SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 1, December 2024, pp.132 - 141
ISSN 2407-0939 print/ISSN 2721-2041 online

Detection of Rice Malnutrition Based on Leaf Imagery with the Convolutional Neural Network (CNN) Algorithm

Denny Pribadi[1], Desi Susilawati[2], Erika Mutiara[3]
[1.2] Informatika, [3]Sistem Informasi, Faculty of Teknik dan Informatika,
Universitas Bina Sarana Informatika
Jl. Cemerlang No. 8 Sukakarya Kota Sukabumi
Email: denny.dpi@bsi.ac.id, desi.dlu@bsi.ac.id, erika.emb@bsi.ac.id

ABSTRACT

Rice is a plant that has an important role in meeting food needs in Indonesia. However, the growth and production of rice plants can be disrupted by several factors, including environmental conditions and plant health. Malnutrition of rice plants is a serious problem that can reduce crop yields and rice quality. This research aims to explore and identify malnutrition of rice leaves and evaluate its impact on farmers. In this research, researchers used the CNN algorithm as a framework to create a better identification model. Through this research, this research hopes to provide a deeper understanding of malnutrition in rice plants and provide valuable insights for farmers. Researchers also attempt to offer solutions or recommendations that can be implemented to overcome this problem. To solve the problems raised, researchers collected and analyzed data from various sources, including Kaggle. Researchers also researched agricultural offices to gain diverse perspectives on this issue. The research results show that the accuracy level of the identification model is 98.89% and contributes to the general understanding of rice leaf malnutrition. Researchers hope that these findings can encourage further discussion and relevant action in the context of dealing with malnutrition in plants. Researchers realize that every problem has certain limitations and limitations. Therefore, researchers suggest that further research be carried out to deepen understanding and overcome obstacles found during this research.

Keywords: *Up to five keywords should also be included*

Introduction

Malnutrition or plant nutritional deficiency is a reduction in the availability of essential chemical elements, resulting in deviations in plant development. This happens because vegetation requires certain chemical elements at different plant phenology levels in certain volumes. Proper vegetation development requires a balance of chemicals and their amounts. [1]. Technological developments affect all aspects of life. Currently, technology is widely used in industry, health, agriculture, and other areas of life. For example, the use of technology in the agricultural sector, from searching for superior varieties to planting systems and harvesting processes. One of the plants that is the focus of industry, science, and research and has repeatedly produced technological innovation is rice [2] is an important food crop, especially in Indonesia. Rice cultivation is the third among all cereals after corn and wheat. As the population increases, the need for rice production as a staple food must increase [3]. In the agricultural sector, especially in rice, many people are interested in studying the problems faced by farmers. In previous research, rice plants were widely used as research subjects with various existing methods, some used the identification method using the ANFIS method, some used GLCM, some also used EfficientNet B3 and all used the CNN algorithm. Previous studies show that CNN is more accurate and has high image recognition accuracy.

CNN has been shown to provide better results in image recognition tasks compared to traditional methods such as SVM (Support Vector Machine) or KNN (K-Nearest Neighbors), especially in terms of accuracy. Its ability to handle large-scale and high-complexity visual data makes it a primary choice in applications such as face recognition, object recognition, and image classification [4].

Machine learning is a branch of artificial intelligence that is used to search for information in a dataset to support decision-making. Some say that machine learning is a collection of computer algorithms that can be used to optimize computer work based on existing samples [5].The use of deep learning can be applied to various tasks, such as predicting possibilities and events, recognizing objects, and diagnosing diseases. Image processing systems aim to help people identify or classify objects efficiently, quickly, and accurately, and can process large amounts of data simultaneously. Several algorithms used in the field of image processing are Naive Bayes, Support Vector Machine, and Neural Networks. Convolutional Neural Network (CNN) is an algorithm for developing neural networks. The CNN algorithm has the most significant results in digital image recognition because CNN is implemented based on the image recognition system in the human visual cortex [6], [7]. This is also supported by research showing that CNN is the best model for object detection and identification problems. Technically, a CNN is a trainable architecture consisting of several steps. The input and output of each step consist of several arrays called feature maps. CNN is a combination of image convolution which works in the feature extraction process and neural networks which work in classification [3].

