# Rice Leaf Disease Classification with ResNet50 optimized by Optuna

**FINAL PROJECT DEEP LEARNING (COMP6826001) - LD01**

Senin, 8 Desember 2025

Member:

Okky Sudibyo Rades - 2702300575

Marcello Evan Wijaya - 2702266026

Azka Farrel Keandra Aristyan - 2702298652

**SCHOOL OF COMPUTER SCIENCE**

**BINA NUSANTARA UNIVERSITY @Kemanggisan**

**2025**

## 1. Abstract

Rice diseases significantly affected the production of rice in Indonesia, where rice is their main food source. This project proposes a deep learning approach to detect and classify rice leaf diseases using a deep learning model. This dataset contains 10.776 labeled images with 5 categories which were preprocessed, augmented, and splitted 8:2 for training and validation. Using pre-trained convolutional neural network model ResNet50, The dataset was fine-tuned using transfer learning and optimized with Optuna for hyperparameter tuning. The result, ResNet50 got 0.93 precision, 0.92 recall, and 0.92 F1_score on weighted average. This project demonstrates the potential of deep learning in predicting and enabling early accurate rice disease detection, which could support many Indonesian farmers in reducing agricultural losses and improving crop yields in the future.


## 2. Introduction

### 2.1. Background

Indonesia is one of the countries where rice is their main food source, with a high level of rice consumption. According to data from the Central Bureau of Statistics, in 2024, it is estimated that each person in Indonesia consumes 1.521 kg of rice per week . In the same year, Indonesia produced 30.34 million tons of rice for consumption and imported 3.85 million tons of rice between January to November 2024. Rice production in 2024 is reduced by 757.13 thousand tons or equal to 2.3% of production in 2023.

However, despite Indonesia's significant rice consumption and production, many rice farmers struggle to earn sustainable income with their average salary of only Rp 15,199.00 per day. Meanwhile, the World Bank has reported that the price of rice in Indonesia is 20 percent higher than the price of rice in the global market. Between 2010-2017, it was even estimated that the price of rice in Indonesia was 60% more expensive than the international price. One of the main reasons is the high cost of production, which includes the rising price of fertilizers, pesticides, and equipment. In addition, Indonesia's unpredictable weather condition also further affects the rice plant to be more vulnerable to diseases such as Bacterial Blight, Brown Spot, and Leaf Smut.

Detecting rice diseases poses a unique challenge for farmers, particularly in larger rice fields where it is more difficult to detect diseases in rice plants. Rice diseases like Bacterial Blight, Brown Spot, and Leaf Smut are difficult to detect manually, therefore technology can play a crucial role in earlier disease detection and support more effective disease management before the disease spreads.

Technologies like computer vision and deep learning can help farmers detect rice leaf diseases. Deep learning as part of machine learning and computer vision has been developed to improve Artificial Intelligence performance in detection and classification images. Some deep learning models for photo detection and classification have been developed and used in many sectors to increase productivity and reduce risk, including agriculture and farming. It can detect what type of disease from the disease symptoms images on the leaf and other parts of the plant. This research study is aimed to make a new optimized deep learning model with the help of Optuna, and determine which deep learning model is compatible for rice disease detection and classification. The goal is to help farmers detect diseases more effectively, therefore improving crop health, reducing crop loss, and supporting agricultural sustainability.

## 2.2. Problem Statement

Detecting rice diseases presents a unique and critical challenge for farmers, particularly in expansive rice fields where manual identification of disease symptoms is difficult. Diseases like Bacterial Blight, Brown Spot, and Leaf Smut can spread rapidly, leading to significant crop loss before they are visually identified. The difficulty of manual, early detection compromises crop health, increases the need for costly interventions, and ultimately threatens agricultural sustainability and farmer income. Technology can play a crucial role in earlier disease detection and support more effective disease management before the disease spreads.

## 2.3. Project Objectives

This project aims to address the challenges of rice disease detection by leveraging advanced computer vision technology. The primary objective is to create an optimized deep learning model for rice disease classification using the pre-trained model ResNet50 with the help of Optuna, a hyperparameter optimization framework.

## 2.4. Significance

The utilization of deep learning and computer vision to analyze disease symptoms from images of the plant and leaves can detect the exact type of disease. This project''s goal is to help farmers detect diseases more effectively, leading to a significant improvement in crop health, a reduction in crop loss, and a vital contribution to agricultural sustainability and food security in Indonesia. The implementation of optimized deep learning models in agriculture increases productivity and reduces risk.

## 3. Literature Review

In the last 15 years, computer vision has developed quite significantly with deep learning. Deep learning has been used to solve problems in computer vision, including image classification, image classification with localization, object detection, object segmentation. The most important factors that boost deep learning development are large, high-quality, publicly available labelled datasets, along with the empowerment of parallel GPU computing, which enables faster computation with GPU based training. Compared to traditional methods, features in deep learning models can be automatically learned from big data, while we need to manually design the parameters in the traditional method to get better accuracy. Deep learning models also contain more parameters than the traditional method, making it more expressive so that it can reach higher accuracy.

As the number of computer vision in agriculture sector publication has increased and provided larger dataset, this helped farmers to implement computer vision for their crops production and proved the usefulness of computer vision for plant disease detection. More models have been created and upgraded to improve the accuracy, such as Resnet.

Residual Neural Network (ResNet), a groundbreaking deep learning structure that addresses the problem of vanishing gradient in very deep layers by introducing residual connections, which allow for the training of much deeper layers than previously possible. However, a deeper layer means gradients can diminish significantly , here ResNet's primary innovation is the use of residual blocks, which include skip connections or shortcuts that bypass one or more layers. These connections enable the network to learn residual functions rather than direct mappings, making it easier to optimize very deep networks. The two popular variants of ResNet include ResNet-50 and ResNet-152, where the number indicates the depth of the network in terms of layers.

ResNet achieved a top 1 accuracy of 78.93% on the CIFAR-100 dataset, which was better than other models such as VGG-16 and GoogLeNet. ResNet can be very deep without suffering from vanishing gradients and is great at classification tasks, but ResNet models require significant computational resources and can overfit on smaller datasets if not properly regularized.

ResNet has outperformed other architectures like VGG and Inception. For example, older publication shows ResNet-50 achieved a test accuracy of 70%, compared to 25% for VGG-16 on Diabetic Retinopathy dataset, which is a significant improvement. However, some studies have noted that ResNet can struggle with very small datasets or require careful tuning of hyperparameters to achieve optimal performance and need high compute demands, which make it slightly difficult if implemented with low computational devices like mobile phones and drones.

# 4. Methodology

## 4. 1 Dataset

The dataset is collected from Mendeley Data which consists of 10766 images with 5 categories of Rice Diseases:

- Bacterial Leaf Blight
- Rice Blast
- Rice
- Tungro
- Healthy Leaf

This dataset has high resolution images of rice leaves that exhibit symptoms of the specified disease, aiding the deep learning image recognition task while also having the symptom description in the form of textual description outlining the major symptoms visible on the leaf.

## 4. 2 Data Splitting and Augmentation

The dataset is splitted into training data and validation data with 8:2 ratio. Training data will be used to train the model and the validation data will be used to evaluate the model by using it to classify the validation data and compare with the label. This splitting process results in around 8000 training images and 2000 test data.

The dataset is then augmented to increase data amount and make the model learn from various images. Several augmentation methods were done to rescale pixel value of the image from the range [0, 255] to the range [0, 1], flip the image horizontally, rotate the images within range ± 20 degrees, and zoom it in or out by up to 20%.

## 4. 3 Preprocessing and Class Weighting

Preprocessing was done to ensure performance so that overfitting of model does not happen. The image preprocessing steps include image resizing where it resizes all images to a fixed input size which is 128x128 for ResNet50. Distribution between classes was balanced by calculating class weight. Giving more importance to the underrepresented class during model training helps to achieve balanced distribution.

## 4. 4 Model Architecture

The model used in this project is Residual Neural Network (ResNet), a deep learning structure which addresses the problem of vanishing gradient in very deep layers by introducing residual connections. This allows for the training of much deeper layers than previously possible. However, a deeper layer means gradients can diminish significantly. ResNet's primary innovation is the use of residual blocks, which include skip connections or shortcuts that bypass one or more layers. These connections enable the network to learn residual functions rather than direct mappings, making it easier to optimize very deep networks. The ResNet variant used in this project is ResNet50 which consists of 50 trainable layers excluding the first convolutional layers and the classification layer.

ResNet50 layers are structured into five stages, including the initial input processing.

| Stage | Output Size | Blocks | Description |
|---|---|---|---|
| Stage 1 (Input) | 112 x 112 | 1 Convolution , 1 MaxPool | Initial 7 x 7 convolution and 3 x 3 max-pooling to reduce spatial dimensions and extract low-level features |
| Stage 2 | 56 x 56 | 3 Bottleneck | 3 residual blocks, maintaining input size |
| Stage 3 | 28 x 28 | 4 Bottleneck | 4 residual blocks, the first block uses a stride of 2 to downsample the spatial size |
| Stage 4 | 14 x 14 | 6 Bottleneck | 6 residual blocks, the first block downsamples. |
| Stage 5 | 7 x 7 | 3 Bottleneck | 3 residual blocks, the first block downsamples. |
| Output | 1 | 1 Average Pool, 1 FC | Global average pooling converts 7 x 7 x 2048 feature maps into 2048 dimensional vector, which is fed into the fully connected layer for classification |

## 4. 5 Training Setup

The training process was configured to ensure efficient convergence and optimal model performance, leveraging the power of transfer learning and automated hyperparameter optimization via Optuna.

Instead of training the ResNet50 model from scratch, we implemented transfer learning. The model's initial weights were loaded from a checkpoint that was trained on the ImageNet dataset. The early layers are frozen to preserve the high quality feature maps learned from ImageNet.

Then the model was trained on the rice disease dataset utilizing hyperparameter optimization Optuna. The parameters that were optimized are:

- Optimizer Selection = Adam, SGD
- Dropout rate = 0.3, 0.4, 0.5, 0.6
- Dense Unit = 256, 512
- Learning Rate = 1e-4, 1e-5, 1e-6

- Unfrozen Layer = 0, 20, 40, 60, 80, 100

We gave Optuna 12 trials to find the best parameters combination based on the training evaluation metrics

### 4.6 Evaluation Metrics

Our evaluation metrics were divided into two sets, primary metrics and secondary metrics.

A. Primary Metrics:
   Used for evaluating model during training process
   - Accuracy: Indicates overall ratio of correct predictions
     $Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Number\ of\ Instances}$
   - Categorical Cross Entropy: Measures the difference between the model's predicted probability distribution and the true one-hot encoded label.
     $L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$

B. Secondary Metrics:
   Used for evaluating the model after the best hyperparameters combination were found. These metrics were derived from the confusion matrix
   - Precision: Measures reliability of positive predictions
     $Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$
   - Recall: Measures model's ability to detect all actual disease cases
     $Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$
   - F1-score (Weighted): Calculates mean of Precision and Recall. This is calculated for each class and then weighted by support.
     $F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$

## 5. Implementation & Results

In this study, we will be using the premade architecture, ResNet50 to classify the data on our dataset into 5 classes: "Bacterial Leaf Blight", "Healthy leaf", "Rice", "Rice Blast", and "Tungro". Our model will be tuned using Optuna to find the optimal combination of the previously listed hyperparameters.

### 5.1 Training Result

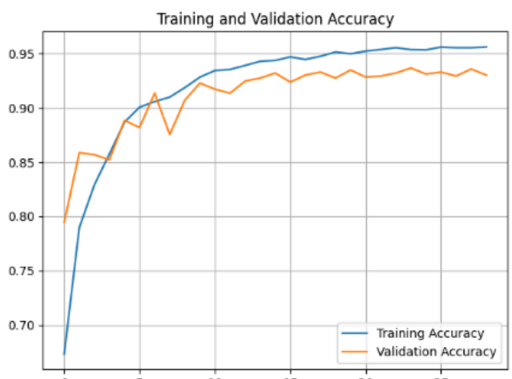
After performing 12 Optuna trials, we found the optimal hyperparameters that yield the highest validation performance for our model.

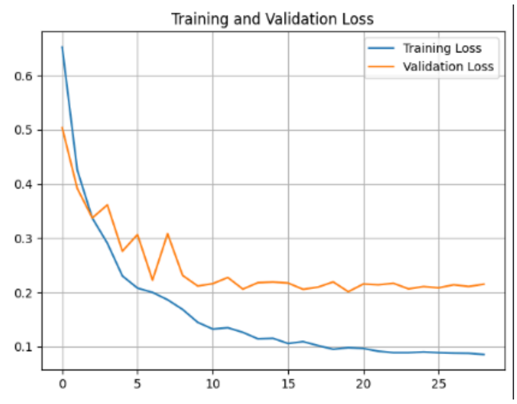
| Hyperparameter | Optimal Value |
|---|---|
| Epochs | 15 |

| | |
|---|---|
| Batch Size | 32 |
| Dense Units | 512 |
| Optimizer | Adam |
| Learning Rate | 1e-4 |
| Dropout Rate | 0.3 |
| Unfrozen Layers | 40 |

Using these documented values to fit the model, we were able to achieve a high training accuracy of 95.44%, paired with a low training loss (categorical cross entropy) of 0.0897. These numbers indicate a strongly fitting model on the training set. This performance is validated by the evaluation scores that we've achieved on the validation set. That is the validation accuracy of 93.03% and the validation loss of 0.2150. The close proximity between the evaluation scores achieved on the training and the validation set suggests that the model has a strong generalization ability, does not indicate an overfit on any significant measure, and will perform well on the unseen test set.

Training and Validation Accuracy in Each Epoch

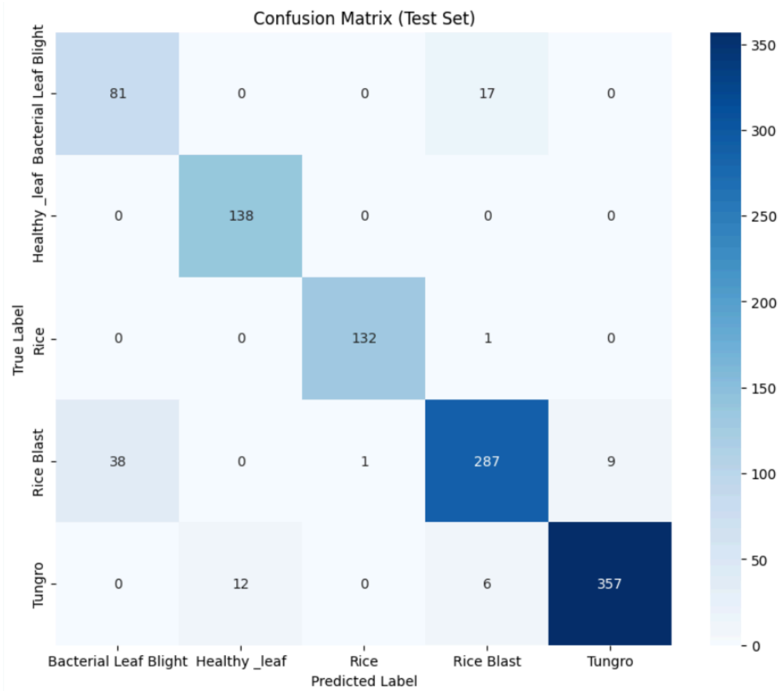



Training and Validation Loss in Each Epoch

### 5.2 Model Evaluation on Test Set

After fitting, the model is then evaluated on the test set. To gain a granular insight on how the model performs its classification on each class, we constructed a Confusion Matrix using the predictions the model made on the dedicated test set.




Confusion Matrix on the Test Set

The figure above visualizes the model's overall efficacy, as well as highlighting the points of strength and weakness of the model. The model demonstrates exceptional performance in identifying "Healthy leaf" (138/138 correct prediction), "Rice" (132/133 correct prediction), and "Tungro" (357/375 correct prediction). Among those listed, the remarkable accuracy/recall score of 100% on the "Rice" class was able to be achieved as it has unique features, such as color and pattern that gave it a noticeable visual difference when compared to the other classes.

Meanwhile, on tasks involving the "Rice Blast" class and the "Bacterial Leaf Blight" class, the matrix identified a major pattern in misclassification. A significant number of "Rice Blast" images were incorrectly predicted as "Bacterial Leaf Blight" (38/335 misidentified as "Bacterial Leaf Blight"). A similar trend is also identified on "Bacterial Leaf Blight" images, where a portion of its instances were incorrectly predicted as "Rice Blast" (17/108 misidentified as "Rice Blast"). This mutual confusion between the "Bacterial Leaf Blight" and "Rice Blast" instances suggests a striking visual or feature ambiguity between these two diseases.

The deep learning model performed strongly, achieving an F1 score, a recall score of 0.92, and precision of 0.93 as derived from the confusion matrix. This consistency indicates a well-balanced, robust model with high predictive power and operational efficiency, effectively minimizing false negatives.

## 6. Discussion & Limitations (analysis of performance, challenges, trade-offs)

### 6.1 Analysis of performance

Using Optuna hyperparameter tuning, the ResNet50 model achieved its best performance with the following configuration: a batch size of 32, 512 dense units, the Adam optimizer with a learning rate of 0.0001, a dropout rate of 0.3, and the last 40 layers unfrozen. This configuration yielded high classification accuracy, demonstrating its suitability for rice leaf disease identification. Tested on a test dataset, the model successfully achieved 0.92 for accuracy, weighted recall, and weighted f1-score, and 0.93 for weighted precision. Overall, the model successfully maintained a high score for all metrics meaning it is robust.

### 6.2 Challenges

The challenge that we faced is the quality and variability, while the dataset is augmented, the dataset is captured in a semi-controlled environment. The challenge is that there is a gap between the quality of a semi controlled environment and a real case environment where farmers often take a picture with poor lighting, blurry, and even complex backgrounds such as soils.

### 6.3 Trade-offs

For the model we selected the ResNet50 model which has heavier architectures compared to other models such as MobileNet and EfficientNet-B0. With this heavier architecture, it provided a far superior feature extraction capabilities and higher stability for image classification. But the tradeoff is the model is computationally heavier and has a larger memory footprint. The h5 model is even almost 300mb, making it less ideal for direct deployment on low-end mobile devices without further optimization.

## 7. Conclusion & Future Work

This report demonstrates that ResNet50 yields a high performance on the dataset achieving 93.03% Accuracy Score. This model was optimized using Optuna, which significantly increased the model performance on the given dataset. With the automation of key hyperparameters, Optuna enabled the ResNet50 model to reach its optimal configuration. The ResNet50 Model has the ability to mitigate vanishing gradient problems and retain subtle details due to its deep architecture combined with skip connection. This model's performance can still be improved with longer training and deeper tuning.

Future work should extend this study by including a more diverse dataset of rice disease to enhance robustness across real-world scenarios. Practical deployment strategies such as integrating models into drone systems may also be considered for real time rice disease detection.

## 8. References

1. Fitrawaty, H., W., Yusuf, M., Maipita, I.: A simulation of increasing rice price toward the disparity of income distribution: An evidence from Indonesia. Heliyon 9(3), e13785 (2023). https://doi.org/10.1016/j.heliyon.2023.e13785
2. Badan Pusat Statistik: Luas Panen dan Angka Produksi Padi di Indonesia 2024 (Angka Sementara). Berita Resmi Statistik, (2024). https://assets.dataindonesia.id/2024/10/17/1729129742184-81-17.-Berita-Resmi-Statistik---Luas-Panen-dan-Produksi-Padi-2024.pdf, Accessed 01 Mar 2024.
3. Metrotvnews.com: Produksi Beras Nasional Turun di 2024. (2024). https://www.metrotvnews.com/read/bzGCzlWa-produksi-beras-nasional-turun-di-2024, Accessed 05 Mar 2025.
4. Nugroho, R.S.: Harga Beras di Indonesia Mahal tapi Pendapatan Petani Rendah, Apa Penyebabnya? KOMPAS.com (2024). https://www.kompas.com/tren/read/2024/09/23/170000665/harga-beras-di-indonesia-mahal-tapi-pendapatan-petani-rendah-apa, Accessed 05 Mar 2025.
5. Mukhlis, M., Ismawati, I., Sillia, N., Fitrianti, S., Ukrita, I., Wisra, R.F., Raflis, H., Hendriani, R., Hanum, L., Ibrahim, H., Nofianti, S., Marta, A., Sari, N.: Characteristics of Production Factors and Production of Zero Tillage System Rice Farming. Jurnal Penelitian Pendidikan IPA 10(8), 6013–6019 (2024). https://doi.org/10.29303/jppipa.v10i8.8542
6. Patil, R.R., Kumar, S.: Rice-Fusion: A Multimodality Data Fusion Framework for Rice Disease Diagnosis. IEEE Access 10, 5207–5222 (2022). https://doi.org/10.1109/access.2022.3140815
7. Yuan, L., et al.: Florence: A New Foundation Model for Computer Vision. arXiv preprint arXiv:2111.11432 (2021). https://doi.org/10.48550/arxiv.2111.11432
8. Motylinski, M., MacDermott, Á., Iqbal, F., Shah, B.: A GPU-based Machine 8. Learning Approach for Detection of Botnet Attacks. Computers & Security, 102918 (2022). https://doi.org/10.1016/j.cose.2022.102918
9. Yuan, Y., Chen, L., Wu, H., Li, L.: Advanced agricultural disease image recognition technologies: A review. Information Processing in Agriculture 9, 14-34 (2021). https://doi.org/10.1016/j.inpa.2021.01.003
10. Alzubaidi, L., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data 8(1) (2021). https://doi.org/10.1186/s40537-021-00444-8
11. Yakkundimath, R., Saunshi, G., Anami, B., Palaiah, S.: Classification of Rice Diseases using Convolutional Neural Network Models. *J. Inst. Eng. India Ser. B*. 103(3), 1029–1038 (2022). https://doi.org/10.1007/s40031-021-00704-4
12. Ritharson, P.I., Raimond, K., Mary, X., Eunice, R.J., Andrew, J.: DeepRice: A deep learning and deep feature based classification of Rice leaf disease subtypes. *Artif. Intell. Agric*. 10, 106–122 (2023). https://doi.org/10.1016/j.aiia.2023.11.001
13. Chen, Y., Li, L., Li, W., Guo, Q., Du, Z., Xu, Z.: Deep learning processors. In: Guo, Q., Li, W., Li, L., Chen, Y., Du, Z., Xu, Z. (eds.) *Deep Learning Processors*, pp. 207–245. Elsevier, Amsterdam (2023). https://doi.org/10.1016/b978-0-32-395399-3.00012-3

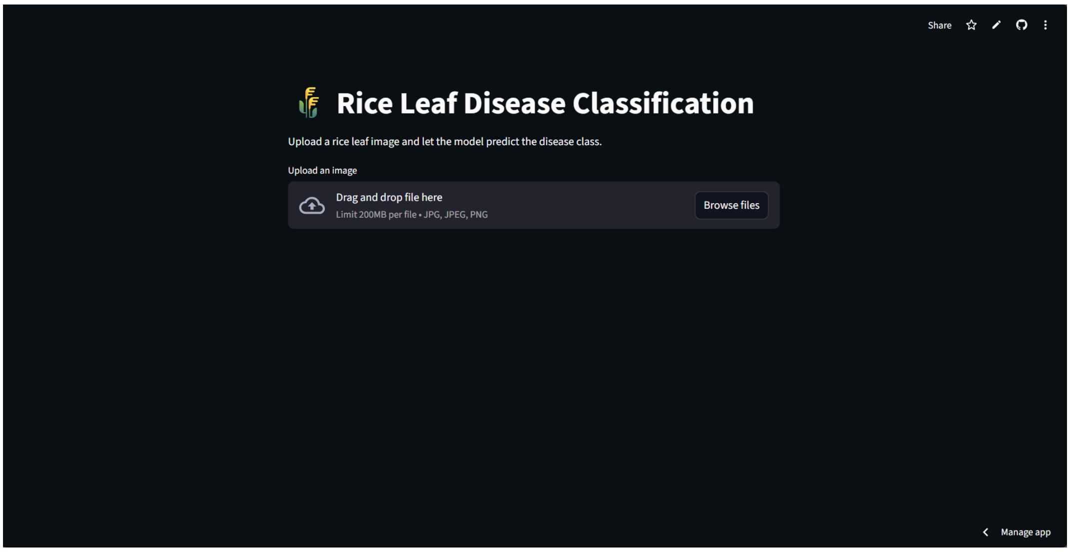
14. Latif, G., Abdelhamid, S.E., Mallouhy, R.E., Alghazo, J., Kazimi, Z.A.: Deep Learning Utilization in Agriculture: Detection of Rice Plant Diseases Using an Improved CNN Model. Plants 11(17), 2230 (2022). https://doi.org/10.3390/plants11172230
15. Sarwinda, D., Paradisa, R.H., Bustamam, A., Anggia, P.: Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer. Procedia Computer Science 179, 423–431 (2021). https://doi.org/10.1016/j.procs.2021.01.025
16. Su, H.: AdaResNet: Enhancing Residual Networks with Dynamic Weight Adjustment for Improved Feature Integration. arXiv preprint arXiv:2408.09958 (2024). https://doi.org/10.48550/arxiv.2408.09958
17. Xu, G., Wang, X., Wu, X., Leng, X., Xu, Y.: Development of Skip Connection in Deep Neural Networks for Computer Vision and Medical Image Analysis: A Survey. arXiv preprint arXiv:2405.01725 (2024). https://doi.org/10.48550/arxiv.2405.01725
18. Dhanalakshmi, S., Arulselvi, S.: Using ResNet architecture with MRI for classification of brain images. Indonesian Journal of Electrical Engineering and Computer Science 39(1), 148–158 (2025). https://doi.org/10.11591/ijeecs.v39.i1.pp148-158
19. K., M., G., A., Revathy, G., Senthilvadivu, S., Venkateswaran, S.: Beyond Depth: Evaluating ResNet-50 and ResNet-152 Performance on Cifar-10 Dataset. In: 2024 International Conference on Sustainable Communication Networks and Application (ICSCNA), pp. 973–977 (2024). https://doi.org/10.1109/icscna63714.2024.10863975
20. Roy, P.S., Kukreja, V.: Vision transformers for rice leaf disease detection and severity estimation: a precision agriculture approach. Journal of the Saudi Society of Agricultural Sciences 24(3), (2025). https://doi.org/10.1007/s44447-025-00007-w

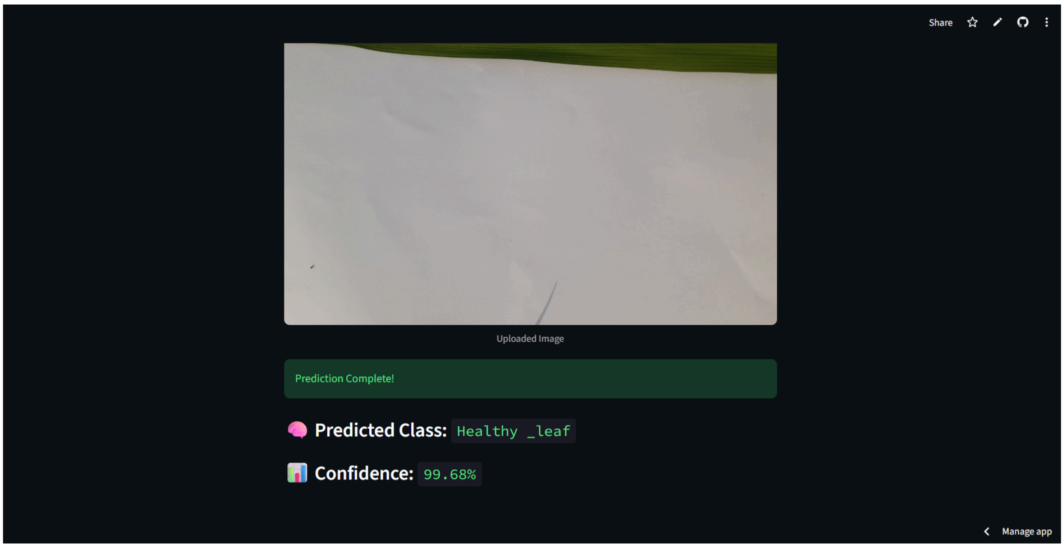
## 9. Appendix (team contribution statement, screenshots, code snippets)

### 9.1. Team Contribution Statement

| Name | NIM | Contribution |
|---|---|---|
| Okky Sudibyo Rades | 2702300575 | Building and Training ResNet50 Model |
| Azka Farrel Keandra Aristyan | 2702298652 | Writing Final Report Paper |
| Marcello Evan Wijaya | 2702266026 | Writing Final Report Paper |

### 9.2. Screenshots




## 9.3. Code Snippets

## Data Preprocessing

```
train_datagen = ImageDataGenerator(
    preprocessing_function=preprocess_input,
    rotation_range=30,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True,
    brightness_range=[0.8, 1.2],
    fill_mode='nearest'
)

val_test_datagen = ImageDataGenerator(preprocessing_function=preprocess_input)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=True
)

val_generator = val_test_datagen.flow_from_directory(
    val_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)
```

```
test_generator = val_test_datagen.flow_from_directory(
    test_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    shuffle=False
)

class_weights_vals = class_weight.compute_class_weight(
    class_weight='balanced',
    classes=np.unique(train_generator.classes),
    y=train_generator.classes
)
class_weights = dict(enumerate(class_weights_vals))

print(f"Data Split Complete.")
print(f"Train images: {train_generator.samples}")
print(f"Val images: {val_generator.samples}")
print(f"Test images: {test_generator.samples}")
```

## Building and Training Model

```python
# Best params = {'dropout_rate': 0.3, 'dense_units': 512, 'optimizer': 'adam', 'learning_rate': 0.0001, 'unfrozen_layers': 40}
BEST_DROPOUT = 0.3
BEST_DENSE_UNITS = 512
BEST_LR = 0.0001
UNFROZEN_LAYERS = 40
```

```python
base_model = ResNet50(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
num_classes = len(train_generator.class_indices)

for layer in base_model.layers:
    layer.trainable = False

for layer in base_model.layers[-UNFROZEN_LAYERS:]:
    if not isinstance(layer, tf.keras.layers.BatchNormalization):
        layer.trainable = True

x = GlobalAveragePooling2D()(base_model.output)
x = Dense(BEST_DENSE_UNITS, activation='relu')(x)
x = Dropout(BEST_DROPOUT)(x)
output = Dense(num_classes, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=output)
```

```python
optimizer = tf.keras.optimizers.Adam(learning_rate=BEST_LR)

model.compile(
    optimizer=optimizer,
    loss='categorical_crossentropy',
    metrics=['accuracy']
)

callbacks = [
    EarlyStopping(patience=5, monitor='val_accuracy', restore_best_weights=True),
    ReduceLROnPlateau(monitor='val_accuracy', factor=0.5, patience=2, min_lr=1e-6)
]

history = model.fit(
    train_generator,
    epochs=100,
    validation_data=val_generator,
    class_weight=class_weights,
    callbacks=callbacks,
    verbose=1
)
```

## 9.4. Important Links

Dataset Source : Rice Leaf and Crop Disease Detection Dataset - Mendeley Data
Github Link : lowwwkyy/ResNet50_Optuna_Rice-Disease-Classification
Deployed Web : Rice Leaf Disease Classifier · Streamlit
Demo Video : video_demo.mp4