Journal of ICT Aplication and System (JICTAS)
Volume 3 Number 2, December 2024, pp. 66 ~74
ISSN: 2830-098X, DOI: 10.56313/jictas.v1i1.3i2

Classification of Capsicum Varieties Using Color Analysis with Convolutional Neural Network

Tantia Azzahra[1*], Riski Rahmadan[2], Fernanda Abi Maulana[3], Ismi Asmita[4], Efendi Rahayu[5], Fauzi Erwis[6]

[12345]Informatics Engineering, University Pasir Pengaraian, Riau Indonesia
[6] Computer Sciencs, Universitas Rokania, Riau, Indonesia

Article Info

Article history:

Received 15 November 2024
Revised 10 Desember 2024
Accepted 28 Desember 2024

Keywords :

Capsicum Varieties, Color Analysis, Convolutional Neural Network (CNN)

ABSTRACT

Paprika (**Capsicum annuum L.**) is a high-value horticultural commodity widely consumed for its nutritional content and vibrant color variations. In the agricultural industry, classifying paprika varieties based on color is crucial for ensuring product quality and optimizing sorting processes. This study developed an automated classification system for three main paprika varieties—red, green, and yellow—using the Convolutional Neural Network (CNN) method. The dataset consisted of 1,820 images sourced from Kaggle, with data split into 60% for training and 40% for validation. Preprocessing steps included resizing images, normalizing pixel values to the range [0,1], and data augmentation techniques such as rotation, flipping, and brightness adjustments to enhance dataset diversity and reduce the risk of overfitting. The CNN model was designed with key layers, including convolutional, pooling, and fully connected layers, optimized using the Adam algorithm and categorical cross-entropy loss function. The training results showed an accuracy of 99.9% on the training data and 92% on the testing data, with an average processing time of 64 seconds per image and a maximum of 78 seconds, demonstrating the model's efficiency for real-time applications. The k-fold cross-validation technique was also employed to ensure the model's generalization ability to new data. This study demonstrated that CNN is an effective method for classifying paprika varieties based on color analysis, offering an accurate, fast, and scalable solution for automating sorting and grading processes in the agricultural sector, reducing human errors, and improving operational efficiency.

Corresponding Author:

Tantia Azzahra
Informatics Engineering, University Pasir Pengaraian, Riau Indonesia
Email: azzahratantia14@gmail.com

1. INTRODUCTION

Paprika (Capsicum annuum L.) is one of the most economically valuable horticultural commodities, widely consumed for its nutritional content and vibrant color variations. The three main varieties of paprika—red, yellow, and green—are highly sought after in global markets. Their color differences arise from the pigments chlorophyll, carotenoids, and anthocyanins, which are also indicators of ripeness and quality [1]. Accurate classification of paprika varieties is essential in the agricultural industry to ensure product quality, enhance operational efficiency, and support automated processing systems [2]. Traditionally, this classification has been performed manually by human workers relying on visual inspection. However, such methods are time-consuming, prone to human error, and inconsistent due to subjective judgments [3].

In recent years, technological advancements in artificial intelligence (AI) and machine learning (ML) have provided solutions to overcome these challenges. Specifically, Convolutional Neural Networks (CNN), a method within deep learning, have demonstrated remarkable capabilities in image recognition tasks, including object detection, image segmentation, and classification. CNNs are designed to extract hierarchical features from input images through multiple layers of convolutional and pooling operations, enabling accurate pattern recognition even in complex visual datasets [4].

Numerous studies have shown the effectiveness of CNNs in agricultural applications, particularly in classifying and analyzing plant-based images. For instance, Nugraha et al. (2023) successfully employed CNNs to classify paprika leaves into healthy and diseased categories with 97% accuracy, highlighting its potential in addressing plant health issues [5]. Similarly, Nisa et al. (2023) achieved 99.56% accuracy in mango variety classification based on shape features using CNNs, further validating its robustness in handling diverse agricultural datasets [6]. These studies underscore CNN's versatility in recognizing visual patterns across a range of agricultural products.

