JURNAL RISET INFORMATIKA **P-ISSN: 2656-1743 |E-ISSN: 2656-1735**
Vol. 8, No. 4. September 2026 DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

HYBRID SOBEL-BASED CONVOLUTIONAL NEURAL NETWORK AND LEARNING VECTOR QUANTIZATION FOR CHILI SEED IMAGE CLASSIFICATION

Gaudensia Asni Makul[-1], Supatman[-2]

Program Studi Informatika
Fakultas Teknologi Informasi
Universitas Mercu Buana Yogayakarta
221110048@student.mercubuana-yogya.ac.id[-1], supatman@mercubuana-yogya.ac.id[-2]

Abstract

This study develops an image classification system for chili pepper seeds using a hybrid approach that combines a manual Convolutional Neural Network (CNN) based on Sobel filters and Learning Vector Quantization (LVQ). The dataset consists of 900 images from three chili pepper varieties, namely gendot, raksasa, and rawit, with 300 images for each class captured using a smartphone camera. Unlike conventional CNN architectures that rely on trainable convolutional kernels, this study adopts predefined Sobel filters as fixed feature extractors combined with Learning Vector Quantization as the classifier, resulting in a lightweight classification framework that does not require a training process in the feature extraction stage. The CNN architecture employs three convolution-pooling layers (depth-3), producing feature maps of size 16×16, which are then transformed into 256-dimensional feature vectors through a flattening process. LVQ training was conducted for 30 epochs with an initial learning rate of 0.001 and five prototypes assigned to each class. The model was evaluated using a 5-fold cross-validation scheme and obtained a mean accuracy of 80.78% with a standard deviation of 4.07% across folds. Based on the combined confusion matrix from all five folds, the gendot class obtained a precision of 1.00 and a recall of 0.93, the raksasa class obtained a precision of 0.75, a recall of 0.69, and an F1-score of 0.72, and the rawit class obtained a precision of 0.71, a recall of 0.80, and an F1-score of 0.75, with most misclassifications occurring between the raksasa and rawit classes due to their morphological similarity. These findings show that the hybrid CNN–LVQ approach can be applied to chili seed classification without requiring specialized hardware or a pre-trained model.

Keywords: Chili Pepper Seeds; Convolutional Neural Network (CNN); Sobel Filter; Learning Vector Quantization (LVQ)

Abstrak

Penelitian ini membangun sistem klasifikasi citra biji cabai menggunakan metode hybrid Convolutional Neural Network (CNN) berbasis filter Sobel dan Learning Vector Quantization (LVQ). Dataset terdiri dari 900 citra dari tiga varietas cabai, yaitu gendot, raksasa, dan rawit, dengan masing-masing kelas berjumlah 300 citra yang diambil menggunakan kamera smartphone. Berbeda dengan model CNN konvensional yang mengandalkan kernel konvolusi terlatih, penelitian ini menggunakan filter Sobel yang telah ditentukan sebelumnya (predefined) untuk ekstraksi ciri dan memanfaatkan Learning Vector Quantization untuk klasifikasi, sehingga menghasilkan kerangka klasifikasi yang ringan secara komputasi. Arsitektur CNN menerapkan 3 lapisan konvolusi-pooling (depth-3) yang menghasilkan peta ciri berukuran 16×16, kemudian diubah menjadi vektor ciri berdimensi 256 melalui proses flattening. Pelatihan LVQ dilakukan selama 30 epoch dengan learning rate awal sebesar 0,001 dan menggunakan 5 prototipe pada setiap kelas. Model dievaluasi menggunakan skema 5-fold cross validation dan memperoleh rata-rata akurasi sebesar 80,78% dengan standar deviasi 4,07% antar fold. Berdasarkan confusion matrix gabungan dari kelima fold, kelas gendot memperoleh precision 1,00 dan recall 0,93, sedangkan kelas raksasa memperoleh precision 0,75, recall 0,69, dan F1-score 0,72, serta kelas rawit memperoleh precision 0,71, recall 0,80, dan F1-score 0,75, dengan kesalahan klasifikasi yang dominan terjadi di antara kelas raksasa dan rawit akibat kemiripan ciri tepi kedua varietas tersebut. Hasil penelitian ini menunjukkan bahwa pendekatan hybrid CNN–LVQ dapat digunakan untuk klasifikasi biji cabai tanpa memerlukan perangkat khusus maupun model terlatih.

Kata kunci: Biji Cabai; Convolutional Neural Network (CNN); Filter Sobel; Learning Vector Quantization(LVQ)

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

INTRODUCTION

Chili is one of the horticultural crops that plays an important role in meeting the food needs of the Indonesian population. Based on data from the Center for Agricultural Data and Information Systems of the Ministry of Agriculture in 2024, national chili production during the 1990–2023 period fluctuated but tended to increase, with an average growth rate of 7.71% per year. In 1990, Indonesia's chili production was recorded at 569,604 thousand tons, then increased to 3.06 million tons in 2023. The chili harvest area in Indonesia over the same period also increased, with an average growth rate of 2.51% per year, with the highest increase occurring in 2017 at 19.19%, reaching 310.15 thousand hectares. High market demand, both for household consumption, traditional markets, the food industry, and exports, has driven the increase in national chili production from year to year(Kementrian Pertanian, 2024).

