Mantik Journal, 8 (4) (2025), ISSN 2685-4236 (Online)

IOCSCIENCE

Published by: Institute of Computer Science (IOCS)

Jurnal Mantik

Journal homepage: www.iocscience.org/ejournal/index.php/mantik

Implementation Of Convolutional Neural Network For Diagnosing Rice Plant Diseases Using Colab Python Integrated With Streamlit

Ahmad Habib[1], Haydar Yahya[2]

[1,2]Informatics, Faculty of Engineering, Computer Science, 17 August 1945 University Surabaya

ARTICLE INFO

Article history:

Received Jan 07, 2025
Revised Jan 10, 2025
Accepted Feb 12, 2025

Keywords:

Agriculture;
Convolutional Neural Network;
Google Colab;
Python;
Streamlit.

ABSTRACT

Agriculture, particularly rice cultivation, is crucial for Indonesia's food security; however, production is often hindered by pests and diseases. With over 30 million hectares of rice fields and millions of farmers relying on this staple crop, the impact of these challenges is significant, threatening both livelihoods and national food supply. This study aims to develop a rice plant disease diagnosis system using Convolutional Neural Network (CNN) methods implemented in a Streamlit-based application. Data were obtained from an open dataset on Kaggle, which includes images of healthy and infected rice leaves. The Streamlit application facilitates users in uploading images and receiving real-time diagnoses. Results show that the CNN model achieved an accuracy of 96.03% in identifying diseases, demonstrating a strong ability to recognize patterns in leaf images. This system offers an efficient solution to help farmers quickly and accurately detect rice diseases, contributing to increased agricultural productivity and food security in Indonesia.

Corresponding Author:

Haydar Yahya,
Informatics/Faculty of Engineering/Computer Science,
17 August 1945 Univerity Surabaya,
Jln Semolowaru No 45, Surabaya, East Java, 60117, Indonesia.
Email: haydarmadridista7@gmail.com

1. INTRODUCTION

Agriculture plays a crucial role in ensuring food security, particularly in countries like Indonesia, where rice is a staple food for the majority of the population (Hutabarat, F. P., & Nasution, 2024). However, rice production is frequently threatened by various pests and diseases that can significantly reduce crop yields (Putra, J. V. P., Ayu, F., & Julianto, 2023). According to the Indonesian Ministry of Agriculture, it is estimated that pests and diseases can cause yield losses of up to 30% annually, translating to millions of tons of rice that could have been harvested (Kementerian Pertanian, 2022). One of the primary challenges faced by farmers is the difficulty in diagnosing plant diseases quickly and accurately. Traditional methods of disease diagnosis often rely on visual inspections by experts, which can be time-consuming and require specialized knowledge (Pratama et al., 2024). This delay in diagnosis can lead to severe consequences, including reduced crop quality and quantity, ultimately impacting the livelihoods of farmers and the overall food supply (Mohanty, S. P., Hughes, D. P., & Salathé, 2019).

In recent years, advancements in technology, particularly in the field of artificial intelligence (AI) and machine learning, have opened new avenues for improving agricultural practices (Khoiruddin et al., 2022). Among these technologies, Convolutional Neural Networks (CNNs) have emerged as a powerful tool for image analysis and classification tasks. CNNs are particularly effective in recognizing patterns and features in images, making them suitable for diagnosing plant diseases based on leaf images (Prisheila Dharmawan et al., 2023). By leveraging CNNs, farmers can obtain rapid and accurate diagnoses, enabling them to take timely action against plant diseases. For instance, a study conducted in West Java demonstrated that implementing AI-based diagnostic tools reduced the time taken for disease identification from several days to mere minutes, significantly improving farmers' response times (Sari et al., 2023).

To further enhance the accessibility and usability of this technology, we propose the development of a web-based application using Streamlit, a popular framework for building interactive web applications in Python (Rizkyansyah & Susilawati, 2024). Streamlit allows for the creation of user-friendly interfaces that can facilitate the interaction between farmers and the diagnostic system. By integrating CNNs with Streamlit, we aim to provide a platform where farmers can easily upload images of their rice plants and receive instant feedback on potential diseases, along with recommended treatments (Aryanto et al., 2023).