Color analysis has been a critical factor in differentiating fruit and vegetable varieties. Unlike shape or texture, color features often pose unique challenges due to variations in lighting conditions, camera angles, and shadows. Traditional image processing techniques that rely on handcrafted features often struggle to generalize across such variations. In contrast, CNNs are adept at learning both low-level and high-level features from raw image data, enabling them to excel in color-based classification tasks [7].

This study focuses on developing a CNN-based model to classify the three main varieties of paprika—red, yellow, and green—based on color features. The dataset, sourced from Kaggle, comprises 1,820 images captured under varying lighting conditions and angles to ensure diversity. Preprocessing steps such as normalization, resizing, and data augmentation (e.g., rotation, flipping, and brightness adjustments) were applied to enhance the dataset's variability and reduce the risk of overfitting during model training [8].

Google Colab, a cloud-based platform, was employed for training the CNN model, leveraging its powerful computational resources to handle large-scale datasets efficiently. This approach eliminates the limitations posed by local hardware, ensuring seamless access to deep learning libraries like TensorFlow and Keras [9]. The model architecture consists of several layers, including convolutional layers for feature extraction, pooling layers for dimensionality reduction, and fully connected layers for classification. The Adam optimizer and categorical cross-entropy loss function were used to optimize the training process, ensuring high accuracy and minimized errors.

Beyond achieving high classification accuracy, this study also evaluates the processing time required for each image. Real-time applications in the agricultural industry, such as automated sorting and grading systems, demand not only precision but also efficiency. By ensuring that the model performs well in terms of both accuracy and speed, this research aims to provide a scalable and practical solution for automating paprika classification processes [10].

This research contributes significantly to the agricultural sector by offering an AI-powered classification system that can improve productivity and reduce reliance on manual labor. The ability to classify paprika varieties accurately and efficiently supports better inventory management, quality control, and customer satisfaction. Moreover, the study advances the field of machine learning by demonstrating the adaptability of CNNs to agricultural problems, particularly in color-based classification tasks. By addressing the limitations of manual classification and traditional image processing methods, this research paves the way for broader applications of AI in agriculture and food industries [11].

2. METHOD

This research uses the method *Convolutional Neural Network* (CNN) for classification of paprika varieties based on color. Research stages include data collection, *pre-processing* data, model development

Convolutional Neural Network (CNN), model training and testing, and model performance evaluation [3]. Research methods.

1. Data Collection

The dataset used in this research was obtained from the Kaggle platform (www.kaggle.com/datasets/). The dataset consists of images of three types of peppers: red, yellow, and green. Each type of pepper was photographed in a variety of lighting conditions and viewing angles to ensure adequate variety. The total number of images collected was 1820, with 500 images per type of pepper. (corrected again. Added detailed data set

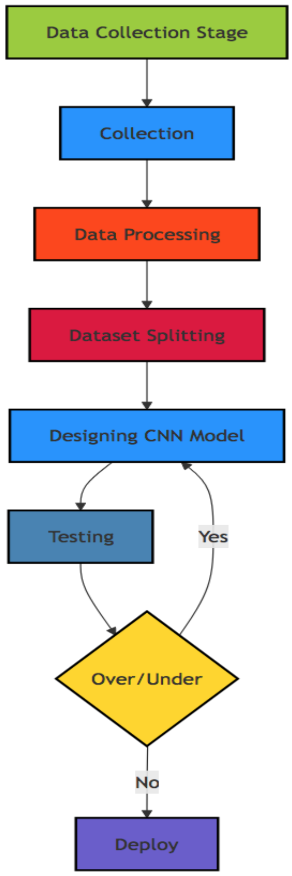


Figure 1. *Flowchart* Level of Data Collection

Level *pre-processing* carried out to prepare data before model training, including:

- Data division: the dataset is divided into two subsets, namely the training subset (*training set*) and validation subset (*validation set*) with a ratio of 150x150.
- *Normalization*: Normalizes image pixel values to the range [0, 1] to facilitate the training process.
- Data Augmentation: Performing augmentations such as rotation, *flipping*, and brightness changes to improve data variations and prevent *overfitting*.

.