In addition to its important role in the food sector and the economy, chili also has various health benefits, such as helping to control chronic disease and potentially reducing the risk of death from non-communicable disease (Sello & Lekatsa, 2023). Chili also contains nutrients such as vitamin C and capsaicin(Alhanannasir et al. 2025). In agriculture, chili is used as an environmentally friendly botanical pesticide (Khoerunisa et al., 2024). This condition shows that the increase in chili production needs to be supported not only in terms of quantity, but also by maintaining the quality of the chili produced. This effort is strongly influenced by the cultivation technique used. The choice of chili cultivation method needs to be adapted to the plant's morphological characteristics so that it can adapt to the environmental conditions in which it grows (Safitri, 2024), which can result in decreased productivity and economic losses (Kementrian Pertanian, 2024).

Therefore, identifying chili varieties from the seedling stage onward is important to help determine the appropriate cultivation technique. If seed variety identification is not carried out accurately, there is a risk of errors in selecting the cultivation method, which can affect plant growth, harvest quality, and chili productivity. Efforts to maintain chili quality through the application of proper cultivation practices can be supported by a variety identification process based on seeds from the seedling stage onward, one of which is through the use of computer vision technology (Zhao et al., 2024)

Various studies have applied the Convolutional Neural Network (CNN) method for classification in chili plants, such as chili type recognition based on fruit morphology (Perlindungan & Risnawati, 2020), disease detection in chili fruit (Nurbuana et al., 2024), ripeness level identification based on color (Tosi et al., 2024), and leaf disease identification using a combination of CNN and Learning Vector Quantization (LVQ) (Siswipraptini et al., 2023). These studies show that CNN is a method that automatically extracts morphological features from plant images, and LVQ is able to classify them well.

However, research specifically addressing classification based on chili seed morphology is still very limited. One study that has been conducted is that of Pebrianto and Haryanto (Pebrianto & Haryanto, 2023), who classified three chili seed varieties using a Python framework-based CNN with image acquisition via a flatbed scanner and a dataset of 150 images. That study achieved an accuracy of around 90% and showed that the CNN architecture could distinguish the three types of chili seeds well. However, that approach has several limitations, such as its dependence on special hardware in the form of a flatbed scanner and a GPU, a still-limited dataset size, and a lack of computational transparency because the entire feature extraction process was carried out automatically by the framework without explicitly showing the convolution mechanism.

The use of a flatbed scanner (Pebrianto & Haryanto, 2023) specifically limits the application of that method under field conditions, since this device is not always available to farmers or small-scale seed providers, while the limited dataset size has the potential to reduce the model's generalization ability across image variations under different capture conditions. Therefore, an easily accessible classification method that does not depend on special hardware is needed so that it can be widely applied by farmers and seed providers alike.

Based on these issues, this study proposes a hybrid approach between a Sobel filter-based CNN (non-trainable convolution) and Learning Vector Quantization (LVQ) for chili seed image classification. This approach is designed to run without special hardware, using images captured with a smartphone camera and a larger dataset, namely 900 images, or 300 images per class. CNN is used as an edge feature extractor based on Sobel filters with a depth of 3 layers, while LVQ is used as the final classifier to replace the fully connected layer in a conventional CNN architecture. The combination of a Sobel filter-based CNN and LVQ was chosen because the Sobel filter can consistently

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

extract edge features without requiring a training process that consumes large computational resources (Linse et al., 2023), while LVQ offers a simple, easily interpretable prototype-based classification mechanism that requires neither a fully connected layer nor a backpropagation process (Sardogan et al., 2018), so that the combination of the two aligns with the research objective of producing a lightweight classification system that can run without special hardware.

RESEARCH METHODS

This study uses a descriptive approach with the research flow arranged as follows:

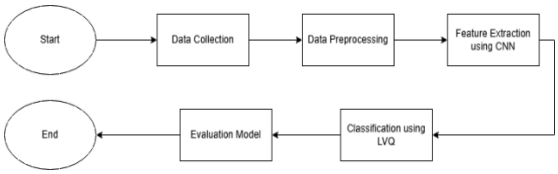


Figure 1. Research Flow

The hardware and software specifications used in this study are shown in Tables 1 and 2.

