International Journal of Informatics and Computation (IJICOM)
Vol.7, No.2, 2025
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Robust Maturity Level Classification of Bell Pepper using CNN

Rino Aziz Fadhillah[1], Irwansyah[2]

Abstract

The development of artificial intelligence has opened up new opportunities in various fields, including Bell Pepper image detection. The remaining issues are that the selection of the ripeness level of bell peppers manually can take a long time and requires more accuracy. The purpose of this research is to classify the maturity level of bell peppers and to determine the level of accuracy. The research method used is Convolutional Neural Network (CNN) with tools or tools, namely Visual Studio Code in Python with TensorFlow Framework, as well as a pre-trained CNN architecture called VGG16. Bell peppers are divided into 3 levels of ripeness with different types of colors: green (unripe), yellow (half-ripe), and red (ripe). The results showed that in classifying the maturity level of bell peppers with an accuracy of 89%, precision 84%, recall of 83%, and F1-Score of 84%.

Keywords:
Ripeness, Classification, Bell Peppers, CNN, VGG16

1. Introduction

Bell peppers (Capsicum annuum) are an essential agricultural commodity widely consumed and valued for their nutritional and economic benefits. Accurately identifying the ripeness stage, ranging from unripe (green), semi-ripe (yellow/orange), to fully ripe (red), is crucial for optimizing harvesting schedules, post-harvest handling, and market distribution. Traditional ripeness classification relies heavily on human inspection, which is subjective and inconsistent. To address this, image-based classification using computer vision and deep learning offers a promising solution, enabling automatic and objective detection based on color, texture, and shape features captured from digital images [1].

Despite the growing adoption of machine learning (ML) methods in agricultural applications, several limitations hinder their effectiveness in real-world scenarios. One major weakness lies in their reliance on large, well-labeled, and balanced datasets. In practice, acquiring high-quality image data with accurate ripeness labels for bell peppers is labor-intensive and often limited, leading to class imbalance. The imbalance samples can cause models to favor majority classes while underperforming on minority ones, especially problematic when distinguishing between visually similar stages like semi-ripe and ripe peppers. Moreover, ML models are sensitive to variations in lighting, occlusion, and background noise, which are common in uncontrolled agricultural environments, reducing the model's generalizability across different conditions [2].

Additionally, traditional ML models often require manual feature engineering, which limits their ability to capture complex, hierarchical patterns within the image data. Even with deep learning techniques like CNNs, performance can degrade when datasets include subtle variations or anomalies that confuse the network. Furthermore, ML models typically

---

Corresponding Author: Irwansyah(irwansyah@uhamka.ac.id)
1 Rino Aziz Fadhillah, Universitas Muhammadiyah Prof. Dr. HAMKA, rinoaziz.ra@gmail.com
2 Irwansyah, Universitas Muhammadiyah Prof. Dr. HAMKA, Irwansyah@uhamka.ac.id

operate as “black boxes,” offering little interpretability, making it difficult for domain experts (e.g., farmers or agronomists) to understand and trust the system’s predictions. The lack of transparency becomes a significant barrier in decision-critical tasks like harvest timing or quality control. To address these challenges, ongoing research explores techniques such as explainable AI (XAI), transfer learning, and synthetic data generation, yet these solutions introduce new layers of complexity [3], [4].

CNNs have become the leading approach in visual recognition tasks, including fruit classification. Their ability to extract hierarchical features from raw images makes them highly effective in distinguishing fine-grained visual differences, such as those present between different ripeness stages of bell peppers. Several studies have successfully employed CNN architectures such as VGG16, InceptionV3, and MobileNet for agricultural applications, achieving high accuracy in classifying fruits like apples, bananas, and tomatoes. These models reduce the need for handcrafted features, relying instead on automated feature learning, which is particularly effective in datasets with complex visual patterns [3][4][5].