Malnutrition in rice plants can cause crop failure, reducing rice production and farmer income. Especially in severe cases, it is difficult to distinguish the symptoms of deficiency. Detecting malnutrition automatically requires collaboration with other fields, especially computer science, so that farmers can act to care for plants and quickly contain the spread of disease. Identifying malnutrition from images requires the best methods for identifying the disease [8].This research aims to determine malnutrition in rice plants by detecting images of rice plant leaves using machine learning and convolutional neural network (CNN) algorithms based on images of rice plant leaves.

Research Methods

Designing research objects is an important process to obtain research results. The data in this research are pictures or images of leaves obtained from Kaggle and the results of interviews with the head of the Sukabumi City Food and Agriculture and Fisheries Department. In this research, The data used were 240 images of rice plants with the categories blast, blight, and tungro. In order for this research to be more focused,a model was designed to identify rice plants [2], [8], [9], [10].

In the process of developing a leaf identification model, a series of processes are carried out including data preparation, data preprocessing, model training, and evaluation, the explanation is as follows:

1. Data Preparation, the data preparation process includes several steps, namely [13]:
   a. Data Extraction
   The first step is to take rice leaf image data from the Kaggle platform. This data is initially saved in compressed RAR format. Accessing the contents of RAR files requires software that supports the RAR format to open and extract the contents [3], [4], [5], [14].
   b. Import Libraries
   Import Library retrieves and uses functions, classes, or objects that have been defined in a library or module in a Python program. The benefits and advantages of import libraries include reading and using existing code, increasing efficiency and productivity, providing additional functionality, and organizing code better. Import Library is the first step in using community-developed algorithms and functions to build machine learning models. By importing appropriate libraries, you can run various machine-learning algorithms, process data, evaluate models, and perform other machine-learning tasks. There are several libraries used in developing rice leaf plant identification models, namely as follows [6], [6], [7], [15]:

```
import tensorflow as tf
import os
```

Figure 1. Syntax for importing the library used

1) TensorFlow
TensorFlow is an open-source framework developed by Google and used to create and train machine learning models. TensorFlow provides advanced tools and APIs for developing various machine learning models, especially deep learning models.
2) OS (operating system)
Os (operating system) in Python is used to interact with the operating system (OS) that is running on the computer. This library provides various functions that allow performing operations on files and directories, accessing environment variables, executing system commands, and much more.
3) Matplotlib
Matplotlib is very useful for visualizing data in the form of graphs and diagrams. With its versatile features and flexibility, this library is the best choice for developers, data scientists, and researchers to create creative and informative data visualizations and communicate information more effectively. This library offers a variety of functions and tools for creating graphs, plots, and data visualization.
4) Loud
Keras is a popular Python programming language library used to build and train deep learning models. Keras provides a complex and easy-to-use interface for developing various neural networks quickly and efficiently. This library is suitable for various tasks such as pattern recognition, classification, regression, natural language processing, and many others.

2. Data Preprocessing

Before training a CNN, the first step is to carry out data preprocessing. At this stage, the rice leaf image is converted into a format that can be processed by a neural network, such as reducing the image size, normalizing pixel intensity, and cropping or smoothing the image if necessary.

```
tf.keras.layers.BatchNormalization(input_shape=(150, 150, 3)),
```

Figure 2. Syntax for Normalizing Pixel Intensity

3. CNN architecture

```
  tf.keras.layers.Conv2D(32, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(64, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Conv2D(128, 3, activation='relu'),
  tf.keras.layers.MaxPooling2D(),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(256, activation='relu'),
  tf.keras.layers.Dense(38, activation= 'softmax')
])
```

Figure 3. CNN architecture

CNN consists of several layers that play an important role in image processing. The main layers in CNN are convolutional, pooling layers, dropout layers, flatten layers, and fully connected layers.