Hardware	Specification
Computer Type	11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz, 2419 MHz, 4 Core(s), 8 Logical Processor(s)
RAM	8.00 GB
Hard Disk	477 GB

Table 1. Hardware specifications

Software	Specification
Operating System	Windows 11 Home Single Language
Application	MATLAB R2021a

Table 2. Software specifications

Data Collection

The dataset used in this study consists of chili seed images. The chili seeds used are divided into three types, namely gendot, raksasa, and rawit. Each type of chili seed consists of 300 images, so the total dataset used amounts to 900 images.

Image capture was carried out using a Redmi 13X smartphone with a 108 MP camera resolution. Image capture was assisted by room lighting, with a capture distance from the object of approximately ±10 cm. A white HVS paper background was used consistently for all images to minimize noise in the edge feature extraction process.

The acquisition process did not involve data augmentation, so the variation in the dataset came entirely from the natural variation in the shape and size of the chili seeds. Image quality control was carried out manually by checking the captured images and repeating the image capture if the image was blurry, had uneven lighting, or had an unclear object position.

Figure 2. Gendot Chili Seed

Figure 3. Raksasa Chili Seed

Figure 4. Rawit Chili Seed

Data Preprocessing

After the data collection process of 900 images from 3 types of chili seeds was completed, each image went through a resizing stage from a size of 3000 × 3000 pixels to 1024 × 1024 pixels. The choice of a 1024 × 1024 pixel image size was based on a consideration of the balance between the quality of the resulting edge features and the required computational load. An image size that is too large would increase the computational load in the convolution and pooling process without providing a significant increase in edge information, whereas an image size that is too small risks losing the morphological detail of the chili seed that serves as the distinguishing feature between varieties. A size of 1024 × 1024 pixels was chosen because, after passing through three convolution-pooling layers with a stride of (2,2), the image produces a final feature map of 16 × 16 pixels that still retains representative edge information, while keeping the computation time light on

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

hardware without a dedicated GPU (Supiyandi et al., 2024)

Feature Extraction Using CNN

Based on (Zhao et al., 2024), Convolutional Neural Network (CNN) is one of the deep learning models most widely used in the field of computer vision. CNN is designed as a deep artificial neural network inspired by the workings of the biological visual system, particularly the receptive field mechanism in the brain. In this study, CNN serves as the feature extractor (Sardogan et al., 2018). The advantage of CNN lies in its ability to automatically recognize and extract spatial patterns through a layered convolution process (Akbar & Wiliani, 2025).

However, it must be emphasized that the architecture used in this study differs from a conventional trainable CNN, in which the convolution kernel values are updated automatically through a backpropagation process. In this study, all convolution kernels use predefined Sobel filters that remain fixed (frozen) throughout the feature extraction process (Linse et al., 2023). The implementation was carried out using MATLAB without relying on any built-in deep learning library, where the convolution operation, Sobel gradient magnitude, and max pooling are computed using pixel-by-pixel loops so that the computation mechanism can be explicitly observed and validated at every stage. In this study, two main types of layers in the CNN architecture were adopted from the study by (Zhao et al., 2024), which are then repeated three times (depth-3) in the feature extraction process, with the stages at each layer as follows:

1. **Convolution Layer**

Convolution is the operation performed in the first layer of the CNN. The convolution mechanism works by sliding the kernel over the input image; each kernel value is then multiplied by the corresponding pixel value and summed to produce a feature map. At each convolution layer, a Sobel filter (predefined) is used to detect edges horizontally and vertically, as described in (Supiyandi et al., 2024). The Sobel filter was chosen because it performs well at highlighting object shape boundaries and contours (Apriansyah Y et al., 2025), which is important information for distinguishing chili types based on the morphology of their seeds.

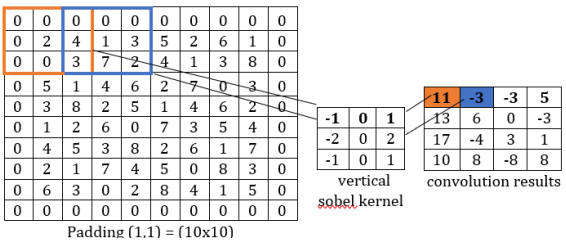
The convolution operation generally causes the feature map size to become smaller than the original image size, because the kernel can only be shifted to positions where all of its elements remain within the image boundary. Therefore, zero padding of size (1,1) is applied to the edges of the image before the convolution process is carried out, adjusted to the 3×3 Sobel kernel size used, so that the edge pixels of the image remain covered by the kernel and the contour information at the edges is not lost during the convolution process.

Next, the convolution process is carried out using a stride of (2,2), meaning the kernel shifts by 2 pixels at each step. This choice of stride 2 is based on the consideration that a stride value that is too small (e.g., 1) produces a large feature matrix size that is redundant between adjacent pixels, whereas a stride that is too large risks losing important edge detail. Stride 2 was chosen as a middle ground that still maintains edge continuity thanks to a one-pixel overlap, while significantly reducing the spatial dimensions of the image so that the resulting number of features is more compact for processing at the LVQ classification stage.