In recent developments, researchers have explored combining CNNs with data augmentation techniques to improve model robustness and generalization. Additionally, transfer learning from pre-trained networks has shown improved performance even with limited training data, a common challenge in agricultural datasets. Despite these advancements, challenges remain, such as class imbalance and visual overlap between maturity stages, which may reduce classification performance. Thus, improving CNN-based classification systems for bell pepper ripeness is a valuable contribution to smart agriculture, offering increased efficiency and accuracy in real-time field applications [6], [7].

2. Related Works

In this study, the author searched for relevant references to support the creation of scientific papers. References were selected based on objectives that were in line with the research being conducted. Research by Qudsiah Nur Azizah (2023) successfully classified diseases in corn leaves using the AlexNet architecture and achieved an accuracy of 90%, proving the effectiveness of this method in pattern recognition in corn leaf images [1]. Furthermore, another paper developed an Android-based CNN application to classify the ripeness level of apples quickly and accurately, as well as provide easy access for farmers in the harvesting and processing of fruits [2].

Several articles researched detecting the ripeness level of papaya fruit using the YOLO algorithm on the Android platform. This method was chosen for its ability to detect objects quickly and accurately, as well as efficiently provide object location information. Another article used a digital image processing approach and fuzzy logic to identify the ripeness level of red dragon fruit. This research achieved an accuracy of 96.42% with an error rate of 3.57% through the use of the Mamdani method and features such as contrast, correlation, energy, homogeneity, mean, variance, and entropy [3][4].

The latest research developed a classification system for the ripeness of oil palm fruits using a CNN with a MobileNet architecture. The dataset used consisted of 300 images divided into three classes: unripe, ripe, and rotten. The best results from five testing scenarios showed 100% training and testing accuracy, with perfect recall, precision, and F1-score values. All these studies demonstrate that deep learning approaches are highly effective for visual classification tasks in the agricultural field [17].

3. Proposed Method

CNN is used as a model in classifying the ripeness of bell peppers that consists of convolution layers, rectification, and fully connected layers for feature extraction and classification [2]. CNN is an algorithm that can train large data sets with millions of parameters and take 2D image forms as input. CNN uses convolution operations with filters to produce the desired output. CNN can automatically extract features from image data without the need for manual feature engineering. This improves efficiency and reduces the need for manual processes. The convolution layers in CNN provide translation invariance, enabling CNN to identify and extract patterns and features from data regardless of variations in position, orientation, scale, or translation. In addition to image classification tasks, CNN can also be applied to various other domains, such as natural language processing, time series analysis, and speech recognition.

CNNs also use regularization techniques such as dropout and batch normalization because they help reduce the potential for overfitting, especially when data is scarce. Dropout works by randomly deactivating neurons during training. This prevents the model from becoming dependent on specific parts. Batch normalization helps balance and accelerate training by normalizing the output from the previous layer [3]. In terms of resource usage, CNNs utilize weight sharing among neurons in the convolutional layer. This significantly reduces the number of parameters compared to fully connected networks. As a result, training becomes faster and memory requirements decrease, without compromising model performance [4]. In recent years, CNNs have been suitable for image segmentation and object detection tasks [5].

VGG16 uses small 3x3 convolution filters to process images. In its implementation, VGG-16 arranges several 3x3 convolution layers in sequence [6]. The VGG-16 architecture itself consists of 16 layers, including 13 convolutional layers, 2 fully connected layers, and 1 output layer for classification [7]. The detailed structure of VGG-16 can be seen in Fig. 1

