this fitting, distance values must be computed by subtracting the class-specific mean AV from each AVs within its respective class. This distance metric quantifies deviations from the centroid of each class in the activation space.

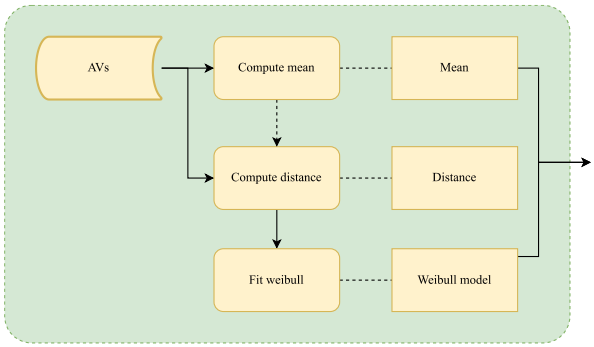



Figure 4. Weibull distribution fitting for OpenMax.

The workflow, as depicted in Figure 4, begins with the AVs stored during the closed-set modeling phase. The process then calculates the mean AV for each class and derives the corresponding distance values. The algorithm proposed by Bendale and Boult [16] subsequently generates two critical outputs: the class-specific mean AV and the parameters of the fitted Weibull distribution. These outputs form the foundation for OpenMax's probabilistic calibration, enabling the rejection of unknown classes during open-set inference.

To optimize OpenMax's performance, this study systematically adjusts its tunable parameters. Specifically, the *tail_size* parameter—a hyperparameter governing the portion of extreme values used in Weibull fitting—is modified based on the methodology outlined in [16]. Additional parameters under investigation are detailed in Table 5, ensuring reproducibility and transparency in the optimization process.

Table 5. OpenMax tuned parameter.

| Distance types | Threshold |
|---|---|
| Euclidean | 0.60 |
| | 0.80 |
| | 0.90 |
| | 0.95 |
| Eucos | 0.60 |
| | 0.80 |
| | 0.90 |
| | 0.95 |

Following Weibull distribution fitting, OpenMax recalibrates the probabilities for each test sample across all known classes. This recalibration redistributes confidence scores to account for the possibility of unknown classes, adjusting likelihoods based on the distance of test samples from class centroids. The final classification is determined by these calibrated probabilities, allowing the model to either assign a sample to a known class or reject it as an unknown class. This mechanism forms the core of OpenMax's open-set recognition capability, bridging the gap between closed-set confidence estimation and real-world diagnostic uncertainty.

### *2.4 Evaluation*

The evaluation stage, illustrated in Figure 5, integrates two key entities derived from prior processes: a partitioned test dataset generated during preprocessing and an augmented test set for OSR evaluation. The latter is constructed by combining the original test data with samples from the UUCs to simulate real-world open-set conditions.

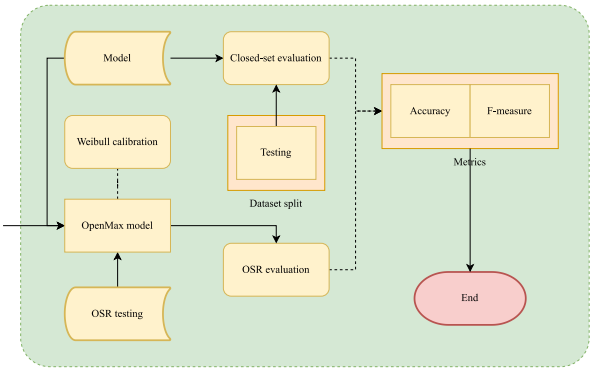



Figure 5. Evaluation flow.

The study employs two commonly used metrics for evaluation: Accuracy (Equation 3) and Weighted F1-score (Equation 4) [25]. Accuracy is defined as the ratio of correctly classified instances (True Positives, *TP*) to the total number of samples (*N*).

$$Accuracy = \frac{\sum_i^{|K|} TP_i}{N} \quad (3)$$

The variable $i$ is employed as an iterator, operating on a set of classes $K$. The objective is to calculate the summation of the total number of instances in the class $i$ that have been correctly classified as belonging to the class $i$, denoted by $TP$. Divided by $N$—the total of samples.

$$f1 = 2 \times \frac{P \times R}{P + R} \quad (4)$$

The $P$ that we got from Equation 5 which calculates weighted-average precision. Also $R$ which calculates the weighted-average recall in Equation 6.

$$P = \frac{\sum_i^{|K|} n_i \times \frac{TP_i}{TP_i + FP_i}}{\sum_i^{|K|} n_i} \quad (5)$$

$$R = \frac{\sum_i^{C} n_i \times \frac{TP_i}{TP_i + FN_i}}{\sum_i^{C} n_i} \quad (6)$$

The weighted-average precision is determined as the ratio of true positives—$TP_i$ to the sum of true positives and false positives—$FP_i$ for each class—$i$. To ensure that each class's contribution reflects its size, the precision for each class is weighted by the number of instances—$n_i$ in that class. The numerator of the weighted precision value total is the sum of the weighted precision values for all classes, while the denominator is normalized by dividing by the total number of instances across all classes.