a. Convolution Layer (Conv2D)
The convolution layer is a layer that functions to extract visual features from rice leaf images. This layer uses a number of filters that are applied to the image in stages to detect relevant patterns. Each filter will produce a "feature map" that represents the features found in the image.

b. Pooling Layer (Maxpooling)
After the convolution layer, followed by the pooling layer. The pooling layer aims to reduce the spatial dimensions of the features extracted by the convolution layer. The commonly used pooling method is max pooling, where only the maximum value in each "window" is retained, while the other values are ignored

c. Dropout Layer
The main purpose of dropout is to reduce model overfitting. Overfitting occurs when the model “remembers” too much of the training data and does not generalize well to new data. Dropout randomly “turns off” (skips) multilayer units (neurons) during training. In this way, dropout forces the model to rely less on individual units and forces the model to learn more robust and general features.

d. Flatten Layer
The flattened layer acts as a bridge between the convolutional layer and the fully connected layer in a convolutional neural network (CNN). After going through several convolution and pooling layers, the Flatten layer transforms the data representation into a one-dimensional vector. This allows the feature extraction performed by the convolution layers to be combined into a fully connected layer, consisting of dense layers.

e. Fully connected (Dense) Layer
The fully connected (FC) layer is the most commonly used in neural networks. This layer is also called the dense layer or linear layer. In a fully connected layer, each neuron of that layer is connected to every neuron of the previous and subsequent layers. In other words, each neuron in this layer receives input from all neurons in the previous layer and sends its output to all neurons in the next layer. The most important feature of the fully connected layer is that each neuron in the layer has a weight and a bias. In a fully connected layer, the activation of each neuron can be calculated using the ReLU activation function. At the end of this layer, there is also an additional activation function such as Softmax to generate class probability distributions in multi-class classification tasks.

4. CNN Training

```
fitting_history = model.fit(
                    train_set,
                    epochs=10,
                    validation_data=test_set,
                    verbose=1,
                    callbacks = [reminderCB,checkpointCB],
                  )
```

Figure 4. CNN Training

To train the CNN, a dataset containing rice leaf images with labels indicating the health status of the leaves (healthy or malnourished) is required. These images are used to train the neural network to recognize patterns associated with malnutrition. Training involves iterating through the training dataset, where the weights and parameters in the CNN are automatically adjusted using a learning algorithm to minimize prediction errors. The output of the training model for rice leaf plant identification using the CNN algorithm is as follows:

```
Epoch 10/10
105/105 [==============================] - ETA: 0s - loss: 0.0664 - accuracy: 0.9818
Target reached 96.00%. Stop Training
105/105 [==============================] - 495s 5s/step - loss: 0.0664 - accuracy: 0.9818 - val_loss: 0.0595 - val_accuracy: 0.
9822
```

Figure 5. Epoch Syntax Output Results

5. Model Evaluation

After going through the training phase, the CNN is evaluated using a separate testing dataset. The testing images are used to test the model's ability to identify malnutrition in rice leaves. Commonly used evaluation metrics are accuracy, precision, recall, and F1-score, which describe the extent to which the model can make correct predictions.

Results and Discussion

This study discusses the model trained in handling malnutrition, namely a condition where disease attacks cause a decrease in the quality and health of rice plants which results in reduced yields for farmers. The implementation of the deep learning method using the CNN algorithm on the rice leaf malnutrition identification model provides a promising level of accuracy. In this study, rice leaf image data with various levels of malnutrition were collected and used to train the CNN model. The processing process begins with the collection of datasets consisting of healthy rice leaf images and malnourished rice leaves. The dataset is then split into training data and test data. Next, load the training image dataset and test dataset from the directory on the file system and produce dataset objects that are ready to be used for model evaluation or testing.

```
In [5]: model = tf.keras.models.Sequential([
          tf.keras.layers.BatchNormalization(input_shape=(150, 150, 3)),
          tf.keras.layers.Conv2D(32, 3, activation='relu'),
          tf.keras.layers.MaxPooling2D(),
          tf.keras.layers.Conv2D(64, 3, activation='relu'),
          tf.keras.layers.MaxPooling2D(),
          tf.keras.layers.Conv2D(128, 3, activation='relu'),
          tf.keras.layers.MaxPooling2D(),
          tf.keras.layers.Dropout(0.2),
          tf.keras.layers.Flatten(),
          tf.keras.layers.Dense(256, activation='relu'),
          tf.keras.layers.Dense(38, activation= 'softmax')
        ])
```

