Volume 12. No 2, May 2025
e-ISSN 2527-9572 / p-ISSN 2354-8924

Early Detection Of Nutrient Deficiency In Plants Using Convolutional Neural Algorithm Network (CNN) Algorithm Based On Leaf Image Processing

Kurniawati
Departement Informatics Engineering,
University of Amikom Yogyakarta
Jl. Ring Road Utara, Kec.Depok, Kab. Sleman,
Daerah Istimewa Yogyakarta
kurniawati@students.amikom.ac.id

Ema Utami
Departement Informatics Engineering,
University of Amikom Yogyakarta
Jl. Ring Road Utara, Kec.Depok, Kab. Sleman,
Daerah Istimewa Yogyakarta
ema.u@amikom.ac.id

***Abstract* – Precision agriculture is one of the modern solutions to increase the efficiency and yield of agricultural production. This study proposes an ensemble model based on ResNet , DenseNet , and EfficientNet to detect nutrient deficiencies in lettuce plants, especially nitrogen, phosphorus, and potassium. This model combines the advantages of deep learning architecture with a weighted average ensemble approach to produce more accurate and reliable predictions. Experiments were conducted on a lettuce plant image dataset covering various nutrient deficiency conditions. The test results show that the proposed ensemble model achieves an accuracy of 1.0000 (100%) , indicating excellent performance in identifying nutrient deficiency symptoms. The advantage of this model lies in the unique combination of features obtained from each constituent model, which complement each other in producing the final prediction. This study proves the great potential of deep learning in supporting precision plant nutrient management, with practical applications that have the potential to reduce the time and cost of monitoring in the field. For further development, it is recommended to test this model on a larger and more varied dataset to improve generalization to various field conditions.**

Keywords : Deep Learning, Ensemble Learning, Resnet, Densenet, Efficientnet, Precision Agriculture, Plant Nutrient Deficiencies.

I. INTRODUCTION

Increasing crop productivity and quality is one of the top priorities in the agricultural sector. Nutrient deficiencies in plants are often the main obstacle that can lead to decreased yields and quality. These deficiencies are usually indicated by changes in leaf color, shape, or structure that are often difficult to detect in the early stages through visual observation. Therefore, technology-based automatic detection, especially using deep learning methods, is very important to help farmers make decisions faster and more accurately.

Lettuce is one of the important commodities in the agricultural industry. Plant health, especially the fulfillment of nutritional needs, greatly affects the harvest. Nutrient deficiencies are often difficult to detect early, so a system is needed that is able to recognize early signs of nutrient deficiencies to prevent a decline in crop quality. Deficiencies of essential nutrients such as nitrogen, phosphorus, and potassium in lettuce plants, especially in the early growth phase, can cause significant visual symptoms. These symptoms include leaf discoloration, stunted growth, and decreased crop quality. Other studies utilize deep learning integrated with the Internet of Things (IoT) to detect plant diseases and manage plant nutrition efficiently. This model uses IoT sensors to collect real-time data from the agricultural environment, which is then processed using deep learning algorithms to provide recommendations regarding plant health and nutritional needs. This system not only increases productivity but also supports sustainable agriculture by optimizing resources [1]. Classification of Rice Plant Diseases Based on Leaf Images Using Convolutional Neural Networks conducted by Istiqomah, N., & Murinto, M. [2] . However, detecting symptoms of nutrient deficiencies in plants is often a challenge, especially on large agricultural land and is time-consuming and expensive when using manual methods.

One of the latest approaches in nutrient deficiency detection is the application of convolutional neural networks (CNN). Previous studies have shown that CNNs are very effective in image classification and segmentation. For example, a study by Ferentinos used five CNN architectures, including AlexNet, GoogleNet, and VGG, to detect diseases in 25 types of plants. The VGG architecture managed to achieve an accuracy of up to 99.48%, demonstrating the ability of deep learning models to recognize complex visual patterns associated with plant conditions [3]. Other studies using CNNs have focused on detecting macronutrient deficiencies. For example, an approach using Inception ResNet-v2 successfully detected nutrient deficiencies based on changes in color gradients in okra plant leaves. This study combined transfer learning and fine-tuning to achieve more stable and accurate results [4]. In addition, the application of data augmentation, such as image rotation and shift, has proven effective in improving model performance despite limited initial data.