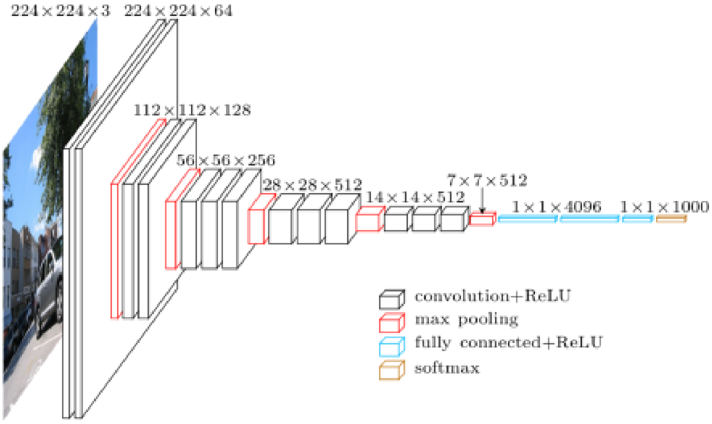


Fig. 1. Details of the VGG-16 model structure

VGG-16 is designed to create a deeper and more complex model without sacrificing ease of use. One of the main features of VGG-16 is the use of small 3x3 convolution filters, which allow the model to capture local information in greater detail and depth. Despite using small filters, the stacking of these convolutional layers allows VGG-16 to capture various levels of visual features from images [8]. The VGG-16 architecture also has advantages such as ease of reuse in feature extraction. The stacked convolution layers can be easily adapted to various other tasks through transfer learning, especially by using pretrained models that have been trained on large datasets such as ImageNet. This allows the model to achieve good performance even when using limited datasets, such as in the case of pepper ripeness classification, which has only 238 images per class [9].

The VGG-16 architecture is a deep CNN model designed to perform image classification tasks, including the classification of bell pepper ripeness. Mathematically, the model transforms an input image $X \in \mathbb{R}^{H\times W\times C}$ (with height H, width W, and channels C) into a probability vector $\hat{y} \in \mathbb{R}^{K}$, where K is the number of output classes (e.g., ripe, semi-ripe, raw). In Convolutional Layer Operation, each convolutional layer applies a set of filters $W \in \mathbb{R}^{f\times f\times C}$ to extract features from the image:

$$Z^{(l)} = W^{(l)} * A^{(l-1)} + b^{(l)} \quad (1)$$

where:

- $Z^{(l)}$ is the output before activation at layer l,
- $A^{(l-1)}$) is the activation from the previous layer,
- $*$ denotes the convolution operation,
- $b^{(l)}$) is the bias term.

After convolution, a non-linear ReLU activation is applied:

$$A^{(l)} = \max(0, Z^{(l)}) \quad (2)$$

In the Pooling Layer, to reduce the spatial dimensions, max pooling is applied:

$$P^{(l)} = \max_{i,j}(A_{i,j}^{(l)}) \quad (3)$$

This helps in reducing computation and controlling overfitting while preserving dominant features. After several convolution and pooling layers, the output is flattened into a 1D vector and passed through fully connected (dense) layers:

$$\begin{gathered} z^{(fc)} = W^{(fc)}x + b^{(fc)} \\ a^{(fc)} = \text{ReLU}(z^{(fc)}) \end{gathered} \quad (4)$$

The final layer applies the SoftMax function to obtain class probabilities:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, i = 1, \dots, K \quad (5)$$

Where $\hat{y}_i$ is the predicted probability of class i. The predicted class is the one with the highest probability. When applied to bell pepper classification, VGG-16 takes preprocessed images (e.g., resized to 224×224×3) and propagates them through its 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers. The model learns to extract features such as color, shape, and texture, which are crucial for distinguishing between raw, semi-ripe, and ripe peppers. During training, the model minimizes a loss function such as categorical cross-entropy to optimize its weights and improve classification accuracy.

Although VGG-16 tends to be more computationally intensive due to its large number of parameters, this architecture has proven to be highly effective in various computer vision tasks such as object classification and face detection. One of its drawbacks is its high memory consumption, particularly due to the large number of convolutional layers that generate a significant number of parameters. However, this issue can be addressed using techniques such as fine-tuning or transfer learning with a pre-trained model [10]. Another advantage is that the VGG-16 has a clear and consistent structure, with an easy-to-understand division of layers [11].