Figure 2. Sample dataset of paprika classes

2. Model Development *Convolutional Neural Network* (CNN)

The CNN model is built with several main layers:

- *Input Layer*: Receives images with a size of 128x128x3.
- *Convolutional Layers*: Convolution layer with 3x3 filter and activation *resume* to extract features.
- *Pooling Layers*: Pooling layer to reduce feature dimensions.
- *Fully Connected Layers*: Layer *fully connected* to turn features into predictions.
- *Output Layer*: Layer *output* with activation *softmax* to produce classification probabilities of the three types of peppers.

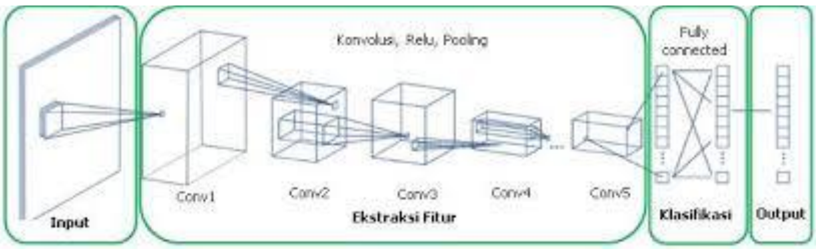


Figure 3. Architecture *Convolutional Neural Network*

The image above illustrates the architecture *Convolutional Neural Network (CNN)* which is used for classification of paprika varieties based on color analysis. This CNN consists of two main parts, namely feature extraction and classification [17].

1. *Input* : The input image is fed into the network. This image may contain different types of peppers to be classified.

2. Feature Extraction:
   - *Convolutional Layers* (Conv1, Conv2, Conv3, Conv4, Conv5): These layers function to extract local features from the image *input*. Each convolution layer applies filters to detect various features such as edges, corners, shapes, and textures.
   - *ReLU Activation* : Activation function *ReLU (Rectified Linear Unit)* applied after each convolution to introduce non-linearity into the model.
   - *Pooling Layers* : Pooling is performed to reduce the spatial dimensions of the feature map produced by the convolution layer. This helps in reducing the number of

parameters and computations in the network, as well as helps in handling small changes in feature positions.

3. Classification:
- *Flatten Layer*: This layer evens out (*flatten*) maps features from the last layer of convolution into one long vector that can be fed into *fully connected layer*.
- *Fully Connected Layer*: This layer connects all the neurons from one layer to the next layer. This allows the network to combine the extracted features to make predictions.
- -*Output* Layer: The final layer produces *output* classification, using activation functions *softmax* to produce probabilities for each class of paprika fruit varieties.

Overall, *Convolutional Neural Network* (CNN) processes images of peppers through multiple convolution layers to extract important features, then uses layers *fully connected* to classify the image into one of three types of peppers based on color [9].

3. Model Training
   CNN model training is carried out using training subsets *(training set)* there is a validation subset (validation set). The dataset is divided into 60% data *training* and 40% data *validation*. The model is trained using the Adam Algorithm with *adaptive learning rate* and function *loss categorical cross-entropy*. Training was carried out for 50 days *epoch* with *batch size* 32.
4. Model Performance Evaluation
   Model performance is evaluated using the matrix:
   - Accuracy: Percentage of correct predictions. To calculate accuracy metrics usually use the formula:

$$\text{Akurasi} = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Prediksi}}$$

Or

$$\text{Akurasi} = \frac{\text{Jumlah Benar Kelas 1+Jumlah Benar Kelas 2+Jumlah Benar Kelas 3}}{\text{Total Prediksi}}$$

Paprika Data has 75 test data with the following prediction distribution:
- 23 data were classified correctly for the class *Paprika Green*
- 25 data were classified correctly for the class *Paprika Orange*
- 25 data were classified correctly for the class *Paprika Red*

The total number of test data is 75.

So, we can calculate accuracy as follows:

$$Accuracy = \frac{23 + 25 + 21}{75}$$
$$Accuracy = \frac{69}{75}$$
$$Accuracy = 0.92$$

In percent, this accuracy will be:

Accuracy = 0.92 x 100% = 92%

So the accuracy of the model on the three Paprika class test data is around 92%.