Several studies have been conducted in the field of deep learning including Zhao, et al. (2019) Analyzing various loss functions for image restoration using neural networks, focusing on improving image quality [5]. Tan & Le (2019) Describes the EfficientNet architecture, which utilizes compound scaling techniques to improve model efficiency without sacrificing accuracy. This model is the basis for various classification and object detection applications [6]. Kim et al. (2024), conducted a study showing that with proper architecture adjustments and training methods, DenseNet can compete with other state-of-the-art models [7]. Ju et al. (2021), Proposed a new pruning method based on the DenseNet architecture for image classification. This method introduces the concept of “threshold” to connect blocks in different ways to reduce memory usage without sacrificing accuracy [8]. Zhang et al. (2020), Introduced dense shortcuts into the ResNet architecture, combining the advantages of ResNet and DenseNet. The study showed that this approach achieves comparable performance to DenseNet with fewer computational resources [9]. Jiang et al. (2021), Using ensemble learning to improve the accuracy of deep learning models in medical image classification [10]. Zhou et al. (2020), Leveraging transfer learning on EfficientNet for early detection of plant diseases using leaf image datasets [11]. Chen et al. (2019), Applying DenseNet to an image prediction system with diverse datasets, shows the flexibility of this model [12]. Jiang, C et al. (2021), Applying DenseNet to non-linear regression tasks, replacing convolutional layers with fully connected layers. The results show that this model outperforms traditional regression models in predicting environmental data [13]. Wang et al. (2019), Using ResNet for plant image classification with a focus on leaf diseases, showed high accuracy on local datasets [14]. Li et al. (2020), Applying DenseNet model in rice disease detection using multispectral image combination [15]. Krešo et al. (2019), Proposed an efficient ladder-style DenseNet architecture for semantic segmentation of high-resolution images, which enables training at megapixel resolution with standard hardware [16]. Abai & Rajmalwar (2019), Built two DenseNet models from scratch for Tiny ImageNet classification, focusing on designing an architecture that fits the image resolution of the dataset [17]. Huang et al. (2020), Introduced Dense Convolutional Network (DenseNet) that connects each layer to all other layers in a feed-forward manner, improving parameter efficiency and addressing the vanishing-gradient problem [18]. Ju et al. (2021), Proposed a novel pruning method based on DenseNet architecture for image classification, which reduces memory usage without sacrificing accuracy [19]. Zhang et al. (2020), Introduced dense shortcuts into ResNet architecture, combining the advantages of ResNet and DenseNet to improve image classification performance [20]. Zhang et al. (2020), Research integrating environmental data with deep learning models to improve detection of nutrient deficiencies in plants [21]. Jiang et al. (2021), Applying DenseNet to non-linear regression tasks, replacing convolutional layers with fully connected layers, and showing superiority over traditional regression models in predicting environmental data [22]. Simarmata et al. (2022) , Implementing the DenseNet architecture for organic and non-organic waste classification, with the aim of increasing accuracy in grouping waste types [23]. Susanto et al. (2021), Conducting a comparative study of Javanese script classification using various deep neural network architectures, including GoogleNet, DenseNet, ResNet, VGG16, and VGG19, to determine the most effective model [24]. Yu et al. (2021), Research developing an ensemble method with a combination of DenseNet and EfficientNet for classification of deficiency symptoms in plants [25]. Li et al. (2022), Analyzing the effectiveness of weighting average-based ensemble learning on various plant image detection tasks [26]. Dosovitskiy et al. (2021), Developing Vision Transformer (ViT), a model that can be an alternative to CNN in plant image classification tasks [27].

In recent decades, Deep Learning-based methods have shown great potential for visual data analysis in agriculture. However, a single model often has limitations in accuracy and generalization. Therefore, this study proposes an optimized CNN-based system to automatically detect nutrient deficiencies in lettuce using a weighting average-based ensemble deep learning method to improve the accuracy of early detection of nutrient deficiencies in lettuce.

Compared with traditional methods that rely on manual chemical or visual analysis, this approach offers better time and cost efficiency. The proposed model also includes various optimization techniques, such as selecting the best CNN architecture and hyperparameter tuning, to ensure a high level of accuracy on the test data.

The study focuses on lettuce crop image processing, using three popular CNN architectures (ResNet, EfficientNet, DenseNet) combined to improve prediction. The success of this study not only makes a significant contribution to the agricultural sector, but also enriches the literature on the application of deep learning in agriculture. With the increasing availability of low-cost hardware and high-quality crop image datasets, this study can have a long-term impact on smart agriculture in the future. This study presents a new approach in early detection of nutrient deficiencies in lettuce plants by applying the Ensemble Deep Learning method based on Weighting Average, which combines ResNet, EfficientNet, and DenseNet. Some of the novel aspects of this study include:

1. Different from previous studies that used individual models such as standard CNN or ResNet separately, this study applies ensemble learning, where the outputs of ResNet, EfficientNet, and DenseNet are combined with a weighting average strategy to improve detection accuracy.
2. This method allows the model to combine the advantages of each architecture, thereby providing more robust and accurate predictions.
3. This study explicitly combines three popular deep learning models and evaluates their performance in detecting nutrient deficiency symptoms in lettuce, which is still rare in the smart agriculture domain.
4. Unlike conventional ensemble learning that uses majority voting or stacking, this study applies weighting average to combine predictions from each model.