4. Experimental Setup

4.1 Dataset

In this study, we gathered a dataset of a total of 444 images as training data and 90 images as test or validation data taken from Roboflow. The purpose of data collection is to obtain accurate and reliable information that can be used for analysis, decision-making, and further research. It is important to ensure that the data collection methods are appropriate for the research objectives and that the data collected is valid and reliable [12].

4.2 Preprocessing

In the classification of bell pepper ripeness using CNN, preprocessing is a critical step to ensure the quality and consistency of input data fed into the model. The preprocessing stage typically begins with image resizing, where all images are scaled to a uniform resolution, such as 224x224 pixels. This ensures compatibility with standard CNN architectures like VGG16 or ResNet, which require fixed input dimensions. Resizing also helps reduce computational load while preserving the visual features essential for classification.

At this stage, we apply image normalization to scale pixel values commonly from a range of 0–255 to a range of 0–1 or standardized to have zero mean and unit variance. This improves the convergence of the learning algorithm by ensuring that all features contribute equally to the training process. In addition, image augmentation techniques are often utilized to increase the diversity of the training dataset. Techniques such as horizontal and vertical flipping, rotation, zooming, and brightness adjustments help the model generalize better and reduce overfitting, especially in scenarios where the dataset is relatively small or imbalanced across ripeness classes. Lastly, the images are labeled according to the ripeness level, categorized as *raw*, *semi-ripe*, and *ripe*. These labeled images are then converted into one-hot encoded vectors for multi-class classification. If class imbalance is detected, we adopt synthetic techniques such as SMOTE to balance the dataset. These preprocessing strategies collectively enhance model robustness and accuracy in detecting subtle differences in pepper ripeness stages [13].

4.3 Training and Evaluation Model

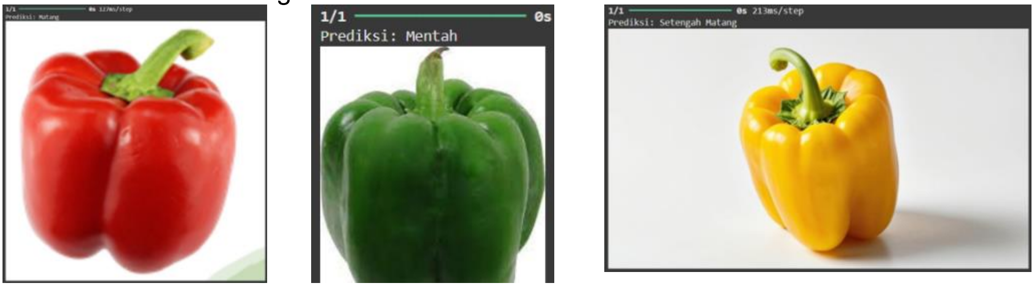
Modeling CNN involves preparing image data, selecting an architecture consisting of convolution, pooling, and fully connected layers, training the model to learn patterns from the training data using an optimization algorithm, validating to adjust the model parameters, and finally testing to evaluate the model's performance on the test data. This is important for object classification and detection in images [14]. Model training refers to the process in which a machine learning or deep learning model is updated with data so that it can learn and adapt to produce better results. This process involves feeding training data to the model, which then uses a CNN algorithm to adjust its internal parameters to match the data. The purpose of model training is to train the model to predict or classify data with accuracy [15]. Model evaluation is an important process in machine learning to measure model performance and accuracy using various metrics and techniques. The goal is to assess how well the model can generalize prediction results to previously unseen data. Common evaluation metrics include accuracy, loss for classification problems, and other metrics appropriate for classification problems [16].

5. Results and Analysis

In this study, we conduct training of a CNN model using the Adam optimizer, with ReLU and Softmax activation functions on the output layer with 50 epochs and a batch size = 32.

5.1 Model Testing

To conduct the testing stage, we utilize the new images from outside the dataset (manual testing). The model processes input images from users via the image upload interface, then outputs label predictions as classification results. Several examples of model prediction results can be seen in Fig. 2:



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 117ms/step
Prediksi: Matang
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s
Prediksi: Mentah
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 213ms/step
Prediksi: Setengah Matang
```

Fig. 2. Sample prediction result

The prediction results show that the model can classify the given images well into one of three maturity level classes. With an accuracy of over 89%, this model is suitable for practical applications in classifying the ripeness of peppers. The results show that the model can classify peppers quite well in new test images. From the training results, it can be seen that the model achieved a training and validation accuracy of 89.0110%. Table 1 depicts the classification reports of the classification model.

Table 1: Classification report of Bell Peppers using CNN

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Ripe (Matang)	0.79	0.72	0.76	47
Unripe (Mentah)	0.96	0.91	0.93	47
Half-ripe (Setengah Matang)	0.79	0.86	0.82	64
Accuracy			0.84	158
Macro Avg	0.84	0.83	0.84	158
Weighted Avg	0.84	0.84	0.84	158

5.2 Evaluation Metrics

The evaluation was conducted using a previously separated validation dataset. At this stage, the Confusion Matrix and Classification Report methods were used to assess the model's performance for each class. Fig. 3depict the results of the classification report of the classification model.

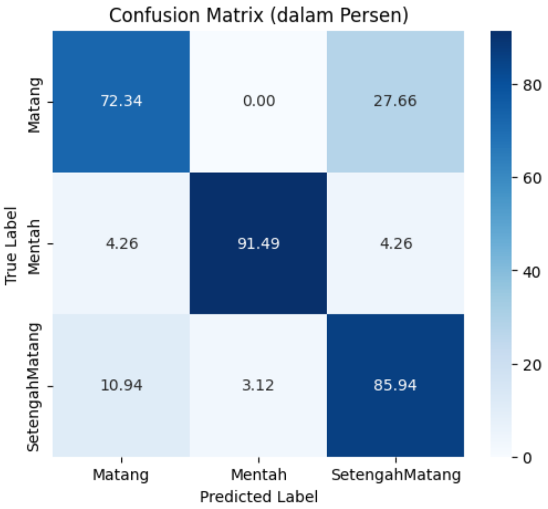


Fig. 3. Confusion Matrix

The evaluation results indicate that the model demonstrates strong and consistent performance, particularly in identifying the Semi-Ripe and Unripe bell pepper classes. However, its performance in the Ripe class is relatively lower. This limitation is likely due to the high visual similarity between Ripe and Semi-Ripe samples, especially in terms of color characteristics. The confusion matrix and classification report reveal that precision, recall, and F1-score for all classes exceed 70%, with minimal misclassification.

The model successfully detects the ripeness level of bell peppers with high accuracy. Several contributing factors include the use of data augmentation to improve generalization, integration of the pre-trained VGG-16 architecture for feature extraction, implementation of dropout to reduce overfitting, and careful optimization of the CNN structure. These design choices have led to a model that performs reliably for classification tasks in agricultural contexts.

6. Conclusion

Based on research conducted on the classification of bell pepper ripeness levels using deep learning methods, it can be concluded that the developed application successfully identified bell pepper ripeness levels automatically into three categories: unripe, semi-ripe, and ripe. This research achieved an accuracy of 89.001%, with a precision value of 84%, a recall value of 83%, and an F1-score of 84%. Further research is recommended to implement and compare other algorithms outside of CNN, such as GAN, DBN, etc., to evaluate the classification performance of various approaches. Expanding the dataset by adding variations in lighting conditions, backgrounds, and types of peppers is highly recommended so that the model has stronger generalization capabilities.

Acknowledgment

The author would like to express his deepest gratitude to Muhammadiyah University, Prof. DR. HAMKA (UHAMKA), for providing facilities and support in conducting this research. Special thanks are extended to Mr. Irwansyah, S.Kom., M.Kom., as the supervisor, for all the guidance, direction, and valuable motivation provided throughout the research process. The author also extends heartfelt thanks and appreciation to family and friends for their moral support, encouragement, and prayers throughout this process.

References

[1] I. Fathurrahman and A. M. Nur, "Identification of Cucumber Fruit Ripeness Based on Digital Image Using Backpropagation Neural Network," *Journal of Informatics and Technology*, vol. 2, no. 1, p. 27, 2019.

[2] R. Namruddin, "Classification of Apple Fruit Freshness Using Android-Based Convolutional Neural Network (CNN) Method."

[3] A. Lutfhi, "The Effect of Layer Batch Normalization and Dropout on CNN Model Performance for Facial Expression Classification," *International Journal on Informatics Visualization*. [Online]. Available: www.joiv.org/index.php/joiv

[4] A. Y. Yulestiono, M. M. Subagio, M. S. Bhakti, and A. P. Sari, "Performance Comparison of Convolutional Neural Network (CNN) and VGG-16 Methods in Traffic Sign Classification."

[5] D. Gunawan and H. Setiawan, "Convolutional Neural Network in Medical Image Analysis," 2022.

[6] A. S. Mashuri and A. Sunyoto, "Chili Leaf Disease Classification Using VGG16 Architecture," *Journal of Electrical Engineering and Computer (JEECOM)*, vol. 6, no. 2, 2024, doi: 10.33650/jeecom.v4i2.

[7] M. Alruily et al., "Ensemble Deep Learning for Alzheimer’s Disease Diagnosis Using MRI: Integrating Features from VGG16, MobileNet, and InceptionResNetV2 Models," *PLoS One*, vol. 20, no. 4, pp. e0318620, Apr. 2025. [Online]. Available: https://doi.org/10.1371/journal.pone.0318620

[8] D. Sumarlie, C. Lubis, and T. Handhayani, "Recognition of Indonesian Traditional Cakes Using Convolutional Neural Network Algorithm," 2022. [Online]. Available: https://arxiv.org/abs/1409.1556

[9] A. B. Wicaksono and P. Hartato, "Performance Analysis of CNN Inceptionv3 and VGG16 Architectures in Brain Cancer Detection Classification," vol. 10, no. 2, pp. 938–948, 2025, doi: 10.29100/jipi.v10i2.6090.

[10] M. F. Syahid, "Implementation of Deep Learning VGG16 with Transfer Learning in Cassava Plant Disease Detection."

[11] N. Rahmatullah, A. P. Dharma, D. A. W. Safitri, and I. Kurnia, "Field Work Practice Guide as a Subject in the Implementation of the Independent Curriculum."

[12] H. A. Saputri, Zulhijrah, N. J. Larasati, and Shaleh, "Assessment Instrument Analysis: Validity, Reliability, Difficulty Level, and Item Discrimination Power," 2023.

[13] A. Putri, B. S. Negara, and S. Sanjaya, “Penerapan Deep Learning Menggunakan VGG-16 untuk Klasifikasi Citra Glioma,” *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 3, no. 4, p. 379, Jun. 2022, doi: 10.30865/json.v3i4.4122.

[14] E. Purnama Giri, *Convolutional Neural Network: Concepts, Applications, and Implementation with Experimental Examples*, 2021.

[15] R. A. Tilasefana and R. E. Putra, "Application of Deep Learning Method Using CNN Algorithm with VGG NET Architecture for Weather Recognition," *Journal of Informatics and Computer Science*, vol. 5, 2023.

[16] H. N. Asna, F. S. Papilaya, and C. Fibriani, "Analysis of Agricultural Land Function Transfer to Housing and Industrial Areas in Semarang City Using Remote Sensing and Geographic Information System Approach," 2020.

[17] R. Triyogi, R. Magdalena, and B. Hidayat, "Classification of Oil Palm Fruit Ripeness Using MobileNet Convolutional Neural Network Architecture," Jurnal Ilmiah Teknik Informatika, vol. 9, no. 2, pp. 123–130, 2023