Figure 6. CNN architecture

After that, in Figure 6, the CNN architecture is created. The CNN model is built using convolutional layers to extract features from rice leaf images. The convolution layer is followed by a Batch Normalization layer to apply batch normalization to the output of the Conv2D layer. Then, the training process will involve calculating the mean and variance on each batch of data to ensure the data remains within the normalization range and is then followed by an activation layer and a pooling layer to reduce the dimensionality of the data. Then the extracted features are forwarded to the fully connected layer which acts as the final classifier.

```
 Layer (type)                Output Shape              Param #
=================================================================
 batch_normalization (BatchN  (None, 150, 150, 3)      12
 ormalization)

 conv2d (Conv2D)             (None, 148, 148, 32)      896

 max_pooling2d (MaxPooling2D  (None, 74, 74, 32)       0
 )

 conv2d_1 (Conv2D)           (None, 72, 72, 64)        18496

 max_pooling2d_1 (MaxPooling  (None, 36, 36, 64)       0
 2D)

 conv2d_2 (Conv2D)           (None, 34, 34, 128)       73856

 max_pooling2d_2 (MaxPooling  (None, 17, 17, 128)      0
 2D)

 dropout (Dropout)           (None, 17, 17, 128)       0

 flatten (Flatten)           (None, 36992)             0

 dense (Dense)               (None, 256)               9470208

 dense_1 (Dense)             (None, 38)                9766

=================================================================
Total params: 9,573,234
Trainable params: 9,573,228
Non-trainable params: 6
```

Figure 7. Leaf Image Weight and Bias Update

During the training process, in Figure 7 the CNN model receives training data and iteratively updates its weights and biases to improve its performance in classifying rice leaf images. This process involves calculating gradients and fitting model parameters using a learning algorithm called backpropagation.

```
Epoch 50/50
105/105 [==============================] - ETA: 0s - loss: 0.0145 - accuracy: 0.9950
Target reached 99.00%. Stop Training
105/105 [==============================] - 541s 5s/step - loss: 0.0145 - accuracy: 0.9950 - val_loss: 0.0016 - val_accuracy:
0.9995
```

Figure 8. Model Training

Then, in Figure 8 model. fit is used to train the model using the train_set training data and validate it on the val_set validation data for 50 epochs. After training the model, the model is evaluated with unprecedented test data. The evaluation is done by calculating performance metrics such as accuracy, precision, and sensitivity. The results of this analysis explain how well the CNN model can detect malnutrition in rice leaves. The results of processing the application of the deep learning method with the CNN algorithm show great potential in detecting malnutrition in rice leaves. With this model, farmers or researchers can quickly and accurately detect malnutrition in rice leaves, which can help take appropriate actions to improve plant health and yields.

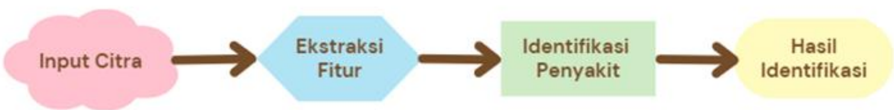


Figure 9. Implementation of Leaf Identification Model

How the rice plant leaf identification model works is as follows:

1. Input rice leaf image
   Input the image of the leaf to be identified to the model as input. Examples of images to be input each have a disease label:


Bacterial leaf blight


bacterial leaf line


bacterial panicle blight


Blas


brown spots


Healthy

Figure 10. Example of a rice picture

2. Ekstraksi Fitur

```
def load_model():
  model = tf.keras.models.load_model('model_best.h5')
  return model
```

Figure 11. Feature Extraction or Load Model

This model uses the extraction steps learned during training to extract key visual features from rice leaf images. These features include information about the structure, shape, and color of the leaf.