II. METHOD

The model used in this study is a combination of three popular deep learning architectures, namely ResNet, EfficientNet, and DenseNet, implemented in a weighting average-based ensemble learning framework. Each model functions as an individual learner in detecting nutrient deficiencies in lettuce plants based on image data. This ensemble architecture consists of three main stages:

1. Feature Extraction: All three architectures (ResNet, EfficientNet, and DenseNet) are used to extract key features from the input data (lettuce leaf images).
2. Feature Combination: The output of each model is passed to a fully connected (FC) linear layer to produce individual prediction outputs.
3. Output Consolidation: Predictions from each learner are combined using the *weighting average method* to produce the final output in the form of a predicted level of nutritional deficiency.

A. ResNet Architecture

ResNet (Residual Neural Network) is a deep learning network architecture that uses residual blocks to solve the *vanishing gradient problem* . In this study, ResNet50 is used, which consists of:

1. Total Layers: 50 layers.
2. Residual Block Structure: Consists of shortcut connections to skip several layers, thus allowing gradient information to be better passed on.
3. Convolutional Layer (Conv2D): Kernel size 3x3 or 1x1
4. Batch Normalization: To normalize the output of each layer.
5. Activation Function: ReLU (Rectified Linear Unit)
6. Global Average Pooling: To reduce the feature dimension before passing to FC.

B. EfficientNet Architecture

EfficientNet is designed with a balanced scaling approach in three dimensions (depth, width, and resolution). This model is lighter and more efficient than other models. In this study, EfficientNet-B3 is used with the following details:

1. Total Layers: 30+ layers (depending on scaling version).
2. Mobile Inverted Bottleneck Conv (MBConv): Uses Depthwise Separable Convolution for efficiency.
3. Swish Activation: A non-linear activation function that improves accuracy.
4. Squeeze-and-Excitation Blocks: Highlight more relevant features in the input.

C. DenseNet Architecture

DenseNet (Densely Connected Convolutional Network) has a unique approach, namely each layer is directly connected to all subsequent layers. This results in better gradient propagation and more optimal feature utilization. In this study, DenseNet121 was used:

1. Total Layers: 121 layers.
2. Dense Block Structure: Consists of several interconnected convolutional layers.
3. Convolutional Layer: Kernel size 3x3 and 1x1.
4. Transition Layer: Reduces dimensionality using 1x1 convolution and pooling.
5. Activation Function: ReLU.

D. Parameter

1. Train/Test/Validation : 80/10/10
2. Batch size : 32
3. Optimizer : Adam
4. Learning rate : 0.001
5. Loss function : CrossEntropyLoss

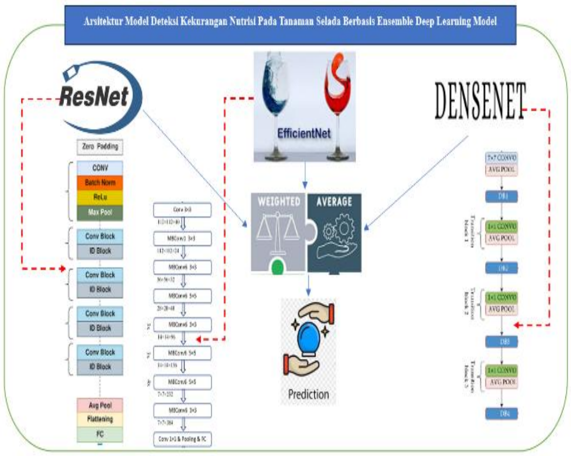

Figure 1. Architectural Model

E. Research Flow

This research begins with the stage of collecting data relevant to the research objectives. The data used includes image datasets that have certain characteristics, such as images of plants showing symptoms of nutrient deficiency. After the data is collected, pre-processing processes are carried out, such as normalization and resizing to ensure the data is ready to be used in the machine learning model. The next stage is model design and development. This research utilizes an ensemble learning model architecture that combines several well-known convolutional networks, namely ResNet, DenseNet, and EfficientNet. Each model is designed to contribute to improving the accuracy and stability of predictions. After the model is designed, training is carried out using previously processed training data. The model is then tested using test data to evaluate performance.