This study aims to design and implement a CNN-based system for diagnosing rice plant diseases, utilizing the Streamlit framework to create an intuitive user interface. The expected outcome is a robust diagnostic tool that not only improves the accuracy and speed of disease detection but also empowers farmers with the knowledge and resources needed to manage their crops effectively. Through this research, we hope to contribute to the advancement of precision agriculture and support the sustainability of rice production in Indonesia (Sulistiyana & Anardani, 2023). By addressing the challenges posed by plant diseases with innovative technology, we can help mitigate the estimated annual losses and enhance food security for millions of Indonesians.

2. RESEARCH METHOD

This section outlines the research methods employed in the development of a Convolutional Neural Network (CNN) based system for diagnosing rice plant diseases, utilizing the Streamlit framework for user interaction. The methodology is divided into several key phases: data collection, system design, model development, and evaluation.(UNGKAWA & HAKIM, 2023)

2.1. Data Collection

Dataset Acquisition and Data Prepocessing.

Table 1. Dataset Structure

Dataset Structure	Description
Source	Kaggle(Open-Source dataset for rice plant)
Total Number of Images	6,000 Images
Images Dimesion	150 x 150 pixels
Classes of Disease	Bacterial Blight Brown Spot Blast Tungro
Data Augmentation Techniques	Rescale 1./255 Rotation_range 30 Width_shift_range 0.2 Height_shift_range 0.2 Shear_range 0.2 Zoom_range 0.2

	Validaton_split 0.2
Prepocessing Steps	Resizing images to 150 x 150 Normalization of pixel values (0-1 range)
Usage	Training and Validation of the CNN Model

The dataset used for training the CNN model was sourced from Kaggle, which provides a variety of open-source datasets related to rice plant diseases. The dataset includes images of rice leaves affected by various diseases such as Bacterial Leaf Blight, Brown Spot, and Blast (Pratama et al., 2024).

Images were resized to a uniform dimension (150x150 pixels) to ensure consistency in input size for the CNN. Data augmentation techniques were applied to enhance the diversity of the training dataset. This included random rotations, shifts, flips, and zooms to create variations of the original images, thereby improving the model's robustness and reducing the risk of overfitting (Khoiruddin, M., Junaidi, A., & Saputra, 2022).

2.2 System Design

a. Framework Selection

Streamlit was chosen as the framework for developing the web application due to its simplicity and effectiveness in creating interactive user interfaces. Streamlit allows for rapid development of web applications with minimal coding, making it ideal for this project (Lesmana et al., 2022).

b. User Interface Design

The user interface was designed to be intuitive, enabling users to upload images of rice leaves and view the results of the diagnosis along with recommended actions. The interface includes. An image upload button for users to submit their leaf images. A display area for showing the prediction results, including the predicted disease and the model's confidence level. (Wisak et al., 2024).

Table 2. User Interface Design

Component	Description
Logo	Image styled with rice plants.
Title	“Rice Plant Disease Diagnosis”.
Description	This application uses a machine learning model to diagnose plant diseases in rice based on uploaded images.
Image Upload Area	A large area for uploading images.
File Sizes Limit	Maximum allowed file size (e.g., 200MB).
Supported File Formats	Supported images (e.g., JPG, PNG, JPEG).
Diagnosis	Area to display the diagnosis and explanation of identified diseases
Footer	Additional information or contact details

2.3 Model Development

a. Convolutional Neural Network Architecture.

The CNN model was built using the VGG16 architecture, which is known for its depth and effectiveness in image classification tasks. The model was initialized with pre-trained weights from the ImageNet dataset to leverage transfer learning (Virtanen, T. E., Räikkönen, E., Lerkkanen, M.-K., Määttä, S., & Vasalampi, n.d.). The architecture included: Convolutional layers for feature extraction, flattening layer to convert the 2D feature maps into a 1D vector, Dense layers for classification, with a final output layer using softmax activation to predict the probability of each disease class.(Hawari, F.H., 2022)