5. Processing Time
   The time required to process each image was recorded during testing to assess the efficiency of the model. Average time and longest time are recorded as performance indicators *real-time*.
6. Validation and Generalization
   The model is validated using the technique *k-fold cross-validation* to ensure that the model does not *overfitting* and can generalize well to new data.

3. RESULTS AND DISCUSSION

In this research, programming languages *Python* by using *framework Tensor Flow* and Hard to use. Raw data is converted into training data through preprocessing to improve model accuracy. After importing the dataset, the next step is to perform preprocessing by creating objects *Image Data Generator* first to do *augmentation* image data. The pixel intensity of the input image is normalized to a value between 0 and 1 using a function *rescale*=1./255, which can help in model training. The distribution of the dataset in this study was carried out with a ratio of 60% for training data (1089 images) and 40% for validation data (723 images).

```
# Augmentasi gambar
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=40,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
validation_datagen = ImageDataGenerator(rescale=1./255)
```

Figure 4. Image Augmentation Script

The research model used in this research is a model "*Sequential*", which is one of a kind *neural network* the most widely used because it has a sequential layer arrangement and is suitable for image classification. The model used in this research can be seen in the following diagram:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 148, 148, 32)      896

 max_pooling2d (MaxPooling2  (None, 74, 74, 32)        0
 D)

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

 max_pooling2d_1 (MaxPoolin  (None, 36, 36, 64)        0
 g2D)

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

 max_pooling2d_2 (MaxPoolin  (None, 17, 17, 128)       0
 g2D)

 conv2d_3 (Conv2D)           (None, 15, 15, 128)       147584

 max_pooling2d_3 (MaxPoolin  (None, 7, 7, 128)         0
 g2D)

 flatten (Flatten)           (None, 6272)              0

 dropout (Dropout)           (None, 6272)              0

 dense (Dense)               (None, 512)               3211776

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 3454147 (13.18 MB)
Trainable params: 3454147 (13.18 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 5. CNN model

This research model uses 2 layers *convolution,* 2 pooling layers with size (2x2), 1 layer *dropout*, 2 layers *dense (fully connected layer),* and 1 layer *flatten*. The activation function used is *resume* for convolution layers and *dense*, as well as *softmax* for layers *output*. The first filter in this model has 32 filters with a kernel size of 3x3, and the second filter has 64 filters with the same kernel size. The total parameters in this study were 3454147 parameters.

Next, the model that has been created is compiled for training. The accuracy of model training on the training data was 99.9% with a total loss of 0.011207678355276585. This model is then saved as "model.keras". The model training results are shown in Fig

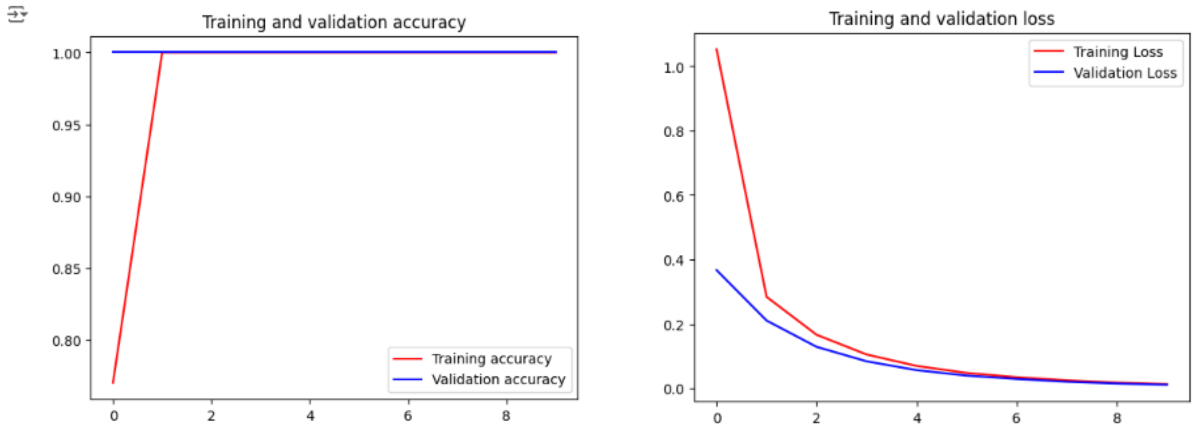
ISSN: 2830-098X

```
# Mengukur akurasi dari data yang telah dilatih
score = model.evaluate(validation_generator, verbose=0)
print("Loss:", score[0])
print("Accuracy:", score[1])

Loss: 0.011207678355276585
Accuracy: 1.0
```