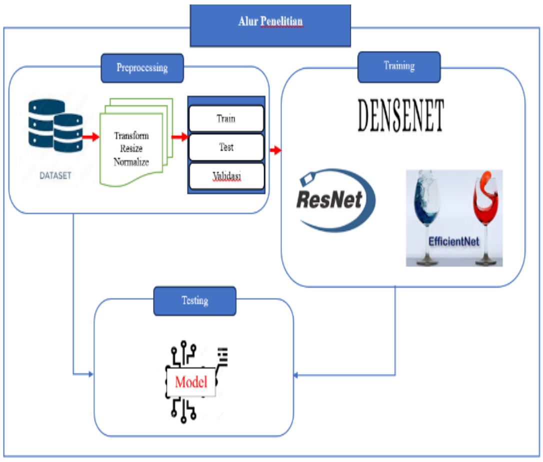

Figure 2. Research Flow

explains the research flow which consists of several stages as follows:

1. Preprocessing
   The dataset is collected and goes through a preprocessing stage. The preprocessing process includes transformation, resizing, and normalization to ensure the data is ready to use. The dataset is then divided into three parts: *train* , *test* , and *validation* for the model training process.
2. Training
   The models used in this study are Deep Learning architectures, namely DenseNet, ResNet, and EfficientNet. The processed dataset is used to train the model with various optimization and parameterization techniques. The model is trained using *train data* and validated using *validation data* to ensure good generalization.
3. Testing
   The trained model is then tested using *test data* to evaluate its performance. Model evaluation is done based on relevant metrics. The test results are used to determine the effectiveness of the model in performing the prediction task.

F. Dataset

The data used in this study is a dataset of lettuce leaf images showing symptoms of nutrient deficiency. This dataset was obtained from two main sources. First, lettuce leaf images were collected from Roboflow with a total of 496 data. Second, the dataset was obtained from Kaggle with a total of 1237 data. This dataset consists of 4 classes of nutrient deficiency symptoms, namely sodium, phosphorus, and potassium (NPK) deficiency factors. The total dataset used is 1,733 images. This dataset contains lettuce leaf images with various conditions, especially related to macronutrient deficiencies, namely sodium (Na), phosphorus (P), and potassium (K), which play an important role in plant growth. The dataset has the following characteristics:

1. Sodium (Na) Deficiency
   Sodium is needed in small amounts to help the plant metabolize. Symptoms of sodium deficiency in lettuce include slow growth, small leaves, and leaves that are paler than normal. Images in the dataset show changes in leaf color, especially yellowing at the edges.
2. Phosphorus (P) Deficiency
   Phosphorus is essential for root growth, flower development, and energy production in cells. Phosphorus deficiency causes lettuce leaves to become darker, sometimes purplish, with more plant growth slow. Some images in the dataset show symptoms of necrosis (tissue death) on the lower leaves as an indication of phosphorus deficiency.
3. Potassium (K) Deficiency
   Potassium helps regulate water balance, photosynthesis, and disease resistance. Lettuce plants that are deficient in potassium show symptoms such as yellowing or drying of leaf tips, and brown spots on leaf edges. The dataset contains images with typical characteristics of curled, dry, or burnt leaves due to potassium deficiency.

Table 1. Dataset

Name	Source	Number of images
Lettuce Multiclass	Roboflow	496
Lettuce	Kaggle	1237

III. RESULTS AND DISCUSSION

In this section, the results of the experiments conducted to build prediction models using the ResNet18, DenseNet121, and EfficientNet_B0 architectures will be explained in detail. The analysis includes evaluating model performance based on loss and validation accuracy during the training process, which is reflected in the training data at each epoch. This assessment is carried out to identify the strengths and weaknesses of each architecture in recognizing patterns in the dataset used. The discussion begins by comparing the training results of the three architectures, including the trend of decreasing loss, stability of validation accuracy, and learning efficiency. Furthermore, an evaluation is carried out on factors that affect performance, such as model complexity, number of parameters, and generalization stability.

A. ResNet18

Loss Trend: Loss decreased significantly from 0.7584 in the first epoch to 0.0075 in the last epoch. This shows that the model learned well during the training process.

Validation Accuracy: Validation accuracy starts from 38.46% in the first epoch to 100% in the sixth epoch and so on. This consistency shows that the model has learned the data pattern perfectly, indicating no overfitting.

B. DenseNet121

Loss Trend: DenseNet121 has a good loss trend, starting from 0.6366 and decreasing to 0.0341 at the 10th epoch. However, the loss value is higher than ResNet18.

Validation Accuracy: The initial validation accuracy was quite high, at 92.31%, but showed fluctuations across several epochs, such as dropping to 78.85% in the ninth epoch, before increasing again to 94.23% in the 10th epoch.

C. EfficientNet

Loss Trend: This model has an initial loss value of 0.8490, decreasing drastically to 0.0155 at the end of training. The efficiency of the loss reduction indicates the strong generalization ability of the model.

Validation Accuracy: Starting from 94.23% in the first epoch, then increasing to 100% in the second epoch, and stable at that number until the 10th epoch.

```
Training ResNet18...
Epoch [1/10], Loss: 0.7584, Val Accuracy: 0.3846
Epoch [2/10], Loss: 0.2355, Val Accuracy: 0.9038
Epoch [3/10], Loss: 0.1422, Val Accuracy: 0.8077
Epoch [4/10], Loss: 0.0815, Val Accuracy: 0.9519
Epoch [5/10], Loss: 0.0415, Val Accuracy: 0.9231
Epoch [6/10], Loss: 0.0302, Val Accuracy: 0.9904
Epoch [7/10], Loss: 0.0133, Val Accuracy: 1.0000
Epoch [8/10], Loss: 0.0132, Val Accuracy: 1.0000
Epoch [9/10], Loss: 0.0117, Val Accuracy: 1.0000
Epoch [10/10], Loss: 0.0075, Val Accuracy: 1.0000
Training DenseNet121...
Epoch [1/10], Loss: 0.6366, Val Accuracy: 0.9231
Epoch [2/10], Loss: 0.1183, Val Accuracy: 0.8846
Epoch [3/10], Loss: 0.0506, Val Accuracy: 0.9808
Epoch [4/10], Loss: 0.0782, Val Accuracy: 0.9423
Epoch [5/10], Loss: 0.1868, Val Accuracy: 0.6442
Epoch [6/10], Loss: 0.0971, Val Accuracy: 0.8558
Epoch [7/10], Loss: 0.1054, Val Accuracy: 0.9327
Epoch [8/10], Loss: 0.1131, Val Accuracy: 0.9327
Epoch [9/10], Loss: 0.0540, Val Accuracy: 0.7885
Epoch [10/10], Loss: 0.0341, Val Accuracy: 0.9423
Training EfficientNet_B0...
Epoch [1/10], Loss: 0.8490, Val Accuracy: 0.9423
Epoch [2/10], Loss: 0.1862, Val Accuracy: 1.0000
Epoch [3/10], Loss: 0.0296, Val Accuracy: 0.9904
Epoch [4/10], Loss: 0.0684, Val Accuracy: 1.0000
Epoch [5/10], Loss: 0.1305, Val Accuracy: 1.0000
Epoch [6/10], Loss: 0.0274, Val Accuracy: 0.9519
Epoch [7/10], Loss: 0.1658, Val Accuracy: 0.9712
Epoch [8/10], Loss: 0.0958, Val Accuracy: 0.9904
Epoch [9/10], Loss: 0.0600, Val Accuracy: 0.9808
Epoch [10/10], Loss: 0.0155, Val Accuracy: 1.0000
```

Figure 3. Training Process

D. Model Performance

Results and discussion are the sections that serve to interpret empirical data obtained from the experimental process. The data is analyzed to explain how the tested models, in this case ResNet18, DenseNet121, and EfficientNet-B0, perform in the classification process. The focus of the analysis is not only on accuracy and loss during training but also on how these models can complete tasks with high efficiency based on their respective architectures. Emphasis is also given to trends emerging from the graphical and tabular data to gain insight into the effectiveness of the models.

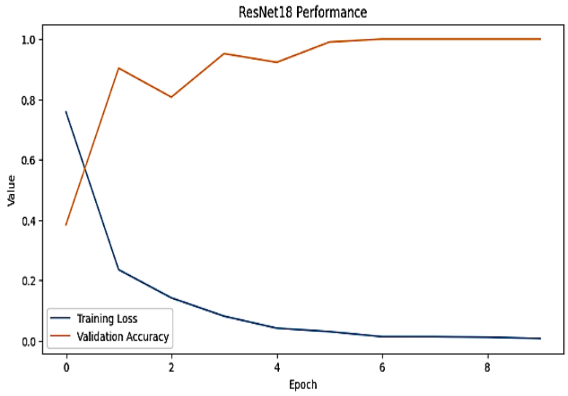

Figure 4. Resnet Performance

Figure 4 shows the performance of ResNet18 during training and validation over 10 epochs. Training Loss (blue line) shows how well the model is able to predict the training data, while Validation Accuracy (orange line) shows the model's accuracy on previously unseen validation data. Decrease in Training Loss: The training loss drops sharply during the first 2-3 epochs, indicating that the model is rapidly learning from the training data. After that, the decline stabilizes to near zero, indicating that the model has learned the pattern very well. Increase in Validation Accuracy: The validation accuracy increases significantly during the first 3-4 epochs, reaching a value close to its maximum. After that, the accuracy stabilizes at around 1.0, indicating that the model has good generalization on the validation data. Trend Fit: The pattern of steadily decreasing loss along with stable accuracy indicates that the model has successfully avoided overfitting, which is often a problem in deep learning models.

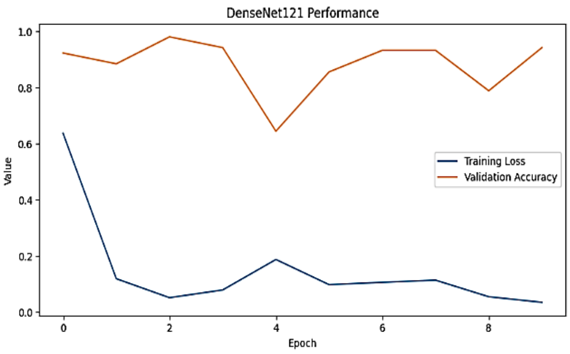

Figure 5. DenseNet Performance

Figure 5 shows the performance graph of the DenseNet121 model during the training process for 10 epochs, illustrating the Training Loss (blue line) and Validation Accuracy (orange line). The loss on the training data consistently decreases from the first to the last epoch, indicating that the model is learning from the data well and correcting its errors. A significant decrease in training loss occurs at the beginning (epochs 1 to 3), after which the loss decreases more slowly, indicating a convergence process. Validation accuracy shows fluctuations between epochs 2 and 8, but overall remains high (above 80%). The increase in validation accuracy is seen at the beginning, although there are some small drops at certain points, which may be due to the variability of the validation data. At the end of training (epoch 10), the validation accuracy increases again to near its maximum value, indicating that the model can produce fairly good predictions on data that was not seen during training.

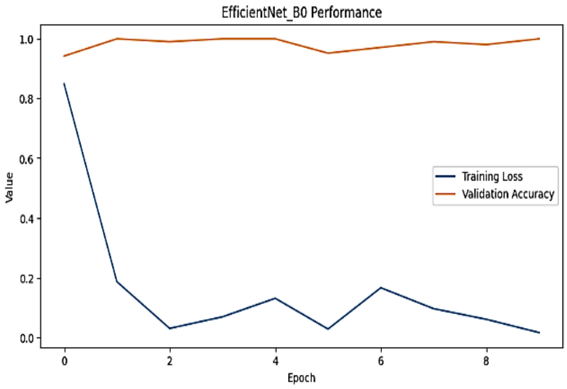

Figure 6. EfficientNet Performance

Figure 6. EfficientNet Performance shows the performance of the EfficientNet B0 model, which illustrates the relationship between Training Loss and Validation Accuracy over 10 training epochs. It can be seen that the loss value decreases significantly since the first epoch and continues to decrease until it approaches zero in the final epoch. This indicates that the model learns well from the training data, and the error generated by the model on the training data becomes very small over time. The validation accuracy value is stable in the range of 0.9 to 1.0, indicating that the model has very good performance for the validation data. This is also an indication that the model does not experience significant overfitting, because the validation performance remains high throughout the training process. The steady decrease in training loss and the stability of validation accuracy indicate efficient optimization and well-calibrated model parameters.

E. Ensemble Model Accuracy

After testing the Ensemble Learning model that combines ResNet, DenseNet, and EfficientNet architectures, the model showed excellent performance with a test accuracy reaching 100% (1.0000). This means that the model successfully predicted all test samples correctly, without any errors. An accuracy of 1.0000 indicates that the model has optimally learned the pattern of the dataset so that all data in the test set can be classified correctly. This accuracy reflects the strength of the ensemble method that combines the advantages of three advanced architectures in deep learning to produce more accurate and stable predictions. These results demonstrate the success of the ensemble model designed in this study. The combination of ResNet, DenseNet, and EfficientNet architectures successfully created a prediction system with optimal performance. With a test accuracy of 100%, this model provides strong evidence of the effectiveness of the ensemble approach to detect nutrient deficiencies in lettuce plants automatically and accurately.

F. Model Analysis

In this study, experiments were conducted using several deep learning architectures, namely ResNet18, DenseNet121, and one additional model for lettuce leaf condition classification. Evaluations were conducted based on loss function metrics and validation accuracy, and analyzed to see the performance of the model in learning data patterns. To provide a broader picture of the advantages and disadvantages of the model used, the experimental results were compared with several previous studies that had a similar focus in image-based plant classification, namely:

1. Istiqomah & Murinto (2024) – using CNN for rice plant disease classification based on leaf images.
2. Wulandhari et al. (2019) – applied Deep CNN to detect nutrient deficiencies in plants.
3. Jiang et al. (2021) – using CNN with data augmentation for stress classification in plants.

The following are the comparison results based on several main aspects, including loss function trends, validation accuracy, indications of overfitting, and the strengths and weaknesses of the models used.

1. Our Research
   a. Dataset : 1733 lettuce leaf images
   b. Models: ResNet18, DenseNet121
   c. Loss Function Trend : - ResNet18: Significantly decreased from 0.7584 → 0.0075 in the last epoch. DenseNet121: Decrease from 0.6366 → 0.0341 with little fluctuation. 3rd model: Decrease from 0.8490 → 0.0155 with high efficiency.
   d. Validation Accuracy : ResNet18: Increased from 38.46% → 100% at 6th epoch. DenseNet121: Fluctuates, from 92.31% → 78.85% → 94.23%.3rd model: Reaches 100% on the 2nd epoch and remains stable.
   e. Overfitting Analysis : ResNet18 & 3rd Model shows no indication of overfitting as loss drops significantly and accuracy is stable. DenseNet121 experienced slight fluctuations in validation accuracy.
   f. Model Advantages: ResNet18 and the 3rd model are very stable and quickly achieve high accuracy. DenseNet121 has fluctuations, but still achieves high accuracy.
   g. Model Weaknesses: DenseNet121 has accuracy fluctuations, indicating a possible lack of stability under some conditions.
2. Istiqomah & Murinto (2024)
   a. Dataset: Rice leaf image dataset with various disease conditions
   b. Model : CNN

c. Loss Function Trend: Not specifically mentioned, but the CNN model shows good convergence in rice disease classification.
d. Validation Accuracy: CNN accuracy reached 97.89% in classifying rice diseases.
e. Overfitting Analysis : It is not explicitly stated whether overfitting occurred or not.
f. Model Advantages: The CNN model is quite accurate in classifying rice diseases.
g. Model Weaknesses: The model may have difficulty generalizing to different lighting conditions.

3. Wulandhari et al. (2019)
a. Dataset: Images of plants with nutrient deficiencies
b. Model: Deep Convolutional Neural Network (CNN)
c. Loss Function Trend: No details of the loss function are mentioned, but CNN is reported to show high accuracy in classifying nutrient deficiencies.
d. Validation Accuracy: CNN accuracy is around 94.5%, but can be improved with data augmentation.
e. Overfitting Analysis : Some models experience overfitting if data augmentation is not used.
f. Model Advantages: CNN is quite powerful for nutrient deficiency classification but requires more data.
g. Model Weaknesses: Models tend to be less than optimal if not given sufficient data augmentation.
h. Jiang et al. (2021)
i. Dataset: Images of plants under various environmental stress conditions such as high temperature, low temperature, excess humidity and drought.
j. Model: CNN with data augmentation
k. Loss Function Trend : Using Cross-Entropy Loss, shows a steady decline with the optimized model.
l. Validation Accuracy: CNN accuracy increases to 98.2% with data augmentation and hyperparameter fine-tuning.
m. Overfitting Analysis: Using regularization techniques such as dropout to address overfitting.
n. Model Advantages: CNN with data augmentation provides the best accuracy in plant stress classification.
o. Model Weaknesses: The model can have difficulty distinguishing between similar stress types without proper preprocessing.

IV. CONCLUSION

Based on the experiments that have been conducted, the ensemble models based on ResNet, DenseNet, and EfficientNet showed very good results in detecting nutrient deficiencies in lettuce plants. The accuracy value of the test results of 1.0000 (100%) indicates that this model is able to predict with a perfect level of accuracy on the test data. The achievement of this accuracy can be explained as follows : Combining the advantages of ResNet (efficiency in handling gradients), DenseNet (rich connections between layers), and EfficientNet (parameter efficiency with high performance) is able to provide strong generalization capabilities to test data. The aggregation strategy using weighted averages helps produce more stable and reliable final predictions, reducing bias from individual models. A well-processed dataset allows the model to accurately understand the pattern of nutrient deficiency symptoms in plants. These results indicate that the ensemble model approach can be an effective solution for applications in precision agriculture, especially for detecting symptoms of nutrient deficiency. This model has the potential to be used practically to improve the efficiency of monitoring and managing plant nutrients. This study shows that the combination of deep learning and ensemble learning can make a significant contribution to the development of artificial intelligence-based technology in precision agriculture.

REFERENCES

[1] Akhtar, N., et al. (2024). "Internet of Things Assisted Plant Disease Detection and Crop Management using Deep Learning for Sustainable Agriculture". IEEE Journals & Magazines. https://ieeexplore.ieee.org/document/10522672

[2] Istiqomah, N., & Murinto, M. (2024). Classification of Rice Plant Diseases Based on Leaf Images Using Convolutional Neural Network (CNN). *JSTIE (Jurnal Sarjana Teknik Informatika) (E-Journal)* , *12* (1),18.

[3] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., & Tang, X. (2019). Residual attention network for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41* (10), 2349–2364. https://doi.org/10.1109/TPAMI.2018.2858826

.[4] Lili Ayu Wulandhari. et al. (2019).” Plant Nutrient Deficiency Detection Using Deep Convolutional Neural Network”. Icic International 2019.

[5] Zhao, H., Gallo, O., Frosio, I., & Kautz, J. (2019). Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, *5* (1), 37–48. https://doi.org/10.1109/TCI.2018.2886931

[6] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *Proceedings of the International Conference on Machine Learning (ICML)* , 6105–6114. https://arxiv.org/abs/1905.11946

[7] Kim, D., Heo, B., & Han, D. (2024). DenseNets Reloaded: Paradigm Shift Beyond ResNets and

ViTs. *arXiv preprint arXiv:2403.19588* . https://arxiv.org/abs/2403.19588

[8] Ju, R.-Y., Lin, T.-Y., & Chiang, J.-S. (2021). New Pruning Method Based on DenseNet Network for Image Classification. *arXiv preprint arXiv:2108.12604* . https://arxiv.org/abs/2108.12604

[9] Zhang, C., Benz, P., Argaw, D.M., Lee, S., Kim, J., Rameau, F., Bazin, J.-C., & Kweon, I.S. (2020). ResNet or DenseNet? Introducing Dense Shortcuts to ResNet. *arXiv preprint:2010.12496* . https://arxiv.org/abs/2010.12496

[10] Jiang, S., Ma, Y., & Wang, H. (2021). Deep learning for plant stress classification using CNNs. *Computers and Electronics in Agriculture* , 190, 106429. https://doi.org/10.1016/j.compag.2021.106429

[11] Zhou, T., Han, G., & Wang, Y. (2020). A CNN-based framework for plant disease recognition. *Computers and Electronics in Agriculture* , 177, 105713. https://doi.org/10.1016/j.compag.2020.105713

[12] Chen, J., Hou, Z., & Zhang, Z. (2019). Plant stress detection using ResNet. *Frontiers in Plant Science* , 10, 1341. https://doi.org/10.3389/fpls.2019.01341

[13] Jiang, C., Jiang, C., Chen, D., & Hu, F. (2021). Densely Connected Neural Networks for Nonlinear Regression. *arXivpreprint:2108.00864* . https://arxiv.org/abs/2108.00864

[14] Wang, H., Li, M., & Sun, Y. (2019). A comparative study on deep learning for plant classification. *Expert Systems with Applications* , 124, 346–354. https://doi.org/10.1016/j.eswa.2019.01.010

[15] Li, X., Yang, Q., & Wang, Z. (2020). Plant leaf using classification transfer learning and CNNs. *Computers and Electronics in Agriculture* , 170, 105232. https://doi.org/10.1016/j.compag.2020.105232

[16] Krešo, I., Krapac, J., & Šegvić, S. (2019). Efficient Ladder-style DenseNets for Semantic Segmentation of Large Images. *arXivpreprint:1905.05661* . https://arxiv.org/abs/1905.05661

[17] Abai, Z., & Rajmalwar, N. (2019). DenseNet Models for Tiny ImageNet Classification. *arXiv preprint arXiv:1904.10429* . https://arxiv.org/abs/1904.10429

[18] Huang, G., Liu, Z., Pleiss, G., Van Der Maaten, L., & Weinberger, K. Q. (2020). Convolutional Networks with Dense Connectivity. *arXiv preprint arXiv:2001.02394* . https://arxiv.org/abs/2001.02394

[19] Ju, R.-Y., Lin, T.-Y., & Chiang, J.-S. (2021). New Pruning Method Based on DenseNet Network for Image Classification. *arXivpreprint:2108.12604* . https://arxiv.org/abs/2108.12604

[20] Zhang, C., Benz, P., Argaw, D.M., Lee, S., Kim, J., Rameau, F., Bazin, J.-C., & Kweon, I.S. (2020). ResNet or DenseNet? Introducing Dense Shortcuts to ResNet. *arXiv preprintarXiv:2010.12496* . https://arxiv.org/abs/2010.12496

[21] Zhang, Y., Wu, X., & Wang, W. (2020). A comprehensive review of deep learning-based plant disease detection techniques. *Computers and Electronics in Agriculture* , 175.105622. https://doi.org/10.1016/j.compag.2020.105622

[22] Jiang, C., Jiang, C., Chen, D., & Hu, F. (2021).Densely Connected Neural Networks for Nonlinear Regression. *arXivpreprint:2108.00864* . https://arxiv.org/abs/2108.00864

[23] Simarmata, AM, Salim, P., & Waruwu, NJ (2022). Densenet Architecture Implementation for Organic and Non-Organic Waste. *Sinkron: Journal and Research of Informatics Engineering*, *7* (4).

[24] Susanto, A., Sari, CA, Rachmawanto, EH, Mulyono, IUW, & Yaacob, NM (2021). A Comparative Study of Javanese Script Classification with GoogleNet, DenseNet, ResNet, VGG16 and VGG19. *Sinkron: Journal and Research of Informatics Engineering*, *7* (2).

[25] Yu, J., Jiang, S., & Zhang, X. (2021). Ensemble deep learning for plant nutrient deficiency classification. *Computers and Electronics in Agriculture* , 186, 106139. https://doi.org/10.1016/j.compag.2021.106139

[26] Li, W., Chen, J., & Zhao, X. (2022). Plant disease identification using CNN-based frameworks. *Plant Methods* , 18(5), 1–12. https://doi.org/10.1186/s13007-022-00766-6

[27] Dosovitskiy, A., Beyer, L., & Kolesnikov, A. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* .