The larger the stride value with the same kernel size, the smaller the resulting convolution matrix size(Affifah et al., 2022). This has an effect on computation time, given that all convolutions are computed manually through loops without a Deep Learning Toolbox or GPU. With stride 2, the number of pixel positions processed in each dimension is reduced by half compared to stride 1, so the total number of positions computed per layer is reduced to a quarter a saving that applies across all three convolution-pooling layers (depth-3) cumulatively speeding up the feature extraction process without sacrificing the edge information that forms the basis of chili seed classification.

2	4	1	3	5	2	6	1
0	3	7	2	4	1	3	8
5	1	4	6	2	7	0	3
3	8	2	5	1	4	6	2
1	2	6	0	7	3	5	4
4	5	3	8	2	6	1	7
2	1	7	4	5	0	8	3
6	3	0	2	8	4	1	5

Figure 5. 8x8 Matrix



P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

Figure 6. Vertical Sobel Convolution

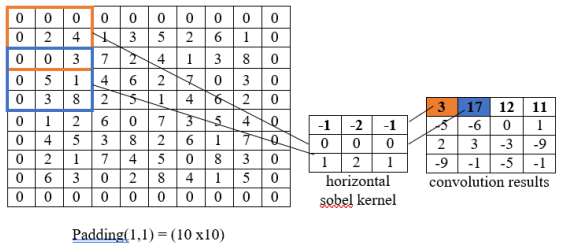


Figure 7. Horizontal Sobel Convolution

The vertical (Sv) and horizontal (Sh) Sobel gradients are then combined using the gradient magnitude equation (Supiyandi et al., 2024):

$$G = \sqrt{(Sv^2 + Sh^2)} \qquad (1)$$

The resulting gradient magnitude becomes the output of each convolution layer.

2. Pooling Layer

After the convolution layer process, the image enters the pooling layer. The pooling layer functions to compress data and parameters, reduce the dimensions of the feature map, and minimize overfitting. There are generally two types of pooling, namely max pooling and average pooling. This study uses max pooling with a stride of (2,2). The max pooling mechanism works by taking the largest activation value in a region as the representation of that region.

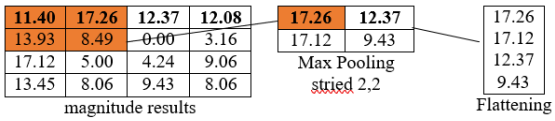


Figure 8. Max Pooling

The CNN architecture in this study uses three convolution and pooling layers. This choice is based on the consideration that too few layers leave local detail that is less relevant for distinguishing the overall seed shape, whereas too many layers risk losing important edge information and produce a feature dimension that is disproportionate to the amount of training data. This consideration refers to the study by (Pelletier et al., 2019), which shows that a depth of two to three convolution layers gives the best accuracy with the most stable variation in similar CNN architectures. With three convolution layers, a 1024 × 1024 pixel chili seed image produces a 16 × 16 feature map, which is then *flattened* into a 256-dimensional feature vector before being classified using LVQ.

Classification Using LVQ

Based on (Pasaribu et al., 2025), Learning Vector Quantization (LVQ) is an artificial neural network method that combines competitive learning and supervised learning. This method applies a supervised learning process at the competitive layer to classify input vectors into specific classes.

In this study, LVQ is used to replace the fully connected layer in classifying the edge features extracted by the Sobel filter that have been processed using CNN(Sardogan et al., 2018),(E. Y. Hidayat & Radiffananda, 2019).

Unlike SVM, which requires kernel selection and relatively complex parameter tuning for high-dimensional data(Tsai & Chang, 2023), and Random Forest, which requires many decision trees and thus increases computational complexity(Bilolikar et al., 2025), LVQ offers a simple and easily interpretable prototype-based classification mechanism, since each prototype directly represents the class characteristics in the feature space (Arif & Pramana, 2022). LVQ also aligns with this study's philosophy of prioritizing a lightweight and transparent approach, since its training process does not require complex non-linear optimization as in SVM, nor an ensemble process as in Random Forest. Compared to KNN, which is a lazy learning method that stores the entire training dataset for the classification process, LVQ only stores a small number of trained prototypes, making it more efficient in terms of memory and computation time during testing (Pasaribu et al., 2025). The main stages in classification using LVQ in this study are as follows:

1. Data Normalization

Before the LVQ training process is carried out, the feature vector is normalized using the Z-score method (mean-std normalization) so that each feature dimension has an equivalent scale. This normalization is important because LVQ uses Euclidean distance as a similarity measure. Without normalization, features with a large value range would dominate the distance calculation (Mansur et al., 2025).

2. Determining Prototypes

The LVQ model is initialized using 5 prototypes for each class, so there are 15 prototypes in total. Prototypes are selected randomly from the training data of each class. The use of several prototypes per class allows LVQ to better represent intra-class variation, so that the

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

decision boundary between classes can be formed more effectively (Sardogan et al., 2018)(Pasaribu et al., 2025).