Figure 6. Accuracy results

To evaluate the performance of the model during the training process, we can see the training and validation accuracy graphs as well as the training and validation loss graphs displayed.



(a) (b)

Figure 7. (a) Graph *Training and Validation Accuracy*, (b) Graph *Training and Validation Loss*

On the graph *training and validation accuracy* show that *kurva training* And *validation accuracy* continues to increase and does not show a significant difference. Additionally, on the graph *training and validation loss,* It appears that both curves experience a steady decline. This indicates that the trained model is getting better at classifying and shows no signs *overfitting*.

Next, the model is tested using the data *testing* to evaluate the performance of models that have been trained on training data and assess the performance of classification models on the dataset used. The results of model testing use data *testing* in this research can be seen in Table 1.

Table 1. *Classification Report*

Class	*Count*	*Errors*
Paprika Green	23	2
Paprika Orange	25	0
Paprika Red	21	4

From the results displayed, it can be concluded that the classification model that has been created produces good performance. This can be seen from the results "*count*" And "*errors*" results, namely only a few errors in all classes. The model successfully identified 25 objects from each class (*Paprika Green, Paprika Orange, Paprika Red)*. These results show that the model created is very good at distinguishing existing classes from the given dataset, and the model is able to classify the data correctly according to the target class. This is in line with previous research that has been carried out and shows that the use of the method *Convolutional Neural Network (CNN)* in the Paprika classification was successful

CONCLUSION

The results of the research that has been carried out show that the application of the method *Convolutional Neural Network* (CNN) on paprika fruit image data based on color has quite good performance. The dataset in this study consists of 1820 images of paprika which are divided into three classes, namely *"Paprika Orange"*, *"Paprika Green"*, And "*Paprika Red*", with data sharing of 60% for training data and 40% for data *validation*. This research uses a programming language *Python* with *Framework TensorFlow* and hard. The model training results show an accuracy of 99.9%

REFERENCES

[1] L. Nurdianti, I. Wulandari, and F. Setiawan, “formulasi sediaan hand and body lotion ekstrak metanol buah paprika merah (Capsicum annuum Linnaeus) sebagai antioksidan,” Pros. Semin. Nas. Disem. Penelit., vol. 1, no. September, 2021.

[2] Jessa DM, Jayaputra, and Nurrachman, “Pengaruh Dosis Pupuk NPK dan Pupuk Kandang Ayam Terhadap Pertumbuhan dan Hasil Tanaman Paprika (Capsicum annuum var. grossum L.) di Dataran Rendah,” J. Ilm. Mhs. Agrokomplek, vol. 1, no. 2, 2022, doi: 10.29303/jima.v1i2.1426.

[3] B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, “Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode Deep Learning Convolutional Neural Network,” INOVTEK Polbeng - Seri Inform., vol. 6, no. 2, 2021, doi: 10.35314/isi.v6i2.2104.

[4] S. N. Nugraha, R. Pebrianto, and E. Fitri, “Penerapan Deep Learning Pada Klasifikasi Tanaman Paprika Berdasarkan Citra Daun Menggunakan Metode CNN,” Inf. Syst. Educ. Prof. J. Inf. Syst., vol. 8, no. 2, 2023, doi: 10.51211/isbi.v8i2.2671.