## 3. Results and Discussions

To be able to experiment, OSR requires data other than KKCs, i.e., data that acts as UUCs. At this stage, the selection is made. As presented in Figure 6, there are 3 classes derived from potato plant leaves that act as KKCs. Among them are: (i) potato late blight; (ii) potato early blight; and (iii) potato healthy. The selection of classes that act as UUCs is also done by selecting the same disease but different plants. Among them are: (i) tomato late blight; (ii) tomato early blight; and (iii) bell pepper healthy. All these samples, both KKCs and UUCs, can be seen in Figure 6, with an openness of about 18.3%.



(a) Sample of potato leaf classes affected by late blight as KKCs. (b) Sample of potato leaf classes affected by early blight as KKCs. (c) Healthy potato leaf grade sample as KKCs. (d) Sample of tomato leaves identified with late blight disease. (e) Sample of tomato leaves identified with early-blight disease. (f) Sample of a healthy bell pepper leaf.

Figure 6. Samples of UUCs and KKCs from three disease-affected plant species in the PlantVillage dataset will be used in this study.

Table 6. Details of KCCs after class balancing process.

| Classes | Total samples |
|---|---|
| Late blight | 304 |
| Early blight | 304 |
| Healthy | 304 |

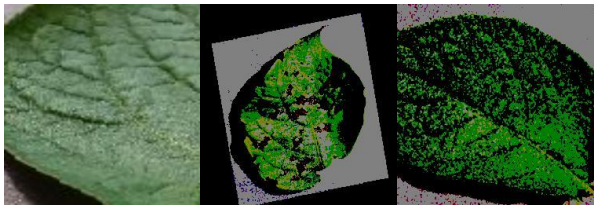


Figure 7. Sample augmentation results for class balancing with *RandomApply*.

At the preprocessing stage, there are two augmentation steps. First of all, it is used to balance the classes, and second, it is used before the sample feeds into the neural network. Table  displays the amount of data that has been up-sampled minor class to balance the data, as much as 100% or twice the minor class with sample results listed in Figure . In this case, the minor class is healthy from 152 potato plants, as shown in Table . The types of augmentation used for up-sampling the minor class are shown in Figure 2.

After balancing the classes, the authors split the samples with details in Table 7 Before training samples are fed into a neural network, the samples will be augmented again, as seen in Table 2, but it is only applied to samples that are not augmented in the balancing stage.

Table 7. Details of the class after splitting the data.

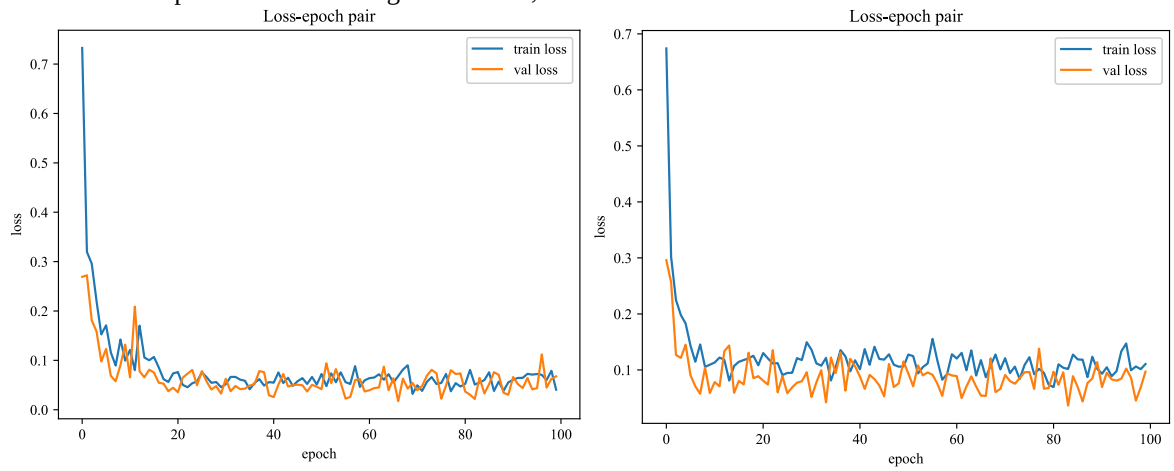
| Plant types | Classes | Train | Val | Test |
|---|---|---|---|---|
| Potato | Late blight | ~213 | ~61 | ~30 |
| | Early blight | ~213 | ~61 | ~30 |
| | Healthy | ~213 | ~61 | ~30 |
| Tomato | Late blight | - | - | ~10 |
| | Early blight | - | - | ~10 |
| Bell pepper | Healthy | - | - | ~10 |

Table 8. Total of AVs-labels pairs.

| Scheduler | AVs-labels pair |
|---|---|
| *ReduceLROnPlateau* | 61087 |
| *StepLr* | 62052 |

The Xception architecture was employed, as depicted in Figure 9 with ImageNet's weight. After modeling the closed-set model, we get an output in the form of a

collection of AVs and labels for each scheduler. The details can be seen in Table 8. Considering that this research uses Xception as the modeling architecture, the resulting AVs has a dimension of 2048 for each AVs obtained in the hidden layer.




(a) Loss plot of *ReduceLROnPlateau* scheduler.

(b) Loss plot of *StepLR* scheduler.

Figure 8. Training and validation loss plots of *ReduceLROnPlateau* and *StepLR*.

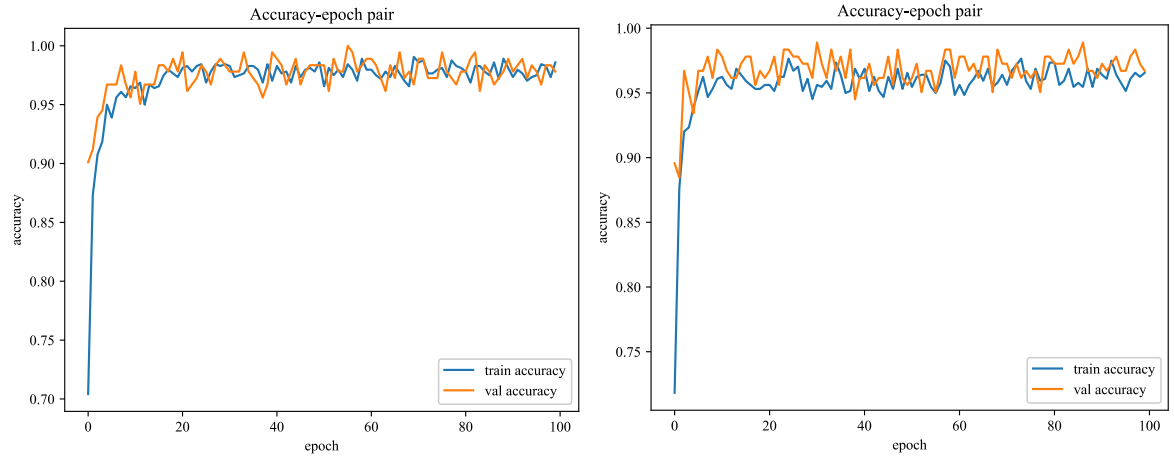
```
================================================================
Total params: 25,009,451
Trainable params: 25,009,451
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 16.37
Forward/backward pass size (MB): 12776.35
Params size (MB): 95.40
Estimated Total Size (MB): 12888.13
----------------------------------------------------------------
```

Figure 9. Summary of the Xception model to be used.

In the plot of Figure 8 , it shows that the model using *ReduceLROnPlateau*—Figure 8a is slightly smoother, and there is no significant change from epoch to epoch, compared to *StepLR*—Figure 8b. It can be concluded that the model using *ReduceLROnPlateau* is more reliable, and no overfitting occurs. Likewise, Figure 10 shows the advantage of *ReduceLROnPlateau*—Figure a over *StepLR*—Figure 10b. Figure 10a is more stable, and there are many common points between the training and validation data. This suggests that the model is not underfitting. Broadly speaking, using loss and accuracy plots, a more reliable model is *ReduceLROnPlateau*. From the computational cost point of view, Table 9, there is an almost 2-fold cost difference. The best result in the model using *ReduceLROnPlateau* is achieved at epoch 55. With the same performance, the best performance of *StepLR* is obtained at epoch 30. This is because StepLr is linear or fixed step size. No matter whether the performance of a model improves or worsens, *StepLR* will decrease the learning rate. From both points of view, the best performance based on the plot is obtained in the model using *ReduceLROnPlateau*. However, when considering the computational cost, *StepLR* is more reliable, with the same best performance but relatively lower computational cost.

Table 9. Closed-set performance metrics for each model against KKCs test data.

| Scheduler | Accuracy | F1 | Epoch |
|---|---|---|---|
| *ReduceLrOnPlateu* | 98.9% | 98.9% | 55 |
| *StepLr* | 98.9% | 98.9% | 30 |




(a) Accuracy plot of *ReduceLROnPlateau* scheduler.

(b) Accuracy plot of *StepLR* scheduler.

Figure 10. Training and validation accuracy plots of *ReduceLROnPlateau* and *StepLR*.



(a) Potato leaf disease test samples in the early blight class.

(a) Potato leaf disease test samples in the late blight class.

(a) Potato leaf testing sample in healthy class.

(a) Testing sample of bell pepper leaves (as UUCs) in healthy class.

Figure 11. Observation Samples.

At this stage, we use both closed-set models to evaluate the OSR performance, see Table 10 and Table 11. Two previously trained closed-set models will also be used at this stage. The purpose is to compare the OpenMax performance of the two models. There is a difference in the OpenMax performance of the two models—Table 10 and Table 11.

Table 10. OSR performance metrics with different parameters against test data on the *ReduceLROnPlateau* model.

| Scheduler | Threshold | Accuracy | F1 |
|---|---|---|---|
| Euclidean | 0.60 | 77.7% | 73.6% |
| | 0.80 | 84.3% | 83.3% |
| | 0.90 | 78.5% | 77.9% |
| | 0.95 | 77.7% | 78.4% |
| Eucos | 0.60 | 75.2% | 70.8% |
| | 0.80 | 76.9% | 75.3% |
| | 0.90 | 78.5% | 77.9% |
| | 0.95 | 85.1% | 85.3% |

Table 11. OSR performance metrics with different parameters against test data on the *StepLR* model.

| Scheduler | Threshold | Accuracy | F1 |
|---|---|---|---|
| Euclidean | 0.60 | 74.4% | 69.4% |
| | 0.80 | 79.3% | 77.8% |
| | 0.90 | 77.7% | 76.6% |
| | 0.95 | 86.8% | 86.4% |
| Eucos | 0.60 | 79.3% | 74.1% |
| | 0.80 | 84.4% | 83.0% |
| | 0.90 | 83.5% | 82.4% |
| | 0.95 | 84.3% | 84.0% |

The OpenMax performance varies significantly across parameters, with the *ReduceLROnPlateau* model excelling in Euclidean distance metrics, while *StepLR* performs better with Eucos distance, ranging up to 9%. Interestingly, there is no difference in the results with threshold 1, regardless of parameters or models. Not only that, the model with the *ReduceLROnPlateau* configuration, gets superior performance on the Euclidean distance type. While the model with the *StepLR* configuration gets superior performance on the Eucos distance type.

The evaluation process extended beyond quantitative metrics, with researchers conducting direct visual inspections of samples, as illustrated in Figure 11. These samples were tested using the highest-performing model configuration, which utilized the *StepLR* scheduler (as detailed in Table 11).

Notably, in the fourth row of Table 12, a sample was classified into the "other" category despite not having the highest score in that class. This outcome stems from the influence of the rejection threshold parameter, set at 0.95 for optimal performance. While the sample exhibited the highest activation score under the healthy class, this value fell below the predefined threshold. Consequently, the model rejected it as a known class and assigned it to the "other" category. This demonstrates the critical role of threshold calibration in balancing closed-set accuracy and open-set robustness.

Unlike SoftMax, OpenMax employs two scenarios for categorizing a sample as the "other" category: (i) the highest OpenMax score for a known class falls below the threshold, or (ii) the "other" class itself attains the highest score. A model can discriminate a sample as "other", if the classifier match either first or second scenario. In the previous example, the fourth row pass the first scenario. This is why Figure 11d classified as "other" category. Another example to match the second scenario, a sample have a higher OpenMax output in "other" category, it's directly classified as "other". This dual-condition mechanism ensures that samples are either confidently assigned to a known class or rejected as novel based on the interplay between activation scores and the predefined threshold.

Table 12. OpenMax scores on each sample for each class.

| Samples | "Other" | Late blight | Early blight | Healthy |
|---|---|---|---|---|
| Figure 11a | 0.000098 | 0.15 | 0.99 | 0.000000071 |
| Figure 11b | 0.00017 | 0.99 | 0.00000059 | 0.000008 |
| Figure 11c | 0.00046 | 0.00012 | 0.00000061 | 0.99 |
| Figure 11d | 0.0610 | 0.1020 | 0.0017 | 0.8354 |

Table 13 presents the logit scores of the SoftMax model in relation to the three known classes. The SoftMax approach, on the other hand, does not accommodate this possibility, as it assumes that all data originates from the known classes. This contrasts with the findings in Table 12, where OpenMax provides probabilities for each class as well as "other" category, and the final decision is also influenced by the 0.95 threshold. This underscores the fundamental limitation of closed-set approaches: the inability to recognize samples from

unseen classes for the potential emergence of novel categories. This methodological constraint confines the model's predictions strictly to predefined classes, rendering it incapable of rejecting inputs that fall outside its operational scope.

Table 13. SoftMax logit score on each sample for each class.

| Samples | Late blight | Early blight | Healthy |
|---|---|---|---|
| Figure 11a | -1.8486 | 9.2106 | -7.2737 |
| Figure 11b | 8.6419 | -5.7062 | -3.100 |
| Figure 11c | -1.3201 | -6.6457 | 7.6754 |
| Figure 11d | 0.5134 | -3.6120 | 2.6168 |

The discrepancies in values between Tablw 12 and Table 13 arise from OpenMax's multi-stage calibration process. Unlike SoftMax classification, OpenMax first employs AVs to compute the Weibull distribution, which estimates the likelihood of samples belonging to unseen classes. This probabilistic adjustment, followed by recalibration via the SoftMax function, enables dynamic decision-making that accounts for both known and potential "other" classes. In contrast, Table 13 reflects output from a conventional closed-set classifier, where scores are generated without such probabilistic adjustments. This distinction underscores how OpenMax's integration of statistical modeling enhances its ability to handle open-set scenarios compared to methods that directly pass inputs to a classifier.

Notably, the retrieval of data from these sources is susceptible to environmental dynamics, including weather conditions, which poses a significant challenge in agricultural contexts. Consequently, the activation vectors drifted, requiring recalibration of the Weibull distribution. To address this challenge, future research endeavors are anticipated to employ adaptive thresholding techniques in the context of weather conditions in order to retrieve the dataset. Alternatively, the utilization of dynamic thresholding methodologies may prove to be a more efficacious approach.

## 4. Conclusions

In this study, we addressed the critical challenge of open-set recognition (OSR) in potato leaf disease identification, where traditional closed-set classifiers fail to account for unknown diseases not encountered during training. By integrating the OpenMax framework with the Xception architecture, we developed a robust model capable of distinguishing between known potato leaf diseases and visually similar unknown pathologies such as tomato late blight and healthy bell pepper leaves. Our approach involved systematic parameter tuning, including distance metrics (Euclidean and Eucos) and threshold values, to optimize performance under open-set conditions.

Quantitatively, the closed-set model demonstrated an accuracy and F1-score of 98.9%, yet it exhibited an inability to recognize novel classes (UUCs). Concurrently, the OpenMax model exhibited a capacity to sustain elevated performance levels within the open-set paradigm, attaining 86.8% accuracy and 86.4% F1-score in the optimal configuration (*StepLR*, Euclidean distance, 0.95 threshold). This indicates a 12.1% decrease in accuracy compared to closed-set, yet it offers the distinct advantage of the ability to reject unknown data.

Furthermore, the discrepancy in performance between distance metrics is indicative of the model's sensitivity to threshold calibration and scheduler selection. In this regard, *ReduceLROnPlateau* demonstrates superior performance in the context of Euclidean distance metrics, while *StepLR* exhibits greater stability in the context of Eucos distance metrics. The model's effectiveness was validated under an *openness* score of 18.3%, simulating real-world scenarios where novel diseases may emerge. Notably, OpenMax outperformed conventional SoftMax-based methods from a robustness point of view, but not by metrics. It was able to dynamically recalibrate probabilities using EVT and Weibull distribution fitting.

After all, this work advances agricultural diagnostics by providing a framework that bridges the gap between laboratory accuracy and field realism. However, limitations include reliance on representative unknown samples for threshold calibration and dataset specificity to the PlantVillage domain. Future research could employ adaptive thresholding or even more sophisticated methods, such as exploring hybrid OSR methods, including the integration of generative adversarial networks (GANs) to synthesize unknown samples or extending the framework to other crops and diseases. By addressing these challenges, the proposed approach paves the way for more adaptable and reliable AI-driven tools in precision agriculture, ultimately supporting sustainable farming practices and global food security.

## Acknowledgements

The authors declare no conflict of interest.

## References

[1] C. Geng, S. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3614–3631, 2020.

[2] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4016–4025.

[3] A. Mahdavi and M. Carvalho, "A survey on open set recognition," in *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, IEEE, 2021, pp. 37–44.

[4] C. Li, E. Zhang, C. Geng, and S. Chen, "All Beings Are Equal in Open Set Recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, pp. 13446–13454.

[5] D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Learning placeholders for open-set recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4401–4410.

[6] M. Barcina-Blanco, J. L. Lobo, P. Garcia-Bringas, and J. Del Ser, "Managing the unknown: a survey on Open Set Recognition and tangential areas," *arXiv preprint arXiv:2312.08785*, 2023.

[7] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward Open Set Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013, doi: 10.1109/TPAMI.2012.256.

[8] M. Soltani, B. Ousat, M. J. Siavoshani, and A. H. Jahangir, "An adaptable deep learning-based intrusion detection system to zero-day attacks," *Journal of Information Security and Applications*, vol. 76, p. 103516, 2023.

[9] A.-K. Mahlein, "Plant Disease Detection by Imaging Sensors – Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping," *Plant Disease*, vol. 100, no. 2, pp. 241–251, 2016, doi: 10.1094/PDIS-03-15-0340-FE.

[10] J. Rashid, I. Khan, G. Ali, S. H. Almotiri, M. A. AlGhamdi, and K. Masood, "Multi-Level Deep Learning Model for Potato Leaf Disease Recognition," *Electronics*, vol. 10, no. 17, 2021, doi: 10.3390/electronics10172064.

[11] R. Mahum *et al.*, "A novel framework for potato leaf disease detection using an efficient deep learning model," *Human and Ecological Risk Assessment: An International Journal*, vol. 29, no. 2, pp. 303–326, 2023.

[12] T. Nazir, M. M. Iqbal, S. Jabbar, A. Hussain, and M. Albathan, "EfficientPNet—an optimized and efficient deep learning approach for classifying disease of potato plant leaves," *Agriculture*, vol. 13, no. 4, p. 841, 2023.

[13] U. Y. Tambe, A. Shobanadevi, A. Shanthini, and H.-C. Hsu, "Potato Leaf Disease Classification using Deep Learning: A Convolutional Neural Network Approach," *arXiv preprint arXiv:2311.02338*, 2023.

[14] A. Sunyoto, Y. Pristyanto, and A. F. Nugraha, "Enhanced Classification of Potato Leaf Disease Using Xception and ReduceLROnPlateau Callbacks," in *2023 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, IEEE, 2023, pp. 52–57.

[15] Z. Lyu, N. B. Gutierrez, and W. J. Beksi, "Metamax: Improved open-set deep neural networks via Weibull calibration," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 439–443.

[16] A. Bendale and T. Boult, "Towards Open Set Deep Networks." 2015. [Online]. Available: https://arxiv.org/abs/1511.06233

[17] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 613–628.

[18] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," *arXiv preprint arXiv:1707.07418*, 2017.

[19] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part III 13*, Springer, 2014, pp. 393–409.

[20] G. Chen *et al.*, "Learning open set network with discriminative reciprocal points," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK , August 23–28, 2020, Proceedings, Part III 16*, Springer, 2020, pp. 507–522.

[21] G. Chen, P. Peng, X. Wang, and Y. Tian, "Adversarial reciprocal points learning for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8065–8081, 2021.

[22] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[23] M. Barulina, S. Okunkov, I. Ulitin, and A. Sanbaev, "Sensitivity of Modern Deep Learning Neural Networks to Unbalanced Datasets in Multiclass Classification Problems," *Applied Sciences*, vol. 13, no. 15, p. 8614, 2023.

[24] C. Cappi *et al.*, "Dataset Definition Standard (DDS)." 2021. [Online]. Available: https://arxiv.org/abs/2101.03020

[25] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, and E. A. Chavez-Urbiola, "Loss functions and metrics in deep learning. A review," *arXiv preprint arXiv:2307.02694*, 2023.