3. Identification of Leaf Diseases

```
def predict_image(path):
  img = tf.keras.utils.load_img(
    path, target_size=(150, 150)
  )

  img_array = tf.keras.utils.img_to_array(img)
  img_array = tf.expand_dims(img_array, 0)

  predictions = model.predict(img_array)
  score = tf.nn.softmax(predictions[0])

  return np.argmax(predictions)
```

Figure 12. Creating the Model Identification Function

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 1, December 2024, pp.132 - 141
ISSN 2407-0939 print/ISSN 2721-2041 online

Once the features are extracted, the model uses a previously learned algorithm to identify which type of leaf disease corresponds to the given image. The model compares the image features to the patterns examined during training to determine the most appropriate identification.

4. Creating Identification Labels

Table 1. Malnutrition Identification Labels

Label	Keterangan
"0"	Lack of agrept, bactocyn, plantomycyn, etc. For maximum results, it can be mixed with fungicides containing copper hydroxide, for example, nordic or Kocide.
"1"	Deficiency of Antagonistic Bacteria CorynebacteriumSp.
"2"	Deficiency of the element Oxolinic acid (5-ethyl-5,8-dihydro-8-oxo-[1,3]dioxolo[4,5-g]quinoline-7-carboxylic acid, Starner).
"3"	Deficiencies of elements Straw, manure, green manure, KCl, wood ash, scouring ash, rice husk ash, isolate/strain.
"4"	Lack of Rabcide 50 WP elements, K elements, antracol, dithane, and fungicides (score, anvil, folicur, Nativo, opus, indar).
"5"	Lack of granular insecticide elements (which contain the active ingredient carbofuran), and spray/liquid insecticides (which contain the active ingredients spinetoram, chlorantraniliprole, and dimehipo).
"6"	Lack of fungicide element (Ethaboxam 100)
"7"	Lack of prophylactic insecticide elements
"8"	Normal (healthy)
"9"	Lack of granular insecticide elements 6 kg/500 m2

```
def cast_label(result):
    with open('label.json', 'r') as file:
        labels = json.load(file)

    for key, value in labels.items():
        if str(result) == key:
            return value
```

Figure 13. Creating a Label Input Command Using JSON

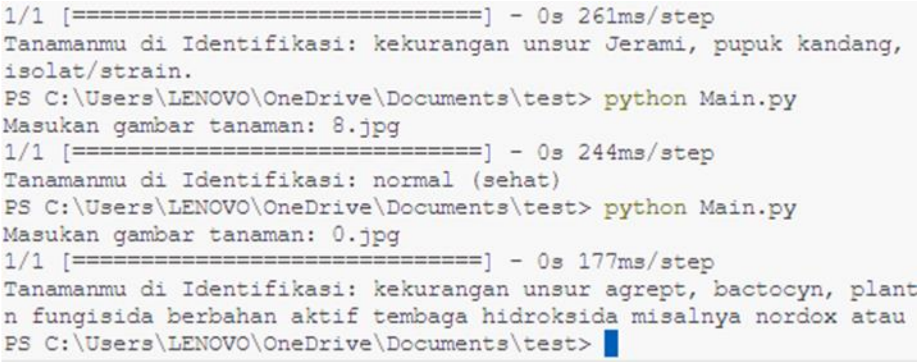
Here is a label.json file that has number labels 0 to 9.

```
{
"0": "kekurangan unsur agrept, bactocyn, plantomicyn, dll. Untuk
hasil yang maksimal bisa dicampur dengan fungisida berbahan aktif
tembaga hidroksida misalnya nordox atau kocide.",
"1": "kekurangan unsur Bakteri Antagonis CorynebacteriumSp.",
"2": "kekurangan unsur Asam oksolinat (5-etil-5,8-dihydro-8-oxo-[1,
3]dioxolo[4,5-g]quinoline-7-carboxylic acid, Starner).",
"3": "kekurangan unsur Jerami, pupuk kandang, pupuk hijau, KCl, abu
bakaran kayu, abu gosok, abu sekam, isolat/strain.",
"4": "kekurangan unsur Rabcide 50 WP, unsur K, antracol, dithane,
dan fungisida (score, anvil, folicur, Nativo, opus, indar).",
"5": "kekurangan unsur Insektisida butiran (yang mengandung bahan
aktif karbofuran), dan Insektisida semprot/cair (yang mengandung
bahan aktif spinetoram, klorantraniliprol, dan dimehipo).",
"6": " kekurangan unsur fungisida (Ethaboxam 100).",
"7": "kekurangan unsur insektisida profilaksis.",
"8": "normal (sehat)",
"9": "kekurangan unsur insektisida butiran 6 kg/500 m2."
}
```

Figure 14. Label.json file

SITEKIN: Jurnal Sains, Teknologi dan Industri, Vol. 22, No. 1, December 2024, pp.132 - 141
ISSN 2407-0939 print/ISSN 2721-2041 online

5. Identification Results



```
1/1 [==============================] - 0s 261ms/step
Tanamanmu di Identifikasi: kekurangan unsur Jerami, pupuk kandang,
isolat/strain.
PS C:\Users\LENOVO\OneDrive\Documents\test> python Main.py
Masukan gambar tanaman: 8.jpg
1/1 [==============================] - 0s 244ms/step
Tanamanmu di Identifikasi: normal (sehat)
PS C:\Users\LENOVO\OneDrive\Documents\test> python Main.py
Masukan gambar tanaman: 0.jpg
1/1 [==============================] - 0s 177ms/step
Tanamanmu di Identifikasi: kekurangan unsur agrept, bactocyn, plant
n fungisida berbahan aktif tembaga hidroksida misalnya nordox atau
PS C:\Users\LENOVO\OneDrive\Documents\test>
```

Figure 15. Results of Rice Leaf Image Identification

The model provides results in the form of leaf disease identification that matches the image provided. The results are in the form of malnutrition labels that indicate the type of disease predicted by the model.

Table 2. Epoch Training Results

Epoch	Loss	Accuracy	Validation loss	Validation Accuracy
30	01%	99%	00%	100%
35	00%	99%	37%	100%
40	01%	99%	44%	100%
45	03%	99%	00%	100%
50	01%	99%	00%	99%

The results above show that training epochs have varying accuracy values and the accuracy results of epoch 50 are better than other epochs. Testing using confusion matrix, Confusion matrix on the identification of image leaf malnutrition is used to evaluate the performance of the model in classifying malnutrition conditions in leaves based on the given image. The confusion matrix contains four main evaluation metrics, namely True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). In the context of identifying image leaf malnutrition, the confusion matrix can be displayed as a 2x2 matrix with four parts as follows:

Table 3. Brief Explanation of Identification Result Representation

Malnutrition	Description
True Positive (TP)	True Positive (TP) shows the number of leaves that are correctly classified as malnourished.
False Positive (FP)	False Positive (FP) shows the number of leaves that are incorrectly classified as malnourished, when in fact they are healthy.
False Negative (FN)	False Negative (FN) shows the number of leaves that are incorrectly classified as healthy, when in fact they are malnourished.
True Negative (TN)	True Negative (TN) shows the number of leaves that are correctly classified as healthy.

To get a clearer picture of the performance of the trained model, a heatmap visualization was created using the Seaborn and Matplotlib libraries. The visualization illustrates the effectiveness of the model in classifying test data by comparing actual labels and predicted labels. The heatmap visualization shows the data in different color representations. The color used to fill each cell reflects the number of samples in each combination of actual and predicted values. The more samples, the darker or stronger the color used. The colors used in heatmaps can provide visual information about the distribution of errors and correct predictions in the model. From the visualization results, it can be seen that many identification errors occur, due to too much data from the trained model.