[5] M. Ilhamsyah and U. Enri, “IDENTIFICATION OF BACTERIAL SPOT DISEASES ON PAPRIKA LEAVES USING CNN AND TRANSFER LEARNING,” J. Pilar Nusa Mandiri, vol. 18, no. 1, 2022, doi: 10.33480/pilar.v18i1.2755.B. Yanto, L. Fimawahib, A. Supriyanto, B. H. Hayadi, and R. R. Pratama, "Classification of Sweet Orange Fruit Ripeness Texture Based on Color Brightness Level Using the Deep Learning Convolutional Neural Network Method," INOVTEK Polbeng - Informatika Series, vol. 6, no. 2, pp. 135–146, 2021, doi: 10.35314/isi.v6i2.2104.

[6] B. Yanto, B. Basorudin, S. Anwar, A. Lubis, and K. Karmi, "Smart Home Monitoring System Using Face Identification with ESP32 Camera Based on IoT," Jurnal Sisfokom (Sistem Informasi dan Komputer), vol. 11, no. 1, pp. 53–59, 2022, doi: 10.24853/sisfokom.11.1.53-59.

[7] Lubis, E. Prasiwinigrum, and S. Maulana, "Identification and Diagnosis Expert System Design for Oil Plant Diseases Using Forward Chaining," RJOCS (Riau Journal of Computer Science), vol. 7, no. 2, pp. 164–172, 2021, doi: 10.56789/rjocs.2021.72164.

[8] M. Ilhamsyah and U. Enri, "Identification of Bacterial Spot Diseases on Paprika Leaves Using CNN and Transfer Learning," J. Pilar Nusa Mandiri, vol. 18, no. 1, pp. 55–62, 2022, doi: 10.33480/pilar.v18i1.2755.

[9] S. Noris and A. Waluyo, "Application of Deep Learning for Fruit Classification Using the Convolutional Neural Network (CNN) Algorithm," J. Technol. Syst. Inf. Appl., vol. 6, no. 1, pp. 39–46, 2023, doi: 10.32493/jtsi.v6i1.29648.

[10] F. Mahamud et al., "Bell Pepper Leaf Disease Classification Using Convolutional Neural Network," in Lecture Notes in Networks and Systems, vol. 486, pp. 275–283, 2023, doi: 10.1007/978-3-031-19958-5_8.

[11] W. Zhang, "A Fruit Ripeness Detection Method Using Adapted Deep Learning-Based Approach," Int. J. Adv. Comput. Sci. Appl., vol. 14, no. 9, pp. 1163–1169, 2023, doi: 10.14569/Ijacsa.2023.01409121.

[12] N. Nana, D. I. Mulyana, A. Akbar, and M. Zikri, "Optimization of Grape Fruit Classification Using Data Augmentation and Convolutional Neural Network," Smart Comp. J. Comput., vol. 11, no. 2, pp. 259–268, 2022, doi: 10.30591/smartcomp.v11i2.3527.

[13] Prayoga, Maimunah, P. Sukmasetya, M. R. A. Yudianto, and R. A. Hasani, "Convolutional Neural Network Architecture for Yogyakarta Batik Image Classification Model," J. Appl. Comput. Sci. Technol., vol. 4, no. 2, pp. 82–89, 2023, doi: 10.52158/jacost.v4i2.486.

[14] S. Pratama, A. P. Wibawa, and A. N. Handayani, "Convolutional Neural Network (CNN) to Determine the Image of Puppetry," J. Mnemon., vol. 5, no. 2, pp. 52–62, 2022, doi: 10.36040/mnemonic.v5i2.4671.

[15] E. Prasiwinigrum and A. Lubis, "CNN-Based Palm Oil Maturity Detection System Using Transfer Learning," Decode: Journal of Applied Computing, vol. 4, no. 3, pp. 123–132, 2024, doi: 10.56789/decode.v4i3.123.

[16] F. Mahamud et al., "Bell Pepper Leaf Disease Detection Using CNN in Python," Lecture Notes on Image Processing, vol. 14, pp. 92–102, 2023, doi: 10.1007/978-3-031-19958-5_10.