3. LVQ Calculation

In this study, the initial weights are selected randomly from the training data of each class. The LVQ training process is carried out by calculating the Euclidean distance between the input data and all prototypes. The prototype with the smallest distance is selected as the winning neuron. The Euclidean distance calculation is shown in the following equation (Arif & Pramana, 2022):

After the winning neuron is obtained, the weights are updated based on class agreement. If the prototype label matches the input data label, the weight is moved closer to the input data using the equation (Arif & Pramana, 2022) :

$$Wbaru = Wlama + \alpha (X - Wlama) \quad (3)$$

Conversely, if the prototype label does not match the input data label, the weight is moved away from the input data using the equation (Arif & Pramana, 2022):

$$Wbaru = Wlama - \alpha (X - Wlama) \quad (4)$$

4. Determining the Learning Rate

The success of the LVQ model depends heavily on the configuration of several parameters, namely the Learning Rate (α), Epoch (Iteration), and Minimum Alpha (Min) (Aziz et al., 2023). The training process is carried out using a 5-fold cross-validation scheme with a maximum of 30 epochs per fold. This number of epochs is determined based on the learning rate decay mechanism, with an initial alpha parameter of 0.001, a decay rate (DecAlfa) of 0.04, and a lower alpha bound (MinAlfa) of 0.00001; the alpha value continues to decrease gradually at each epoch until it approaches this lower bound, so that the prototype weight updates become progressively smaller and the training process naturally stops making significant changes by the end of the 30th epoch. The choice of parameters in this study is based on the results of an initial preliminary trial using several combinations of values, given that the success of the LVQ model depends heavily on the configuration of parameters such as learning rate, epoch, and number of prototypes (Aziz et al., 2023)

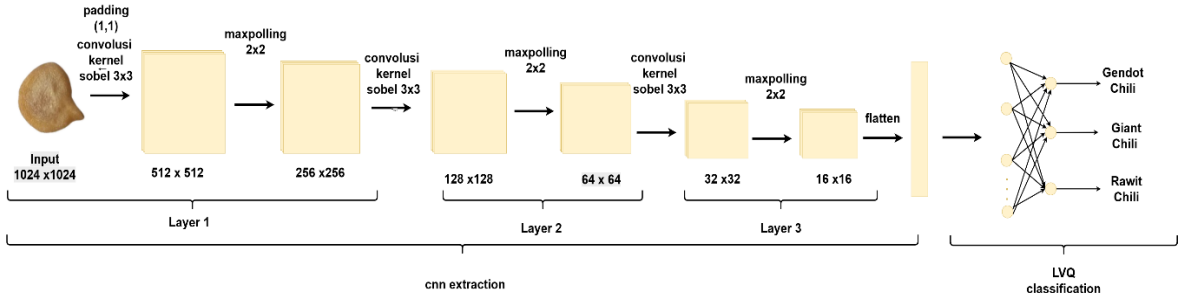


Figure 9. CNN and LVQ Architecture

After the training process is complete, the model is tested using the testing data to determine its classification ability on data it has not previously learned. Performance evaluation is carried out using a confusion matrix along with the accuracy, precision, recall, and F1-score metrics.

Model Evaluation

The 5-fold cross-validation scheme applied at the data-splitting stage is used to evaluate the consistency and reliability of the model's performance more thoroughly(F. M. Hidayat et al., 2026). In each iteration, four folds are used as training data and the remaining fold is used as test data, and this process is repeated five times until every fold has been used as test data once. This approach was chosen to reduce the potential bias that can arise from using a single data-splitting scheme, so that the resulting evaluation better represents the model's generalization ability.

Model performance is evaluated using several metrics, namely accuracy, precision, recall, and F1-score.

Accuracy is used to measure the model's overall correctness in classifying data(Ripa'i et al., 2022).

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (5)$$

Precision is used to measure the model's accuracy in predicting a particular class (Ripa'i et al., 2022).

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (6)$$

Recall is used to determine the model's ability to detect data according to its true class (Ripa'i et al., 2022).

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (7)$$

F1-score is a combination of precision and recall, in order to give a comprehensive picture of the model's performance(Ripa'i et al., 2022) using the formula:

$$\text{F1-Score} = \frac{(2 \times \text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (8)$$

Notes:
TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

RESULTS AND DISCUSSION

Data Preprocessing Results

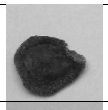
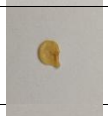
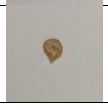
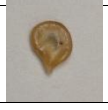
Chili Seed Type	3000 x 3000	1024 x 1024	Grayscale Image
Gendot			
Raksasa			
Rawit			

Table 3. Preprocessing Results

From the preprocessing results, the grayscale chili seed images with a size of 1024 x 1024 pixels are then used as input for the edge feature extraction process using the Sobel-based CNN.

CNN Extraction Results

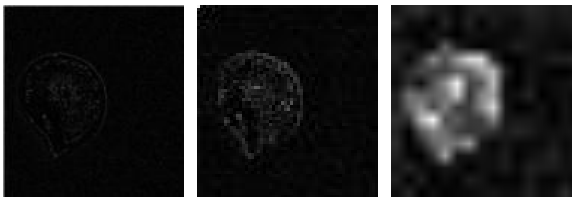
The extraction results are obtained from the convolution and max pooling process from the first to the third layer, producing edge images of size 16 × 16.

Layer 1 Layer 2 Layer 3

Figure 10. Extraction Results of Gendot Chili Seed

Layer 1 Layer 2 Layer 3

Figure 11. Extraction Results of Raksasa Chili Seed



Layer 1 Layer 2 Layer 3

Figure 12. Extraction Results of Rawit Chili Seed

The figures above show the chili seed images after passing through three layers of Sobel convolution and pooling, with the feature map size gradually shrinking from 256×256 to 16×16 pixels. In the first layer, the Sobel operation results still highlight local edges around the seed boundary and background, so surface texture detail is still dominant. In the second layer, the pooling process begins to merge these local edges into patterns representing parts of the seed's structure, such as the base, tip, and lateral sides. In the third layer, the 16×16 feature map already represents the overall global shape of the seed, such as the length-to-width proportion and silhouette curvature, which are the main distinguishing features between varieties.

This pattern is consistent with the characteristics of the dataset, since the three chili seed varieties in this study are distinguished mainly by their global shape and proportions, rather than by surface texture detail. The Gendot seed has a fairly round silhouette that is quite different from Raksasa and Rawit, so it can already be separated well at the third layer. In contrast, the Raksasa and Rawit seeds produce relatively similar Sobel edge contrast across all layers, so the two often produce neighboring patterns in the feature space one of the

main causes of classification errors between these two varieties.

The final 16×16 feature map size was chosen based on three considerations: first, this size still preserves the global shape pattern of the seed without losing the ability to distinguish between varieties; second, the dimension of the flattened feature vector (256 elements) is proportionate to the amount of training data per class, so the risk of overfitting in LVQ can be reduced, given that LVQ works based on distance between prototypes and performs less well when the feature dimension is too high relative to the amount of data; and third, three convolution-pooling layers are consistent with similar CNN architectures, which generally use a depth of two to three layers to maintain a balance between feature representation and model stability.

The following table shows an example of the edge feature extraction results for each type of chili seed using the Sobel-based CNN method with a depth of 3 layers. The extraction process produces a feature matrix of size 256 × 900 for the entire dataset, with 256 representing the number of features per image and 900 representing the total number of images across the three chili seed types (300 images per type). These extracted features are then used as input for the classification process using Learning Vector Quantization (LVQ).

Label	Feature 1	Feature 2	…	Feature256
Gendot1.jpg	180.4057 45	121.0933693	…	60.68076992
Gendot2.jpg	146.1397885	103.7901458	…	180.1280847
Gendot3.jpg	184.5094216	210.5974075	…	46.77441304
Gendot4.jpg	229.2092304	461.6868163	…	231.5890613
Gendot5.jpg	251.0207067	135.9783871	…	51.69323127
…	…	…	…	…
Gendot300.jpg	234.527874	168.8270539	…	72.90801693
Raksasa1.jpg	312.0337117	244.5207646	…	126.0747771
Raksasa2.jpg	204.1716435	241.1691633	…	122.9448085
Raksasa3.jpg	358.4553958	306.9864911	…	193.5857943
Raksasa4.jpg	242.2431662	205.274763	…	244.8540301
Raksasa5.jpg	218.3155393	129.3529318	…	77.95279505
…	…	…	…	…
Raksasa300.jpg	257.0174435	224.8993563	…	275.6774887
Rawit1.jpg	275.7772497	233.8811041	…	128.4065118
Rawit2.jpg	339.915407	186.1481416	…	112.5602377
Rawit3.jpg	253.4741635	304.1049115	…	113.6888868
Rawit4.jpg	224.623676	293.0894716	…	175.0322611
Rawit5.jpg	258.4243442	194.9182063	…	401.2199541
…	…	…	…	…
Rawit300.jpg	333.1395164	157.0048588	…	84.37479105