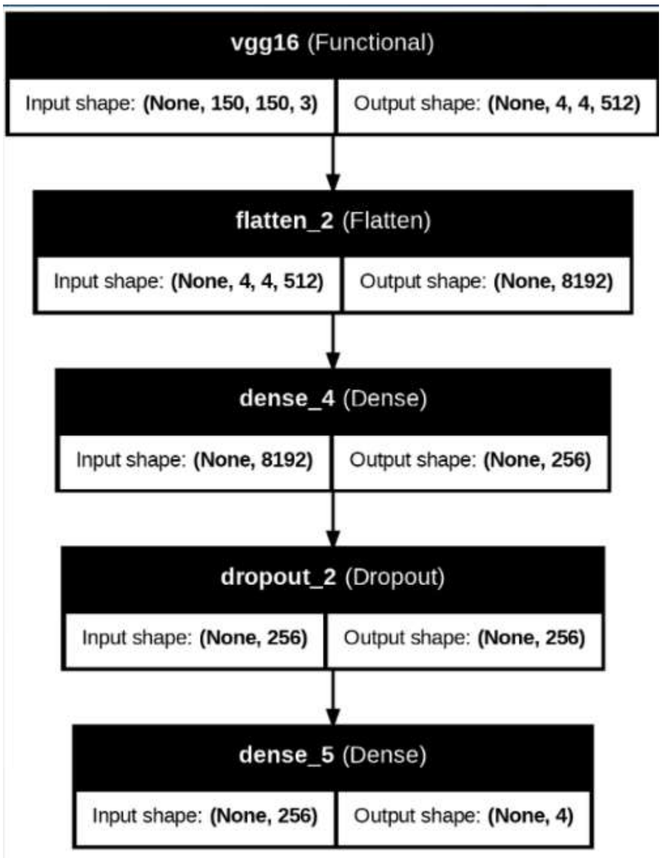

Figure 2. Convolutional Neural Network Architecture

2.4 Training The Model

The model was trained using the training dataset, with a validation split to monitor performance during training. The training process involved optimizing the model using the Adam optimizer and minimizing the categorical cross-entropy loss function.(Zuraida, V., Kusbianto, D., & Pahlevi, 2023). Early stopping was implemented to prevent overfitting by halting training when the validation loss did not improve for a specified number of epochs (C. M. Putra et al., 2023).

```
Epoch 1/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 53s 318ms/step - accuracy: 0.6063 - loss: 1.0923 - val_accuracy:
0.8418 - val_loss: 0.3999
Epoch 2/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 305ms/step - accuracy: 0.8199 - loss: 0.4620 - val_accuracy:
0.8856 - val_loss: 0.3112
Epoch 3/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 303ms/step - accuracy: 0.8635 - loss: 0.3579 - val_accuracy:
0.9002 - val_loss: 0.2656
Epoch 4/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 50s 309ms/step - accuracy: 0.8855 - loss: 0.3108 - val_accuracy:
```

```
0.8946 - val_loss: 0.2617
Epoch 5/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 303ms/step - accuracy: 0.8849 - loss: 0.2966 - val_accuracy: 0.9205 - val_loss: 0.2200
Epoch 6/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 48s 300ms/step - accuracy: 0.8808 - loss: 0.3017 - val_accuracy: 0.9294 - val_loss: 0.1833
Epoch 7/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 304ms/step - accuracy: 0.8940 - loss: 0.2709 - val_accuracy: 0.9343 - val_loss: 0.1811
Epoch 8/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 301ms/step - accuracy: 0.9046 - loss: 0.2461 - val_accuracy: 0.9140 - val_loss: 0.2156
Epoch 9/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 302ms/step - accuracy: 0.8996 - loss: 0.2549 - val_accuracy: 0.9416 - val_loss: 0.1622
Epoch 10/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 302ms/step - accuracy: 0.9167 - loss: 0.2157 - val_accuracy: 0.9530 - val_loss: 0.1440
Epoch 11/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 48s 301ms/step - accuracy: 0.9194 - loss: 0.2127 - val_accuracy: 0.9513 - val_loss: 0.1391
Epoch 12/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 302ms/step - accuracy: 0.9097 - loss: 0.2292 - val_accuracy: 0.9505 - val_loss: 0.1394
Epoch 13/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 301ms/step - accuracy: 0.9167 - loss: 0.2116 - val_accuracy: 0.9562 - val_loss: 0.1177
Epoch 14/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 303ms/step - accuracy: 0.9207 - loss: 0.1900 - val_accuracy: 0.9497 - val_loss: 0.1308
Epoch 15/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 48s 300ms/step - accuracy: 0.9317 - loss: 0.1760 - val_accuracy: 0.9530 - val_loss: 0.1321
Epoch 16/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 302ms/step - accuracy: 0.9244 - loss: 0.1945 - val_accuracy: 0.9254 - val_loss: 0.1853
Epoch 17/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 304ms/step - accuracy: 0.9161 - loss: 0.2168 - val_accuracy: 0.9635 - val_loss: 0.1113
Epoch 18/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 48s 300ms/step - accuracy: 0.9349 - loss: 0.1770 - val_accuracy: 0.9546 - val_loss: 0.1255
Epoch 19/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 304ms/step - accuracy: 0.9316 - loss: 0.1811 - val_accuracy: 0.9546 - val_loss: 0.1418
```

```
Epoch 20/20
155/155 ━━━━━━━━━━━━━━━━━━━━ 49s 301ms/step - accuracy: 0.9250 - loss: 0.2037 - val_accuracy: 0.9367 - val_loss: 0.1520
```

Figure 3. Training Model

2.5 Evaluation

Confussion Matrix, The image shows a confusion matrix generated by a model trained on classifying rice diseases. The model predicts four different types of rice diseases: Bacterialblight, Blast, Brownspot, and Tungro. The confusion matrix is a table that shows how many instances of each class were correctly classified and how many were incorrectly pclassified(Fajhar Muhammad et al., 2024).

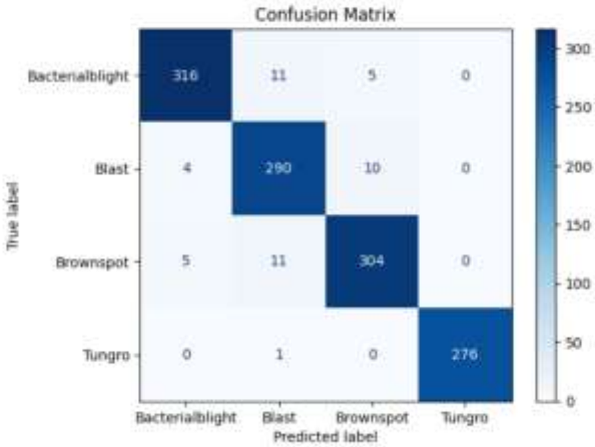

Figure 4. Confussion Matrix

a. Classification Report

Table 2. Classification Report

Class	Precision	Recall	F1-Score	Support
Bacterial Blight	0.97	0.95	0.96	332
Blast	0.93	0.95	0.94	304
Brown Spot	0.95	0.95	0.95	320
Tungro	1.00	1.00	1.00	277

Accuracy		0.96	1233	
Macro Avg	0.96	0.96	0.96	1233
Weighted Avg	0.96	0.96	0.96	1233

The classification report summarizes the performance of the CNN model in diagnosing rice plant diseases, showing high effectiveness across all classes. Bacterial Blight achieved a precision of 0.97, indicating that 97% of its predicted cases were correct, while Tungro had a perfect recall of 1.00, meaning all instances were accurately identified. The F1-Score for Tungro was also 1.00, reflecting flawless performance. Overall, the model attained an accuracy of 0.96, with both macro and weighted averages at 0.96, demonstrating consistent and reliable performance. This high level of accuracy and reliability is crucial for providing timely and accurate diagnoses to farmers, ultimately supporting better crop management and food security (Imantyar, R., Fudholi, 2021).

2.6 Implementation

Streamlit application designed for diagnosing rice plant diseases using a pre-trained Convolutional Neural Network (CNN) model. The application begins by loading the model and setting up the user interface with a title. Users can upload images of rice leaves in formats such as JPG, PNG, or JPEG. Once an image is uploaded, the application processes the image by resizing it to the target dimensions of 150x150 pixels, normalizing the pixel values, and preparing it for prediction by adding a batch dimension.(Maulana et al., 2023).