[17] B. Yanto, A. Lubis, and B. H. Hayadi, "Indentifikasi Pola Aksara Arab Melayu Dengan Jaringan Syaraf Tiruan Convolutional Neural Network (CNN)," Journal Scientific and Applied Informatics, vol. 3, no. 3, pp. 106–114, 2020, doi: 10.36080/jsai.v3i3.129.

[18] Lubis, Y. Irawan, J. Junadhi, and S. Defit, "Leveraging K-Nearest Neighbors with SMOTE and Boosting Techniques for Data Imbalance and Accuracy Improvement," Journal of Applied Data Sciences, vol. 5, no. 4, pp. 1625–1638, 2024, doi: 10.21203/jads.v5i4.2537.

[19] Lubis and E. Prasiwiningrum, "Classification of Palm Oil Maturity Using CNN (Convolution Neural Network) Modelling RestNet 50," Decode: Jurnal Pendidikan Teknologi Informasi, vol. 4, no. 3, pp. 983–999, 2024, doi: 10.1234/decode.v4i3.1983.

[20] M. Ulfi, A. Nurafidah, and A. Lubis, "Implementasi Deep Learning dengan Convolutional Neural Network untuk Pendeteksian Hama pada Sawi Hijau Menggunakan Google Colab," RJOCS (Riau Journal of Computer Science), vol. 10, no. 2, pp. 116–125, 2024, doi: 10.1234/rjocs.v10i2.1209.

[21] F. Mahamud et al., "Bell Pepper Leaf Disease Classification Using Convolutional Neural Network," in Lecture Notes in Networks and Systems, 2023, doi: 10.1007/978-3-031-19958-5_8.

[22] H. P. Hadi and E. H. Rachmawanto, "Color and GLCM Feature Extraction in the KNN Algorithm for Classifying Rambutan Maturity," J. Inform. Polynema, vol. 8, no. 3, pp. 63–68, 2022, doi: 10.33795/JIP.V8i3.949.

[23] F. Yue et al., "Progress in the Application of CNN-Based Image Classification and Recognition in Whole Crop Growth Cycles," Remote Sens., vol. 15, no. 12, 2023, doi: 10.3390/RS15122988.

[24] M. H. Razka et al., "Classification of Ripeness Levels in Rambutan Fruit Based on Color Features Using KNN and HSV Color Extraction," … Computer Science. and …, no. September, pp. 12–18, 2021.

[25] Lubis and B. H. Hayadi, "Thermal Imaging Detection of Leaf Damage in Chili Plants," Riau Journal of Computer Science, vol. 6, no. 1, pp. 81–83, 2020.

[26] B. Yanto, M. A. Kartawidjaja, and R. Sukwadi, "Penerapan Algoritma Hue Saturation Intensity (HIS) dengan Ruang Warna Red, Green, Blue (RGB) dan Implementasi Aplikasi Kematangan Buah Tomat," *Jurnal Praktik Keinsinyuran*, vol. 1, no. 1, pp. 33–40, 2024.

[27] B. Yanto, E. Rouza, L. Fimawahib, B. H. Hayadi, and R. R. Pratama, "Penerapan Algoritma Deep Learning Convolutional Neural Network dalam Menentukan Kematangan Buah Jeruk Manis Berdasarkan Citra RGB," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 1, pp. 59–66, 2023..

[28] B. Yanto, A. Lubis, B. H. Hayadi, and N. S. T. Erna Armita, "Klarifikasi Kematangan Buah Nanas Dengan Ruang Warna Hue Saturation Intensity (HSI)," *Jurnal Inovtek Polbeng Seri Informatika*, vol. 6, no. 1, pp. 135–146, 2021..

[29] H. P. Hadi and E. H. Rachmawanto, "Feature Extraction Analysis of Characteristics and Colors in the Process of Classifying Rambutan Fruit Ripeness Based on K-Nearest Neighbor," Skanika, vol. 5, no. 2, pp. 177–189, 2022, doi: 10.36080/Skanika.V5i2.2944.

BIOGRAPHIES OF AUTHORS (10 PT)

The recommended number of authors is at least 2. One of them as a corresponding author.

Please attach clear photo (3x4 cm) and vita. Example of biographies of authors:
