Neural Network Approach Using PyTorch to Predict the Growth of Various Types of Plants

Freddy Artadima Silaban[*1], Ahmad Firdausi[2]
[1,2] Faculty of Engineering, Electrical Engineering Study Program, Mercu Buana University, Jakarta, Indonesia
E-mail: [*1]**freddy.artadima@mercubuana.ac.id**, [2]ahmad.firdausi@mercubuana.ac.id

Abstract

In the era of rapid technological advancement, agriculture faces increasing challenges in optimizing production efficiency and managing resources sustainably. In Indonesia, various plant types are essential agricultural commodities, yet their productivity is often disrupted by erratic weather, poor land management, pest infestations, and land-use change. This study proposes a predictive model for plant growth using a neural network implemented in the PyTorch framework, integrating multiple environmental features such as temperature, humidity, soil moisture, nutrients, pH, and NPK levels. Unlike previous works that typically focus on specific crops or limited variables, this research introduces a multivariate approach combining diverse agro-environmental data to classify plant types accurately. The model architecture was tuned using GridSearchCV, resulting in optimal hyperparameters (e.g., batch size 32, learning rate 0.001, activation: tanh), achieving high performance with Area Under the Curve (AUC) values nearing 1.0 across most classes. Visualization of network weights reveals how input features are transformed through hidden layers, providing interpretability and transparency in decision-making. The proposed system demonstrates strong generalization capability, as validated on unseen data, and offers real-time prediction feasibility for deployment on edge devices such as NVIDIA Jetson Nano. This work contributes a novel, data-driven approach to smart agriculture by enabling precise growth prediction across multiple plant types, enhancing strategic planning for resource allocation and crop management. Future work includes model adaptation for time-series forecasting and validation with live sensor inputs in real-world agricultural environments.

Keywords *— Neural Networks, Machine Learning, PyTorch, ROC, Tuning Hyperparameters*

1. INTRODUCTION

In the current era of globalization and technological advances, the agricultural sector faces increasingly complex challenges, especially in terms of increasing production efficiency and sustainable resource management[1]. In this context, various types of plants are strategic agricultural commodities that have an important role in the economy and food needs[2] in Indonesia. However, the productivity of various types of plants faces problems that are hampered by various factors, ranging from erratic weather conditions, and inappropriate land management, to pest attacks, disease, and land conversion. Therefore, an effective approach is needed to overcome these challenges, one of which is through the application of technology that can provide certainty such as machine learning models[3] in predicting plant growth[4]. Effective agricultural management requires a deep understanding of how various environmental variables affect plant growth[5]. These variables include temperature, humidity,

water content, nutrients, soil pH, and NPK, all of which interact in complex ways to influence crop yields[6]–[8]. Although farmers and experts in agricultural sciences have long understood the importance of these factors, the ability to accurately measure and analyze their combined impacts has generally been problematic.

These limitations are largely due to traditional methods in agriculture that are often unable to integrate data from multiple sources or process it quickly to produce decision-able insights. As a result, decision-making in agricultural management is often based on experience or trial-and-error methods, which do not always produce optimal results.

Artificial intelligence (AI) technology[9]–[13] especially using the Neural Network model[14]–[17] offers great potential in overcoming these problems. With the ability to learn patterns from big data and make accurate predictions, neural networks can help farmers and agricultural scientists identify key factors that influence plant growth and optimize their management. Application of Neural Networks in the agricultural sector[18], [19], especially for predicting the growth of various types of plants, can provide insight and a clearer picture based on data and information.

However, even though the potential of the Neural Network being built is significant, its application in agriculture, especially for a variety of crops, is still in the early stages of development due to many factors. One of the main challenges is the development and adaptation of Neural Network models that can effectively integrate and analyze multi-variable data. Most existing models are designed for more general applications and may not be directly applicable to specific farming conditions without significant modification and farmer needs.

This research applies a Neural Network model with a framework[20]–[22] in the field of smart agriculture. Neural Networks have the power to handle multi-variable data and make accurate predictions. Hence, Neural Network selection, adaptation, and optimization[23]–[25] The right approach is key to developing effective predictive models for the growth of various types of plants.

This research provides significant contributions in several aspects. First, this research introduces a new approach to agricultural management by using Neural Networks to predict the growth of various plant variations. Second, this research integrates various complex and interacting environmental variables, which were previously difficult to analyze comprehensively with traditional methods. Third, the Neural Network model developed in this research is expected to increase prediction accuracy and provide more appropriate recommendations for farmers in managing land and resources.

In this context, this research aims to develop a comprehensive predictive model to ensure the growth of various types of plants by integrating various environmental variables using a neural network approach. With a focus on proven neural network adaptation and optimization, this research is not only to evaluate model performance in a specific agricultural context but also to identify the best strategies for integrating and analyzing multi-variable data. The expected results of this research can provide information and data for farmers and agricultural scientists in making decisions to increase efficiency and productivity in the

cultivation of various types of plants, as well as pave the way for the application of similar technology to other agricultural commodities.

2. RESEARCH METHOD

The three stages consist of data study and pre-processing, as shown in Figure 1. The first stage provides details about the data studied and the pre-processing methods used to clean the data. The second stage discusses the selected neural network model and its tuning and optimization process. The final stage presents the testing and evaluation methods used for The following sections of this paper present complete details of each stage.

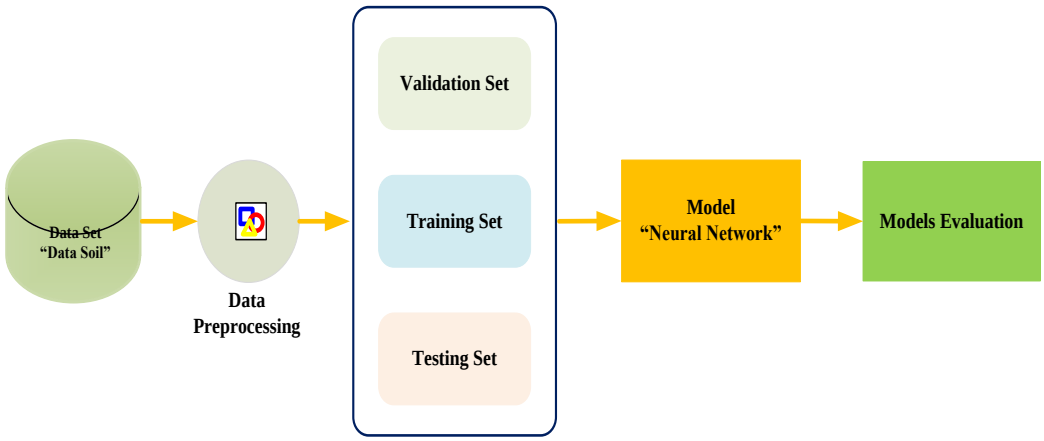


Figure 1. Research Methodology Flow

2.1. Data Study and Pre Process

To ensure that the data used in the neural network model is clean and ready for analysis, the first stage of research is the study and pre-processing of the data. The data collected includes various environmental variables that influence plant growth, such as temperature, humidity, soil water content, nutrients, soil pH, and NPK. To ensure sufficient variation and accuracy, data is carefully collected from multiple sources, such as agricultural sensors, weather stations, and soil laboratories. Data cleaning is the first step in the pre-processing process. Raw data often contains anomalies, missing values, or inconsistencies that can affect model performance. Therefore, data cleaning includes identifying and removing missing or incomplete data, handling anomalies, and applying appropriate imputation methods to fill in missing values. Data is changed after cleaning. Encoding techniques such as one-hot encoding are used in this transformation to convert categorical data into numerical data, which allows the model to process the data correctly. To enable faster and more stable convergence during model training, data normalization was also performed to ensure that all variables were on the same scale. This is done using standard scales for numeric variables.

2.2. Split Data

Once the data is complete, the next step is to divide the dataset into three parts: training, validation, and testing data. A percentage of 70% of the data is used for training, 15% for validation, and 15% for testing. This is important to ensure that the model can be trained

and tested correctly. Training data is used to train the model, validation data is used to test the model's performance during training to avoid overfitting, and testing data is used to test the model's performance during testing.

2.3. Neural Network Models

In this research, a neural network model is specifically designed to manage complex, multivariate data related to plant growth. The model architecture consists of three main components: an input layer, multiple hidden layers, and an output layer. This structure is selected for its proven capability in modeling non-linear relationships and handling high-dimensional data in agricultural and environmental domains.

1) Input Layer: This layer receives all pre-processed and normalized environmental features, including temperature, humidity, soil moisture, nutrients, pH, and NPK content. Normalization ensures numerical stability during training and accelerates convergence.
2) Hidden Layers: The model uses multiple hidden layers, each with varying numbers of neurons. The ReLU (Rectified Linear Unit) activation function is applied at each layer to efficiently learn non-linear dependencies. ReLU is chosen over sigmoid or tanh at this stage because it reduces the risk of vanishing gradients and accelerates training without saturating the output. The number of neurons and layers was determined empirically through hyperparameter tuning (see Section 3.5), balancing model complexity and generalization ability.
3) Output Layer: A Softmax activation function is used in the output layer to convert the final layer’s outputs into a probability distribution across multiple plant classes. This is appropriate for multi-class classification problems, allowing the model to predict the most likely plant type based on the input feature vector.

The overall architecture reflects best practices in classification tasks using feedforward neural networks and is tailored to maximize interpretability, accuracy, and suitability for deployment on resource-constrained edge devices.

2.4. Evaluation Model

Once the model is trained, its performance is evaluated using test data. Some of the evaluation metrics used include:

1) Accuracy: Measures the percentage of correct predictions out of all predictions made by the model.
2) Precision, Recall, and F1-Score: These metrics are used to assess model performance in terms of precision and sensitivity. Precision measures how many positive predictions are correct, while recall measures how many positive predictions out of the total positive cases are true. F1-Score is the harmonic average of precision and recall.
3) Confusion Matrix: This matrix visualizes the model's classification performance by showing the number of correct and incorrect predictions for each class. The confusion matrix helps in understanding the specific errors made by the model.

4) ROC Curve and AUC: The ROC (Receiver Operating Characteristic) curve is used to evaluate the model's performance in predicting classes probabilistically. AUC (Area Under the Curve) measures a model's ability to differentiate between different classes.

This evaluation provides a comprehensive picture of the performance of neural network models in predicting the growth of various types of plants. Evaluation results are used to identify model strengths and weaknesses, as well as to make necessary adjustments to improve model performance. This metric provides a comprehensive picture of the model's performance in classifying data and helps assess the model's ability to correctly find classes under various conditions[26].

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$
$$Precision = TP / (TP + FP) \quad (2)$$
$$Recall = TP / (TP + FP) \quad (3)$$
$$F1 = 2 x (Precision * recall) / (precision + recall) \quad (4)$$

Where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative. All model parameters that have been trained in inference use Nvidia Jetson Nano with NVIDIA Maxwell architecture specifications with 128 NVIDIA CUDA® cores, Quad-core ARM Cortex-A57 MPCore processor, 4GB 64-bit LPDDR4 memory, 1600 MHz, 16 GB eMMC 5.1 storage, and Maximum power: 10 watts, Debian operating system.

2.5. Research Procedure Algorithm

The procedure algorithm is shown in Algorithm 1 below, where the evaluation of the metric uses Confusion Matrix, Classification Report, and ROC Curve.

Algorithm 1: Data Classification and Model Evaluation

1: **Input:** dataset `datasetjd2.csv`
2: **Init:** Load dataset into `data`
3: Display basic information and check for missing values
4: Visualize target variable distribution and feature relationships
5: Encode the target variable `label`
6: Split data into features `X` and target `y`
7: Standardize features and split into training, validation, and testing sets (70%, 15%, 15%)
8: Convert data into PyTorch tensors and create DataLoaders
9: **Define PlantClassifier model:**
- Layer1: Linear(input_size, 64), Activation: ReLU
- Layer2: Linear(64, 32), Activation: ReLU
- Layer3: Linear(32, num_classes)
10: Initialize model, loss function (CrossEntropyLoss), and optimizer (Adam)
11: **Train the model:**
- For `epoch` in `num_epochs`:
  - For each batch in `train_loader`:
    - Forward pass, compute loss, backward pass, and optimize

Algorithm 1: Data Classification and Model Evaluation

12: **Evaluate the model on test data:**
- Predict on `test_loader` and compute `y_pred` and `y_true`

13: Compute confusion matrix and classification report
14: Plot confusion matrix as heatmap
15: **Compute ROC curve and AUC for each class:**
- Binarize output labels, compute `fpr`, `tpr`, and `roc_auc` for each class

16: Plot ROC curve
17: **Visualize weights of each layer:**
- Plot heatmap of weights for each layer

18: **Classify new data:**
- Preprocess, predict, and decode new data labels

19: **Hyperparameter tuning using GridSearchCV:**
- Define `create_model` function
- Initialize `NeuralNetClassifier` with Skorch
- Define parameter grid, perform GridSearchCV
- Display best parameters and results

3. RESEARCH RESULTS AND DISCUSSION

3.1. Statistical Summary of Data

An overview of the distribution and variation of data for each attribute analyzed is explained in the statistical summary of the dataset in Table 1. The attributes nitrogen (N), phosphate (P), and potassium (K) respectively have an average of 50,552, 53,363, and 48,149, with standard deviations of 36,917, 32,986, and 50,648, respectively. There are significant differences between them. With some very high data points, these values indicate varying levels of soil nutrient content. The temperature (temperature) and humidity (humidity) attributes averaged 25,616 and 71,482, respectively indicating the general circumstances in which the data were collected. Large variations in temperature and humidity can also affect plant growth

Table 1. Data Summary Statistics

Attribute	mean	Std	Min	25%	50%	75%	max
N	50,552	36,917	0,000	21,000	37,000	84,250	140,000
P	53,363	32,986	5,000	28,000	51,000	68,000	145,000
K	48,149	50,648	5,000	20,000	32,000	49,000	205,000
temperature	25,616	5,064	8,826	22,769	25,599	28,562	43,675
Humidity	71,482	22,264	14,258	60,262	80,473	89,949	99,982
Ph	6,469	0.774	3,505	5,972	6,425	6,924	9,935
Rainfall	103,464	54,958	20,211	64,552	94,868	124,268	298,560

The pH and rainfall attributes have an average of 6,469 and 103,464 respectively, which indicates the condition of soil acidity and the level of rainfall received by plants. The pH standard deviation of 0.774 shows a relatively small variation compared to other attributes,

while rainfall has a large variation with a standard deviation of 54.958. This information is important for understanding how these environmental factors influence plant growth and can be used to develop more accurate predictive models. The source of the dataset used in this research is https://github.com/Elemento24/Smart-Agriculture

3.2. Descriptive Analysis of Data

The classification results of the neural network model that has been trained to predict plant types based on various environmental variables are shown in the confusion matrix. This matrix shows the number of correct and incorrect predictions for each plant class. The numbers on the main diagonal of the matrix indicate correct predictions; This model succeeded in predicting the number of correct predictions for various types of plants such as apples, mangoes, chickpeas, coconuts, and kidney beans. No significant misclassification errors were found, indicating that the model performed well in determining plant types according to the given environmental variables.

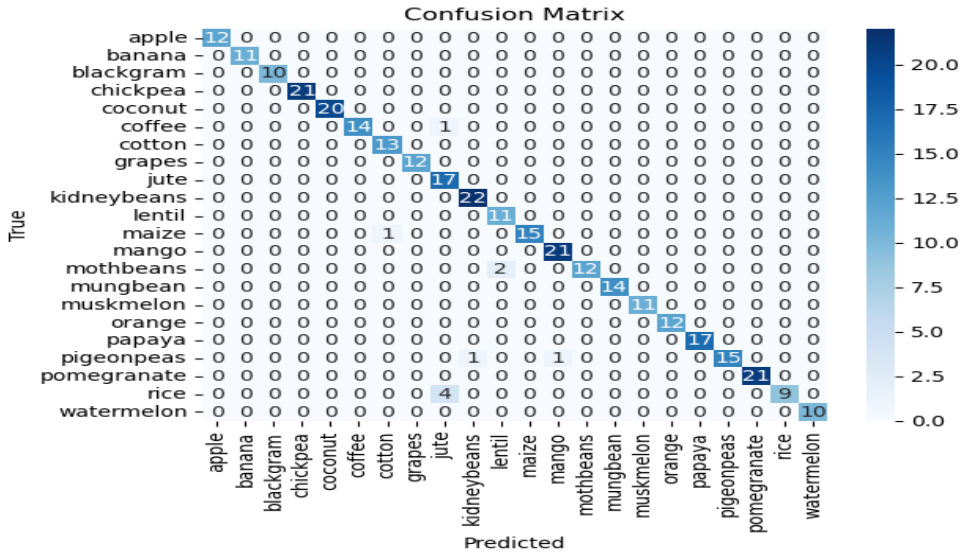


Figure 2. Confusion Matrix

Overall, Figure 2 is a confusion matrix showing that the neural network model developed is very accurate because all classes make correct predictions without significant classification errors. These results show that the model can be used effectively to help farmers use diverse environmental data to make decisions about crop management. To gain a better understanding of model performance, additional evaluation can be performed by calculating metrics such as overall accuracy, accuracy, recall, and F1 score. So, the use of neural networks in agriculture has great potential to increase productivity and efficiency through accurate and reliable predictions.

3.3. Correlation Between Prediction Features

In Figure 3, the correlation matrix between features shows the linear relationship between the various variables included in the dataset. Values between 1 and 1 are indicated as a perfect positive relationship, a perfect negative relationship, and no linear relationship at all.

Certain variables have significant relationships with each other, as shown by this correlation matrix. For example, phosphate (P) and potassium (K) have a strong positive correlation, with a correlation value of 0.74. This shows that land that contains a lot of phosphorus also has a lot of potassium.

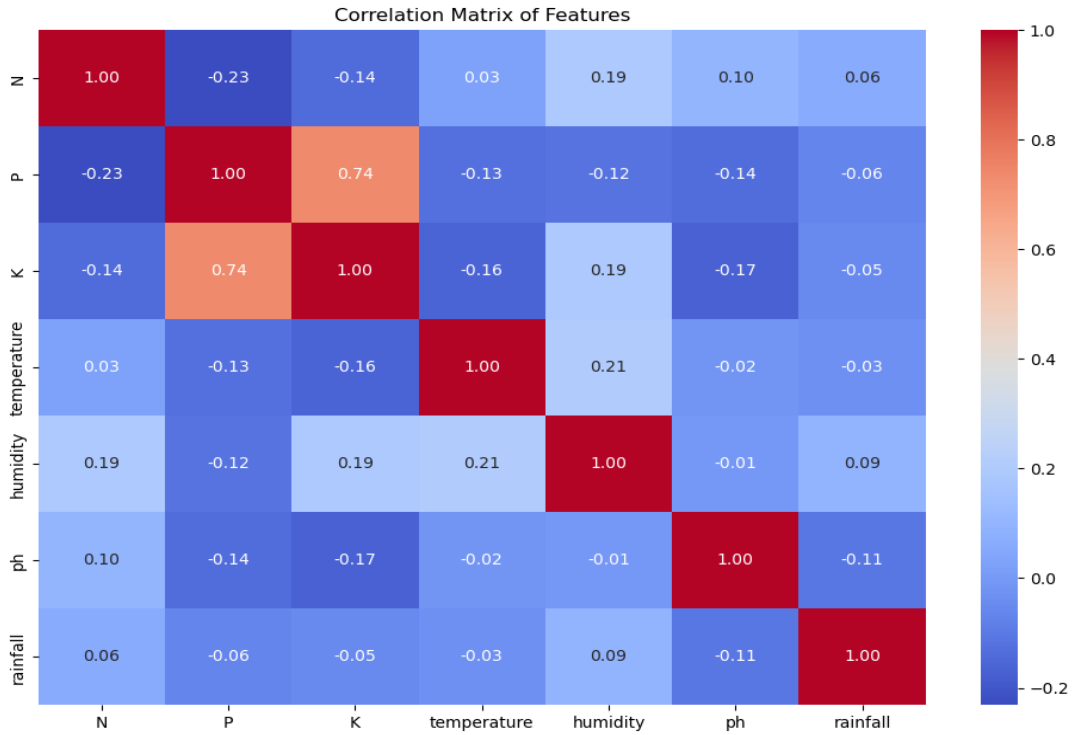
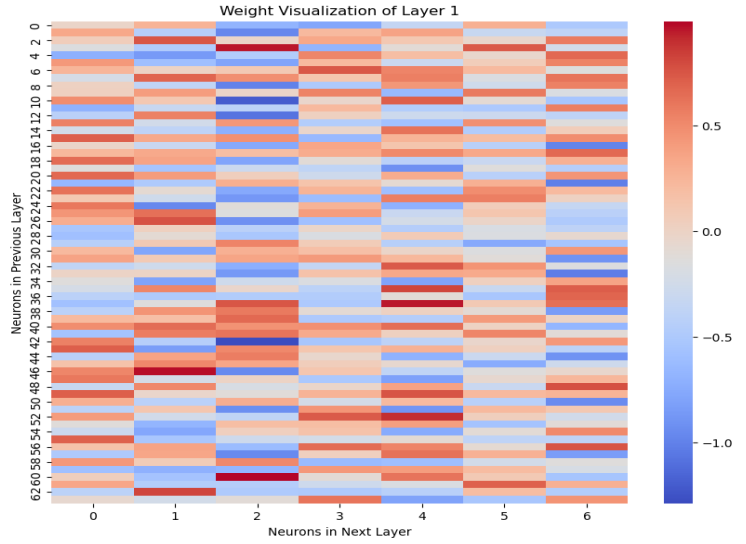


Figure 3. Correlation Matrix Between Attributes

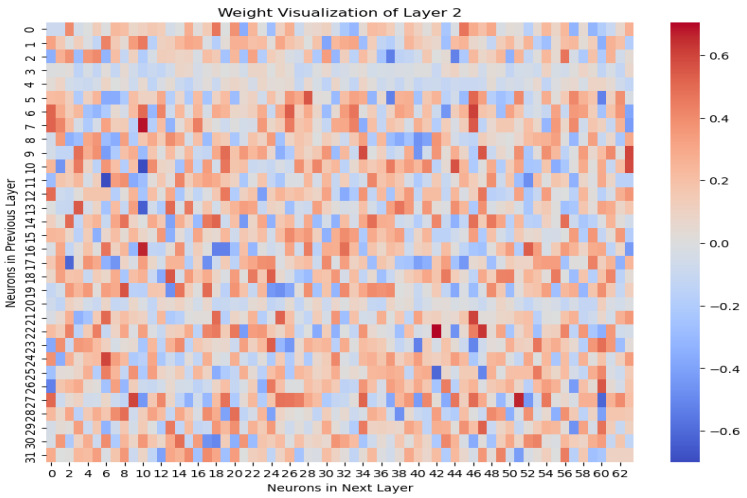
On the other hand, some variables show low or even negative correlation. For example, the negative correlation of nitrogen (N) with phosphorus (P) is -0.23, and potassium (K) is -0.14. This shows that increasing phosphorus and potassium levels is not always accompanied by increasing nitrogen levels. In addition, environmental variables such as temperature (temperature) and humidity (humidity) have different correlations with soil variables. However, there were lower correlation values, indicating that these variables may play separate roles in influencing plant growth. To understand the complex interactions between variables, correlation analysis is very important. It can also help in creating predictive models for better plant growth.

3.4. Visualization of Weights at the Neural Network Layer

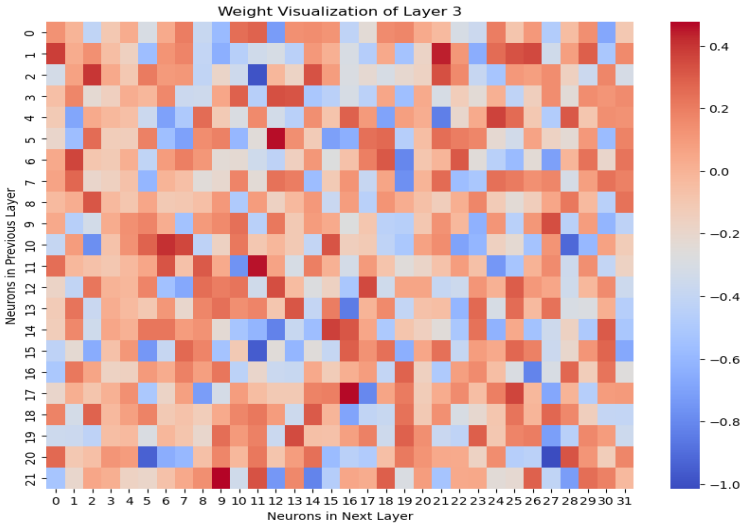
In the neural network model used for plant classification, the visualization of the three-layer weights is shown in the third figure 4 below. In the heatmap, red indicates a higher weight value, while blue indicates a lower weight value. These weights show how each neuron in the first layer contributes to the neurons in the next layer; The distribution of weights in the first layer varies greatly with seemingly random patterns, indicating that this layer is responsible for capturing the basic features of the input data.



(a). Layer 1



(b). Layer 2



(c). layer 3

Figure 4. Visualization of Neural Network Weights in (a). Layer 1; (b). Layer 2; (c). Layer 3

More complex patterns are displayed in the weight visualization in the second layer. This layer is responsible for processing the features that have been learned by the first layer, and non-linear transformations are carried out by this layer. Color variations indicate that this layer performs further non-linear transformations on the input data. The third layer, which is the last layer before the output, shows a more focused weight distribution. This pattern shows that the third layer Overall, this visualization shows how the weights in each layer of the neural network work together to make predictions and process data.

3.5. Results of Hyperparameter Tuning of Neural Network Models

Table 2 shows that the combination of various hyperparameters greatly influences the performance of the neural network model. Hyperparameter settings are carried out to find the most effective parameter combination for the neural network model. In this process, several hyperparameters are tested. These include the batch size (batch_size), learning rate (learning_rate or lr), the maximum number of epochs (max_epochs), activation function (module__activation), dropout rate (module__dropout_rate), and the number of neurons in the hidden layer.

Table 2. Results of Hyperparameter Tuning of the Neural Network Model

Hyperparameters	Mean Test Score	Std Test Score
batch_size: 32, lr: 0.001, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.2, module__num_neurons: 128	0.987016	0.003298
batch_size: 64, lr: 0.001, max_epochs: 100, module__activation: relu, module__dropout_rate: 0.0, module__num_neurons: 128	0.986366	0.002741
batch_size: 64, lr: 0.01, max_epochs: 100, module__activation: relu, module__dropout_rate: 0.5, module__num_neurons: 128	0.986365	0.001579
batch_size: 32, lr: 0.001, max_epochs: 100, module__activation: relu, module__dropout_rate: 0.0, module__num_neurons: 128	0.985719	0.005097
batch_size: 64, lr: 0.001, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 128	0.985719	0.005097
batch_size: 64, lr: 0.01, max_epochs: 50, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 128	0.985718	0.003658
batch_size: 64, lr: 0.01, max_epochs: 50, module__activation: relu, module__dropout_rate: 0.0, module__num_neurons: 64	0.985716	0.003302
batch_size: 32, lr: 0.001, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 128	0.985072	0.007499
batch_size: 64, lr: 0.01, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 64	0.98507	0.006413
batch_size: 32, lr: 0.001, max_epochs: 100, module__activation: relu, module__dropout_rate: 0.2, module__num_neurons: 128	0.984421	0.006348
batch_size: 64, lr: 0.01, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 32	0.984418	0.003168

Hyperparameters	Mean Test Score	Std Test Score
batch_size: 32, lr: 0.001, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 64	0.983772	0.006409
batch_size: 64, lr: 0.01, max_epochs: 50, module__activation: relu, module__dropout_rate: 0.0, module__num_neurons: 128	0.98377	0.004848
batch_size: 64, lr: 0.01, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 128	0.98377	0.003656
batch_size: 64, lr: 0.001, max_epochs: 50, module__activation: relu, module__dropout_rate: 0.0, module__num_neurons: 128	0.983768	0.004586
batch_size: 32, lr: 0.01, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.2, module__num_neurons: 32	0.983766	0.005591
batch_size: 32, lr: 0.01, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 64	0.983124	0.007326
batch_size: 32, lr: 0.01, max_epochs: 100, module__activation: relu, module__dropout_rate: 0.0, module__num_neurons: 32	0.983122	0.006003
batch_size: 64, lr: 0.01, max_epochs: 50, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 32	0.983122	0.006003
batch_size: 32, lr: 0.01, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.0, module__num_neurons: 128	0.983121	0.004842

From table 2 above, it shows that the best hyperparameter combination is batch_size: 32, lr: 0.001, max_epochs: 100, module__activation: tanh, module__dropout_rate: 0.2, and module__num_neurons: 128. The average test score is 0.987, with a standard deviation (std) of 0.003. This combination shows that the model can achieve excellent performance with tanh activation functions, small batch sizes, and low learning rates. The tanh function's ability to handle vanishing gradient problems in deeper networks as well as the use of dropouts that help prevent overfitting may be the cause.

3.6. ROC (Receiver Operating Characteristic) Curve Value

The performance of the classification model for various types of plants is shown in the ROC curve image. The ROC curve shows the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR) at various classification thresholds. How well the model can differentiate between various classes is indicated by the area under the curve (AUC). This graph shows that almost all classes have very high AUC values, around 1.0, which shows that the model has a high level of accuracy in classifying plant types.

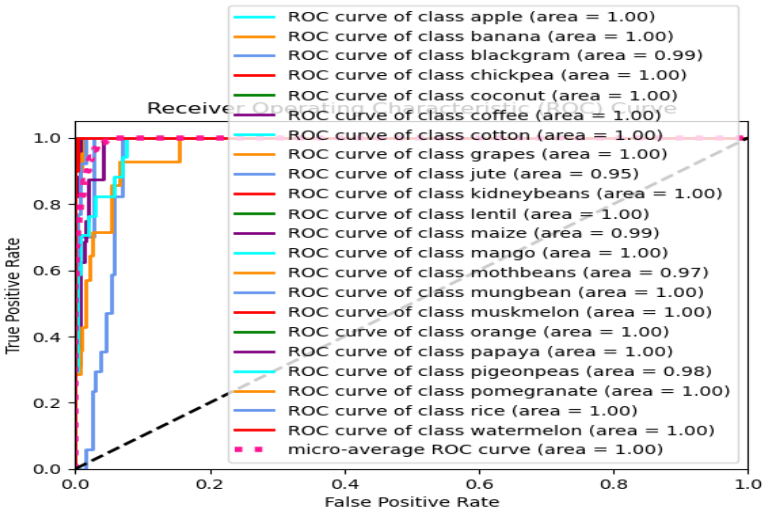


Figure 5. ROC curve

In particular, some classes such as apple, banana, chickpea, coconut, coffee, grapes, kidney beans, lentil, mango, muskmelon, orange, pomegranate, rice, and watermelon have an AUC of 1.00, indicating that the model can differentiate between these classes without error. Only a few classes, such as blackgrams and mothbeans, have AUCs slightly below 1.00, but still show excellent performance. A high AUC value indicates that the model used is very effective in determining and classifying plant types based on the features that have been trained. These results show that the applied neural network model can capture important patterns in the data and apply them well to the test data.

3.7. Testing Predictions on New Data

The new data classification process using a pre-trained Neural Network model is shown in Figure 6. Several environmental features, such as levels of nitrogen (N), phosphorus (P), potassium (K), temperature (temperature), humidity (humidity), pH land, and precipitation (rain), are part of the new data that will be classified. Once entered into a pandas DataFrame, these values are passed to the classify_new_data function to find the plant type that best suits the conditions. The classification results show that the new data is thought to be the "mango" plant.

```
new_data = pd.DataFrame({
    'N': [20],
    'P': [24],
    'K': [20],
    'temperature': [24],
    'humidity': [23],
    'ph': [5],
    'rainfall': [223],
})

# Panggil fungsi untuk mengklasifikasikan data baru
predictions = classify_new_data(new_data)
print("Predicted Labels for New Data:")
print(predictions)
```

```
Predicted Labels for New Data:
['mango']
```

Figure 6. New Data Prediction Simulation

In this classification process, a Neural Network model that has been previously trained with similar data is used. This model uses various environmental features to determine the type of plant-based on learned patterns and successfully predicts that the conditions are most suitable for mango plants. This shows the model's ability to analyze feature combinations and make accurate predictions based on the training data it has received. Next, the model that has been tested is continued to process storage in the form of .pth format, used for the inference process on edge computing devices.

4. CONCLUSION

This study demonstrates that neural network models, when integrated with diverse environmental variables such as temperature, humidity, soil moisture, nutrients, pH, and NPK levels, are highly effective in predicting plant growth. The model developed using the PyTorch framework achieved high classification accuracy, with AUC values approaching 1.0 for most plant types, confirming its robustness and reliability. Furthermore, visualization of model weights across layers offers interpretability, enhancing transparency in the decision-making process.nBy combining multivariate environmental data and deep learning, this research provides a novel, data-driven approach to agricultural management, enabling more efficient resource allocation and crop planning. The ability to deploy the trained model on edge computing devices such as the NVIDIA Jetson Nano further supports real-time implementation in field conditions. However, this study is limited by the use of static datasets, and the model has not yet been validated with real-time sensor data or tested under varying geographic and climate conditions. Future work will focus on integrating time-series data for dynamic prediction, testing model generalization in diverse agricultural settings, and deploying the system in practical smart farming environments to support sustainable and scalable agricultural practices.

5. ACKNOWLEDGEMENTS

This work was partially supported by Mercu Buana University

6. REFERENCES

[1] S. Anwar, A. Hartono, and AD Susila, "Management and Fertilization of Phosphorus and Potassium in Intensive Shallot Farming in Four Villages in Brebes," vol. 9, no. April, p. 27–37, 2018.

[2] P. Hidayah, M. Izzati, and S. Parman, "Growth and Production of Potato Plants (Solanum tuberosum L. var. Granola) in Different Cultivation Systems,"*Bul. Anat. and Physiol.*, vol. 2, no. 2, p. 218, 2017, doi: 10.14710/baf.2.2.2017.218-225.

[3] M. Altalak, MA Uddin, A. Alajmi, and A. Rizg, "Smart Agriculture Applications Using Deep Learning Technologies: A Survey,"*Appl. Sci.*, vol. 12, no. 12, 2022, doi: 10.3390/app12125919.

[4] AT Balafoutis, FK van Evert, and S. Fountas, “Smart farming technology trends: Economic and environmental effects, labor impact, and adoption readiness,”*Agronomy*, vol. 10, no. 5, 2020, doi: 10.3390/agronomy10050743.

[5] AJ Bongole, JP Hella, and KMK Bengesi, “Combining Climate Smart Agriculture Practices Pays Off: Evidence on Food Security From Southern Highland Zone of Tanzania,”*Front. Sustain. Food Syst.*, vol. 6, 2022, doi: 10.3389/fsufs.2022.541798.

[6] D. Sarkar*et al.*, “Low input sustainable agriculture: A viable climate-smart option for boosting food production in a warming world,” Ecol. Indic., vol. 115, 2020, doi: 10.1016/j.ecolind.2020.106412.

[7] H. Rajaguru and RN Susheel, “Solar based automated plant watering bot for Indian agriculture scenario,”*Int. J. Mech. Eng. Technol.*, vol. 9, no. 288–294, p. 288–294, 2018, [Online]. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85059197633&partnerID=40&md5=a9fdc6076d48d3ffc7811021ad20b1fa

[8] A. Sennhenn, DMG Njarui, BL Maass, and AM Whitbread, “Understanding growth and development of three short-season grain legumes for improved adaptation in semi-arid Eastern Kenya,”*Crop Pasture Sci.*, vol. 68, no. 5, p. 442–456, 2017, doi: 10.1071/CP16416.

[9] P. Nevavuori, N. Narra, and T. Lipping, “Crop yield prediction with deep convolutional neural networks,”*Comput. Electrons. Agric.*, vol. 163, 2019, doi: 10.1016/j.compag.2019.104859.

[10] E. Arulmozhi, BE Moon, JK Basak, T. Sihalath, J. Park, and HT Kim, “Machine learning-based microclimate model for indoor air temperature and relative humidity prediction in a swine building,”*Animals*, vol. 11, no. 1, p. 1–24, 2021, doi: 10.3390/ani11010222.

[11] VDA Kumar, A. Kumar, RS Batth, M. Rashid, SK Gupta, and M. Raghuraman, "Efficient data transfer in edge envisioned environment using artificial intelligence based edge node algorithm,"*Trans. Emergency. Telecommun. Technol.*, vol. 32, no. 6, 2021, doi: 10.1002/ett.4110.

[12] D. Ather*et al.*, “Selection of Smart Manure Composition for Smart Farming Using Artificial Intelligence Technique,” J. Food Qual., vol. 2022, 2022, doi: 10.1155/2022/4351825.

[13] J. Zhang, F. Zhang, X. Huang, and X. Liu, “Leakage-Resilient Authenticated Key Exchange for Edge Artificial Intelligence,”*IEEE Trans. Dependable Security. Comput.*, vol. 18, no. 6, p. 2835–2847, 2021, doi: 10.1109/TDSC.2020.2967703.

[14] N. Farhi, E. Kohen, H. Mamane, and Y. Shavitt, "Prediction of wastewater treatment quality using LSTM neural network,"*Environ. Technol. Innov.*, vol. 23, p. 101632, 2021, doi: 10.1016/j.eti.2021.101632.

[15] E. Kristen, C.-T. Yang, C.-Y. Huang, P.-C. Ko, and H. Fathoni, "On construction of sensors, edge, and cloud (isec) framework for smart system integration and applications,"*IEEE Internet Things J.*, vol. 8, no. 1, p. 309–319, 2021, doi: 10.1109/JIOT.2020.3004244.

[16] PK Reddy Maddikunta*et al.*, “Unmanned Aerial Vehicles in Smart Agriculture: Applications, Requirements, and Challenges,” IEEE Sens. J., vol. 21, no. 16, p. 17608–17619, 2021, doi: 10.1109/JSEN.2021.3049471.

[17] N. Ohana-Levi, I. Zachs, N. Hagag, L. Shemesh, and Y. Netzer, “Grapevine stem water potential estimation based on sensor fusion,”*Comput. Electrons. Agric.*, vol. 198, 2022, doi: 10.1016/j.compag.2022.107016.

[18] T. Balaji, R. M. Bhavadharini, R. Poorni, and N. Krishnaraj, “A prediction of crops using machine learning techniques,”*J. Green Eng.*, vol. 10, no. 11, p. 12392–12403, 2020, [Online]. Available at: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85098446318&partnerID=40&md5=b68bd4df454531eb5fc540041dcf353a

[19] M. Catelani, L. Ciani, A. Bartolini, C. Del Rio, G. Guidi, and G. Patrizi, "Reliability analysis of wireless sensor networks for smart farming applications,"*Sensors*, vol. 21, no. 22, 2021, doi: 10.3390/s21227683.

[20] M.A. Farooq, P. Corcoran, C. Rotariu, and W. Shariff, “Object Detection in Thermal Spectrum for Advanced Driver-Assistance Systems (ADAS),”*IEEE Access*, vol. 9, p. 156465–156481, 2021, doi: 10.1109/ACCESS.2021.3129150.

[21] N. Hussain*et al.*, “Application of deep learning to detect Lamb's quarters (Chenopodium album L.) in potato fields of Atlantic Canada,” Comput. Electrons. Agric., vol. 182, no. September 2020, 2021, doi: 10.1016/j.compag.2021.106040.

[22] J. He, J. Qiu, A. Zeng, Z. Yang, J. Zhai, and J. Tang, “FastMoE: A Fast Mixture-of-Expert Training System,” p. 1–11, 2021, [Online]. Available at: http://arxiv.org/abs/2103.13262

[23] R. Rayhana, G. Xiao, and Z. Liu, “Internet of Things Empowered Smart Greenhouse Farming,”*IEEE J. Radio Freq. Identical.*, vol. 4, no. 3, p. 195–211, 2020, doi: 10.1109/JRFID.2020.2984391.

[24] S. Rajeswari and K. Suthendran, “C5.0: Advanced Decision Tree (ADT) classification model for agricultural data analysis on cloud,”*Comput. Electrons. Agric.*, vol. 156, p. 530–539, 2019, doi: 10.1016/j.compag.2018.12.013.

[25] H. Pang, Z. Zheng, T. Zhen, and A. Sharma, “Smart farming: An approach for disease detection implementing iot and image processing,”*Int. J. Agric. Environ. Inf. Syst.*, vol. 12, no. 1, p. 55–67, 2021, doi: 10.4018/IJAEIS.20210101.oa4.

[26] D. Wu, H. Xu, Z. Jiang, W. Yu, X. Wei, and J. Lu, “EdgeLSTM: Towards Deep and Sequential Edge Computing for IoT Applications,”*IEEE/ACM Trans. Netw.*, vol. 29, no. 4, p. 1895–1908, 2021, doi: 10.1109/TNET.2021.3075468