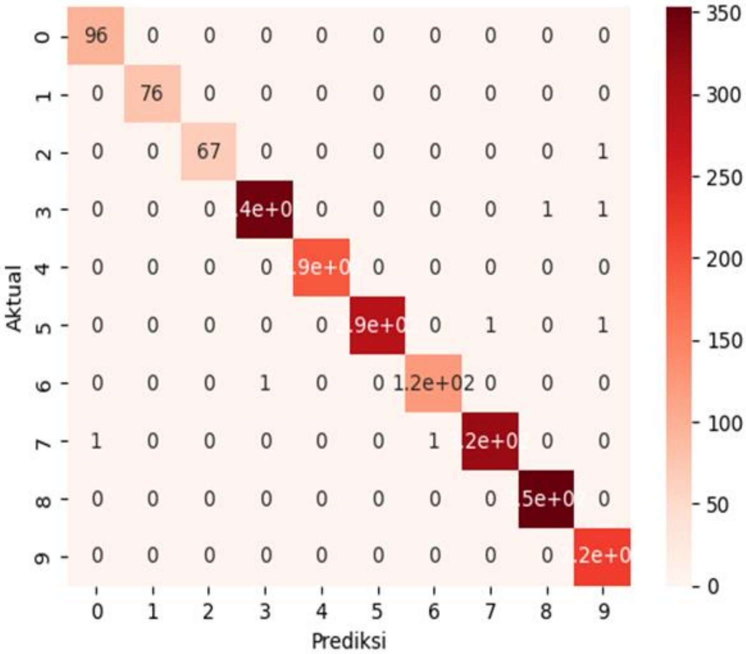


Figure 16. Heatmap visualization of the Trained Model

The confusion matrix can be used to calculate various evaluation metrics such as accuracy, precision, recall, and F1-score. These metrics provide information about the effectiveness of the model in identifying data in the correct class. These metrics are calculated by aggregating the results by category using a certain average, micro-average, or macro-average. Micro-average calculates the score by summing the total TP, FP, and FN regardless of class. Macro-average calculates the evaluation metrics separately for each class and then averages them.

To view the identification performance data using these metrics, you can use the classification report() function in the scikit-learn library. This function returns precision, recall, and F1-score data for each class along with the number of samples (support). Aggregate metrics such as accuracy, macro-average, and weighted average of class-specific metrics are also available. The output of this function is shown in Table 4. This is class-specific information and identification performance information.

Table 4. Information About the Identification Performance of the Trained Model

	precision	recall	f1-score	support
Class 1	0.99	1.00	0.99	96
Class 2	1.00	1.00	1.00	76
Class 3	1.00	0.99	0.99	68
Class 4	1.00	0.99	1.00	347
Class 5	1.00	1.00	1.00	193
Class 6	1.00	0.99	1.00	289
Class 7	0.99	0.99	0.99	124
Class 8	1.00	0.99	1.00	319
Class 9	1.00	1.00	1.00	353
Class 10	0.99	1.00	0.99	218
Accuracy			1.00	2083
Macro avg	1.00	1.00	1.00	2083
Weighted avg	1.00	1.00	1.00	2083

Conclusion

The conclusion of this study shows that the Convolutional Neural Network (CNN) model is effective for image identification, this algorithm can recognize patterns and features in images with an accuracy rate of 98.89% and higher performance in specific tests, namely 99.6% for accuracy, and precision, recall, and f1-score values of 99.5% each. This study also revealed that the CNN model has good reliability and generalization capabilities against different variations of image data, indicating the potential for broad applications in recognizing objects or features in images that have never been seen before.

This research has the potential to provide great benefits for farmers, especially in early detection of plant problems such as diseases or pests using digital images. CNN technology can help farmers reduce pesticide use, save costs, and increase agricultural yields. On a large scale, this technology can be applied to drone-based or IoT monitoring systems to monitor plant health in real time and support food security.