After processing the image, the model makes predictions based on the input. The application identifies the predicted class from a predefined list of disease labels, which includes Bacterial Blight, Blast, Brown Spot, and Tungro. The predicted label is then displayed to the user, along with the uploaded image for reference. This interactive tool provides farmers with a quick and efficient way to diagnose potential diseases in their rice crops, ultimately aiding in better crop management and health monitoring.(Maulana, F.F., & Rochmawati, 2019)

```
import streamlit as st
from keras.models import load_model
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt

st.title('Diagnosa Penyakit Padi')

# load model
model =
load_model('/content/drive/MyDrive/TugasAkhir/1461900123_model.keras')

def load_and_preprocess_image(image_path, target_size):
    img = Image.open(image_path)
    img = img.resize(target_size)
    img_array = np.array(img)

    if img_array.shape[-1] == 4:  # Jika gambar memiliki saluran alpha
        img_array = img_array[..., :3]  # Menghapus saluran alpha

    img_array = img_array / 255.0  # Normalisasi
    img_array = np.expand_dims(img_array, axis=0)  # Menambahkan dimensi
batch

    return img_array

uploaded_file = st.file_uploader("Upload Picture", type=["jpg", "png",
"jpeg"])

if uploaded_file is not None:
    # load and proces picture
    target_size = (150, 150)
    processed_image = load_and_preprocess_image(uploaded_file,
target_size)

    # make prediction
    predictions = model.predict(processed_image)


    class_labels = ['BacterialBlight', 'Blast', 'BrownSpot', 'Tungro']
    predicted_class = np.argmax(predictions, axis=1)
    predicted_label = class_labels[predicted_class[0]]
    st.write("Predicted Label:", predicted_label)

      image = Image.open(uploaded_file)
    st.image(image, caption='Gambar yang Diunggah', use_column_width=True)
```

Figure 5. Streamlit Code Program.

Figure 5. Streamlit app results

3. RESULTS AND DISCUSSIONS

The CNN model was successfully implemented to diagnose rice plant diseases using a dataset sourced from Kaggle, which included various images of infected rice leaves. The architecture utilized was VGG16, a well-known deep learning model that has shown high accuracy in image classification tasks. The model was trained over 20 epochs, achieving a training accuracy of 93.17% and a validation accuracy of 96.03%. The loss decreased consistently throughout the training process, indicating effective learning and generalization capabilities of the model.

The classification report generated from the model's predictions showed high precision, recall, and F1 scores across all classes of rice diseases. Specifically, the model achieved precision of 0.97, recall of 0.95, and F1 score of 0.96 for Bacterial Blight; precision of 0.93, recall of 0.95, and F1 score of 0.94 for Blast; precision of 0.95, recall of 0.95, and F1 score of 0.95 for Brown Spot; and a perfect score of 1.00 for both precision and recall for Tungro. These results indicpate that the model is highly effective in identifying and classifying the diseases present in the rice plants (Jinan et al., 2022).

The confusion matrix further illustrated the model's performance, showing that it correctly classified a significant number of images for each disease category. For instance, it accurately identified 316 images of Bacterial Blight and 290 images of Blast, demonstrating the model's robustness in distinguishing between similar symptoms (Malik Ibrahim et al., 2023). Visualizations of the model's predictions confirmed its ability to accurately classify images of rice plant diseases. The model's predictions were compared against true labels, with a high degree of accuracy observed in the results. This visual confirmation is crucial for building trust in the model's application in real-world scenarios (J. V. P. Putra et al., 2023).

The successful implementation of this CNN-based diagnostic system has significant implications for rice farmers. By providing a tool that can quickly and accurately diagnose plant diseases, farmers can take timely action to mitigate crop losses, thereby enhancing food security and agricultural productivity in Indonesia (Hutabarat, F. P., & Nasution, 2024). Despite the promising results, the study acknowledges limitations such as the dataset's diversity and the need for further augmentation techniques to improve model robustness. Future work should focus on expanding the dataset and exploring transfer learning techniques to enhance the model's performance across a broader range of conditions (Pratama et al., 2024).

4. CONCLUSION

The implementation of the Convolutional Neural Network (CNN) model using the VGG16 architecture has effectively diagnosed rice plant diseases, achieving training and validation accuracies of 93.17% and 96.03%, respectively. The model's performance is validated by metrics such as precision, recall, and F1 scores, particularly for diseases like Bacterial Blight, Blast, Brown Spot, and Tungro. Confusion matrix analysis demonstrated accurate classification across all disease categories, showcasing the model's robustness. This CNN-based diagnostic system offers significant benefits for rice farmers, enabling quick and accurate disease diagnosis, which can lead to timely interventions and improved food security in Indonesia.

However, the study acknowledges limitations in dataset diversity, which may affect the model's generalizability. Future research should focus on expanding the dataset and exploring transfer learning strategies to enhance performance (Khairunnas Khairunnas et al., 2022). The potential for successful real-world implementation is promising, as this system can empower farmers to make informed decisions, leading to better crop management and increased yields. Integrating this technology with mobile applications could further enhance accessibility for farmers in remote areas (Khoiruddin, M., Junaidi, A., & Saputra, 2022)

REFERENCES

Aryanto, R., Alfan Rosid, M., & Busono, S. (2023). Penerapan Deep Learning untuk Pengenalan Tulisan Tangan Bahasa Aksara Lota Ende dengan Menggunakan Metode Convolutional Neural Networks. *Jurnal Informasi Dan Teknologi*, *5*(1), 258–264. https://doi.org/10.37034/jidt.v5i1.313

Fajhar Muhammad, Agung Triayudi, & Eri Mardiani. (2024). Implementation of Face Recognition for Lecturer Attendance Using Deep Learning CNN Algorithm. *SaNa: Journal of Blockchain, NFTs and Metaverse Technology*, *2*(2), 123–130. https://doi.org/10.58905/sana.v2i2.275

Hawari, F.H., D. (2022). No Title. *Klasifikasi Penyakit Padi Menggunakan Algoritma CNN (Convolutional Neural Network).*

Hutabarat, F. P., & Nasution, Y. R. (2024). (2024). *Sistem Pakar Diagnosis Hama dan Penyakit pada Tanaman Padi menggunakan Metode Certainty Factor. MEANS (Media Informasi Analisa Dan Sistem).*

Imantyar, R., Fudholi, D. . (2021). No Title. *Kajian Pengaruh Dataset Dan Bias Dataset Terhadap Performa Akurasi Deteksi Objek.*

Jinan, A., Hayadi, B. H., & Utama, U. P. (2022). Klasifikasi Penyakit Tanaman Padi Mengunakan Metode Convolutional Neural Network Melalui Citra Daun (Multilayer Perceptron). *Journal of Computer and Engineering Science*, *1*(2), 37–44.

Khairunnas Khairunnas, Husna Gemasih, & Hendri Syahputra. (2022). Rancang Bangun Sistem Pakar Penyakit Tanaman Cabai Menggunakan Metode Naïve Bayes Berbasis Web. *Ocean Engineering : Jurnal Ilmu Teknik Dan Teknologi Maritim*, *1*(3), 59–73.

https://doi.org/10.58192/ocean.v1i3.374
Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). *). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. Journal of Dinda : Data Science, Information Technology, and Data Analytics,.*
Khoiruddin, M., Junaidi, A., & Saputra, W. A. (2022). Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network. *Journal of Dinda : Data Science, Information Technology, and Data Analytics*, *2*(1), 37–45. https://doi.org/10.20895/dinda.v2i1.341
Lesmana, A. M., Fadhillah, R. P., & Rozikin, C. (2022). Identifikasi Penyakit pada Citra Daun Kentang Menggunakan Convolutional Neural Network (CNN). *Jurnal Sains Dan Informatika*, *8*(1), 21–30. https://doi.org/10.34128/jsi.v8i1.377
Malik Ibrahim, M., Rahmadewi, R., & Nurpulaela, L. (2023). Pendeteksian Nominal Uang Pada Gambar Menggunakan Convolutional Neural Network: Integrasi Metode Pra-Pemrosesan Citra Dan Klasifikasi Berbasis Cnn. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(2), 1394–1400. https://doi.org/10.36040/jati.v7i2.6863
Maulana, F.F., & Rochmawati, N. (2019). No Title. *Klasifikasi Citra Buah Menggunakan Convolutional Neural Network*.
Maulana, R., Prayoga, J., & Yasir, A. (2023). Sistem Pakar Diagnosa Penyakit Pada Tanaman Jeruk Menggunakan Metode Forward Chaining. *Warta Dharmawangsa*, *17*(4), 1546–1563. https://doi.org/10.46576/wdw.v17i4.3806
Mohanty, S. P., Hughes, D. P., & Salathé, M. (2019). (2019). *Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419*.
Pratama, M. D., Gustriansyah, R., & Purnamasari, E. (2024). Klasifikasi Penyakit Daun Pisang menggunakan Convolutional Neural Network (CNN). *Jurnal Teknologi Terpadu*, *10*(1), 1–6. https://doi.org/10.54914/jtt.v10i1.1167
Prisheila Dharmawan, D., Dwi Anggraeni, S., Fahmi Al-Hafidz, A., & Puspita Sari, A. (2023). Perbandingan Penerapan Logika Fuzzy Metode Mamdani Dan Metode Tsukamoto Pada Sistem Diagnosa Penyakit Tanaman Padi. *Seminar Nasional Informatika Bela Negara (SANTIKA)*, *3*, 28–36.
Putra, J. V. P., Ayu, F., & Julianto, B. (2023). *Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN. Stains (Seminar Nasional Teknologi & Sains)*.
Putra, C. M., Triayudi, A., & Ningsih, S. (2023). Face Mask Recognition Menggunakan Model CNN (Convolutional Neural Network) Berbasis Python dan OpenCV. *Journal of Computer System and Informatics (JoSYC)*, *4*(3), 722–730. https://doi.org/10.47065/josyc.v4i3.3532
Putra, J. V. P., Ayu, F., & Julianto, B. (2023). Implementasi Pendeteksi Penyakit pada Daun Alpukat Menggunakan Metode CNN. *Stains (Seminar Nasional Teknologi & Sains)*, *2*(1), 155–162.
Rizkyansyah, M. D., & Susilawati, I. (2024). *Sistem Pakar Diagnosa Penyakit Pada Tanaman Padi (Oryza) Menggunakan Metode Naïve Bayes Expert System For Diagnosing Diseases In Rice Plants (Oryza) Using The Naïve Bayes Method*. *14*(2), 168–181.
Sulistiyana, F., & Anardani, S. (2023). Aplikasi Deteksi Penyakit Tanaman Jagung Dengan Convolutional Neural Network dan Support Vector Machine. *Seminar Nasional Teknologi Informasi Dan Komunikasi (SENATIK)*, *6*(1), 423–432.
UNGKAWA, U., & HAKIM, G. AL. (2023). Klasifikasi Warna pada Kematangan Buah Kopi Kuning menggunakan Metode CNN Inception V3. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *11*(3), 731. https://doi.org/10.26760/elkomika.v11i3.731
Virtanen, T. E., Räikkönen, E., Lerkkanen, M.-K., Määttä, S., & Vasalampi, K. (n.d.). Development of Participation in and Identification With School: Associations With Truancy. *2021*.
Wisak, S. A., Safirah, N. A., & Kaesmetan, Y. R. (2024). Identifikasi Jenis Mangga Berdasarkan Ciri Daun Menggunakan Metode Cnn. *Jurnal Sistem Informasi Dan Informatika (Simika)*, *7*(2), 121–129. https://doi.org/10.47080/simika.v7i2.3295
Zuraida, V., Kusbianto, D., & Pahlevi, M. R. (2023). *Sistem Pakar Diagnosis Penyakit dan Hama pada Tanaman Padi dengan Metode Forward Chaining. Jurnal Minfo Polgan, 12(1), 378–384*.