Acknowledge

The authors would like to thank all those who have supported the completion of this research process, and who have contributed both time and thought. The author would also like to thank Kaggle for publishing the dataset in the form of digital images of rice leaves.

References

[1] M. H. Aabidi, A. EL Makrani, B. Jabir, and I. Zaimi, “A Model Proposal for Enhancing Leaf Disease Detection Using Convolutional Neural Networks (CNN): Case Study.,” *Int. J. Online Biomed. Eng.*, vol. 19, no. 12, 2023.

[2] R. Muslim, Z. Zaeniah, A. Akbar, B. Imran, and Z. Zaenudin, “Disease Detection of Rice and Chili Based on Image Classification Using Convolutional Neural Network Android-Based,” *J. Pilar Nusa Mandiri*, vol. 19, no. 2, pp. 85–96, 2023.

[3] T. Rashid, M. M. Mokji, and M. Rasheed, “Cracked concrete surface classification in low-resolution images using a convolutional neural network,” *J. Opt.*, pp. 1–13, 2024.

[4] T. J. Nandhini and K. Thinakaran, “Object Detection Algorithm Based on Multi-Scaled Convolutional Neural Networks,” in *2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP)*, IEEE, 2023, pp. 1–5.

[5] A. Shah *et al.*, “A comprehensive study on skin cancer detection using artificial neural network (ANN) and convolutional neural network (CNN),” *Clin. eHealth*, 2023.

[6] Q. Li, G. Chen, and L. Luo, “Mineral prospectivity mapping using attention-based convolutional neural network,” *Ore Geol. Rev.*, vol. 156, p. 105381, 2023.

[7] A. E. Ilesanmi, T. Ilesanmi, and G. A. Gbotoso, “A systematic review of retinal fundus image segmentation and classification methods using convolutional neural networks,” *Healthc. Anal.*, vol. 4, p. 100261, 2023.

[8] T. W. Qurâ, “Implementasi Metode Convolutional Neural Network (CNN) untuk Klasifikasi Motif pada Citra Sasirangan,” *Madani J. Ilm. Multidisiplin*, vol. 1, no. 7, 2023.

[9] M. H. Kurniawan, “APLIKASI IDENTIFIKASI KENDARAAN KARYAWAN DENGAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN),” *Sci. J. Ilm. Sains dan Teknol.*, vol. 2, no. 2, pp. 251–255, 2024.

[10] F. A. Febriyanti, “Image Processing Dengan Metode Convolutional Neural Network (Cnn) Untuk Deteksi Penyakit Kulit Pada Manusia,” *Kohesi J. Sains dan Teknol.*, vol. 3, no. 10, pp. 21–30, 2024.

[11] R. Arifuddin and B. F. Hidayatulail, “Klasifikasi Penyakit Pneumonia menggunakan Metode Convolutional Neural Network (CNN),” *Techné J. Ilm. Elektrotek.*, vol. 23, no. 2, pp. 233–244, 2024.

[12] R. Aryanto, M. A. Rosid, and S. Busono, “Penerapan Deep Learning untuk Pengenalan Tulisan Tangan Bahasa Aksara Lota Ende dengan Menggunakan Metode Convolutional Neural Networks,” *J. Inf. dan Teknol.*, pp. 258–264, 2023.

[13] D. I. Permatasari, “IMPLEMENTASI METODE CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK KLASIFIKASI TANAMAN HERBAL BERDASARKAN CITRA DAUN,” *Kohesi J. Sains dan Teknol.*, vol. 3, no. 9, pp. 1–10, 2024.

[14] S. Mehta, V. Kukreja, A. Bansal, K. Kaur, and A. Singh, “Exploring the Potential of Convolutional Neural Networks in Automatic Diagnosis of Dragon Fruit Diseases from Plant Photographs,” in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2023, pp. 536–540.

[15] M. Reyad, A. M. Sarhan, and M. Arafa, “A modified Adam algorithm for deep neural network optimization,” *Neural Comput. Appl.*, vol. 35, no. 23, pp. 17095–17112, 2023.