Table 4. Feature Extraction Results

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

LVQ Classification Results

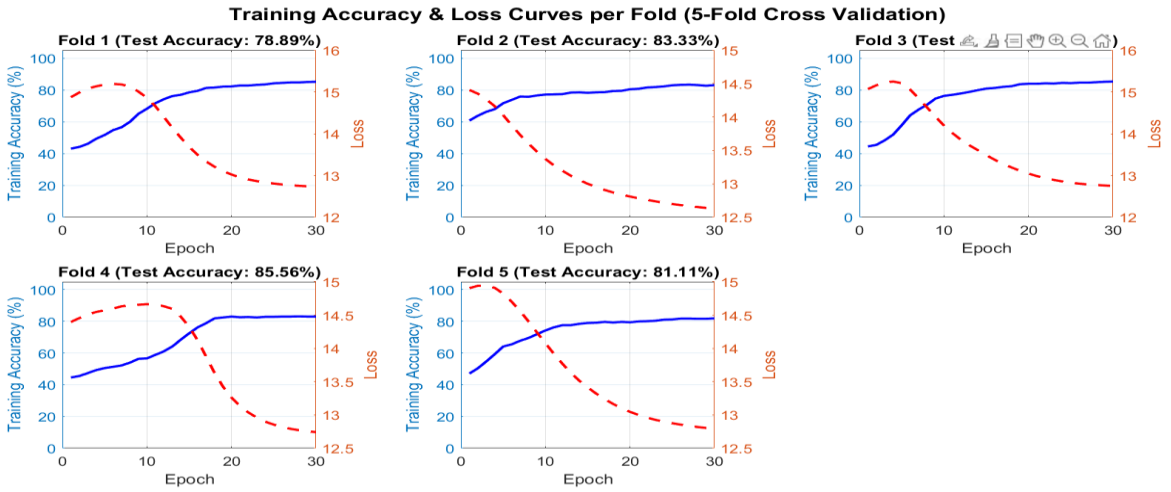


Figure 13. Training Accuracy and Loss Curves for 5-Fold CV

Based on Figure 13, the test results for the five folds show accuracies of 78.89% (fold 1), 83.33% (fold 2), 75.00% (fold 3), 85.56% (fold 4), and 81.11% (fold 5), respectively, with a mean accuracy of 80.78% and a standard deviation of 4.07%. This relatively small standard deviation indicates that the model's performance is fairly stable and does not depend too heavily on any particular data split, although fold 3 shows a slightly lower accuracy compared to the other folds.

Based on the training history across the five folds, the average training accuracy increased sharply from 47.97% at the first epoch to 70.50% at epoch 10, then continued to rise to 81.81% at epoch 20. The accuracy gain began to slow after epoch 20, with the difference between epoch 20 and epoch 30 amounting to only 1.88% (from 81.81% to 83.69%), accompanied by a loss decrease that likewise leveled off from 13.03 to 12.73. This pattern of leveling accuracy growth and loss reduction between epochs 20 and 30 occurred consistently across all folds, so a total of 30 epochs can be considered sufficient to reach a stable condition consistent with the alpha-decay scheme used, without requiring additional epochs that could risk overfitting on the training data.

Unlike a single-training scheme that relies solely on training accuracy as a performance indicator, the 5-fold cross-validation approach in this study includes a testing stage on test data that is separate from the training data in each fold, so that the model's generalization ability can be evaluated directly rather than merely assumed from the training curve alone. The test results across the five folds show a mean test accuracy of 80.78% with a standard deviation of 4.07%, which is relatively consistent with the final training accuracy of 83.69%, so the small difference between the training and test accuracy indicates that the model does not experience significant overfitting on the training data.

Model Evaluation

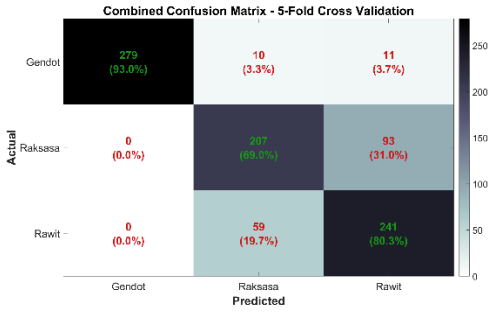


Figure 14. Confusion Matrix

Figure 14 shows the combined confusion matrix from the classification results of the five folds in the 5-fold cross-validation scheme, with a total of 900 test data points (300 data points per class). Based on this matrix, the Gendot class achieved the highest correct classification rate, namely 279 out of 300 data points (93.0%), with classification errors distributed in small numbers to the Raksasa (10 data points, 3.3%) and Rawit (11 data points, 3.7%) classes. Not a single data point from the Raksasa or Rawit classes was misclassified

P-ISSN: 2656-1743 |E-ISSN: 2656-1735
DOI: https://doi.org/10.34288/jri.v8i4.572
Accredited rank 4 (SINTA 4), excerpts from the decision of the DITJEN DIKTIRISTEK No. 230/E/KPT/2023

as Gendot, which shows that the model is able to consistently distinguish the Gendot class from the other two classes.

In the Raksasa class, out of 300 data points, 207 data points (69.0%) were classified correctly, while 93 data points (31.0%) were misclassified as Rawit. A similar pattern occurred in the Rawit class, where 241 data points (80.3%) were classified correctly, but 59 data points (19.7%) were misclassified as Raksasa. The classification errors in both of these classes occurred entirely between Raksasa and Rawit, with no errors involving the Gendot class, so it can be concluded that the main source of the model's error is concentrated in the similarity of characteristics between the Raksasa and Rawit classes.

Class	Precision	Recall	F1-Score
Gendot	1,00	0,93	0,96
Raksasa	0,75	0,69	0,72
Rawit	0,71	0,80	0,75

Table 5. Average Model Evaluation Results on 5-Fold Cross Validation

The Gendot class achieved a precision of 1.00, meaning that all data predicted as Gendot by the model genuinely originated from the Gendot class, with no misprediction from other classes. A recall value of 0.93 for this class shows that the model successfully recognized 93% of all actual Gendot data, so that the F1-score, which is the harmonic mean of precision and recall, reached 0.96.

The Raksasa class achieved a precision of 0.75 and a recall of 0.69, with an F1-score of 0.72. The higher precision compared to recall for this class shows that although most Raksasa predictions were accurate, the model still often failed to recognize actual Raksasa data (low recall), consistent with the confusion matrix finding that 31.0% of Raksasa data was misclassified as Rawit. In contrast, the Rawit class achieved a precision of 0.71 and a recall of 0.80, with an F1-score of 0.75, showing the opposite pattern the model more often correctly recognized Rawit data, but also more often mistakenly predicted data from other classes (particularly Raksasa) as Rawit.

Overall, the relatively lower F1-scores of the Raksasa and Rawit classes compared to the Gendot class confirm that these two classes have a high degree of feature similarity, making them the main source of the decrease in the model's overall classification performance.

Discussion

The system's main difficulty lies in separating the Raksasa and Rawit classes. These two chili types have similar morphology in the edge patterns and contours detected using the Sobel filter. Using a single type of filter across all layers limits the diversity of feature representation compared to a trainable CNN that can learn filters adaptively. Nevertheless, the non-trainable CNN-Sobel approach has the advantage of lighter computation, not requiring a GPU, and a more transparent process because it does not use backpropagation.

Compared to the study by Pebrianto and Haryanto (2023), which classified three chili seed varieties using a Python framework-based CNN with an accuracy of around 90%, the result of this study (80.78%) is slightly lower. However, the study by Pebrianto and Haryanto used flatbed scanner images that produced more consistent image quality and sharper edge contrast, whereas this study uses images from a smartphone camera with natural lighting variation, so the resulting image characteristics are more varied and more challenging for the Sobel-based feature extraction process. Compared to the study by Sardogan et al. (2018), which combined CNN and LVQ for tomato leaf disease classification, this study shows that the combination of CNN-Sobel and LVQ can also be applied to smaller-scale objects such as chili seeds, although the challenge of separating classes with high visual similarity (Raksasa and Rawit) remains a common obstacle for edge-feature-based approaches. Meanwhile, the study by Siswipraptini et al. (2023), which applied LVQ for chili leaf disease classification, also showed that LVQ can be used effectively as a final classifier for chili plant images, consistent with the results of this study, which show good LVQ performance on classes with fairly distinctive visual features, such as the Gendot class.

CONCLUSIONS AND SUGGESTIONS

Conclusion

This study successfully built a chili seed image classification system using a hybrid method combining a Sobel filter-based CNN (depth-3) and Learning Vector Quantization (LVQ). Model evaluation was carried out using a 5-fold cross-validation scheme on a dataset consisting of 900 images from three chili varieties, namely gendot, raksasa, and rawit, and produced a mean testing accuracy of 80.78% with a standard deviation of 4.07%. Based on the average recall value per class, the Gendot class achieved a recognition rate of 93%, while the Raksasa and Rawit classes achieved 69% and 80%, respectively, with classification errors

dominated by confusion between the Raksasa and Rawit classes due to the similarity in morphological characteristics between the two varieties. The relatively small standard deviation in accuracy across folds shows that the model's performance is fairly stable and does not depend on any single particular data-splitting scheme. This hybrid approach can be used for chili seed classification without requiring special hardware or a pre-trained CNN model, making it potentially applicable to systems with limited computational resources.

Suggestion

Based on the limitations found, the following recommendations are proposed for future research. First, combining the Sobel filter with texture-based filters such as Gabor or Local Binary Pattern to help distinguish the Raksasa and Rawit classes, which in this study still show a fairly large misclassification rate (31.0% and 19.7%) due to morphological similarity. Second, enlarging and diversifying the dataset, particularly for the Raksasa and Rawit classes, as well as varying the lighting, background, and ripeness level of the chili to reduce overlap between classes. Third, because this study was intentionally focused on evaluating the feasibility of the non-trainable CNN-Sobel + LVQ pipeline, rather than conducting a comparative study between classifiers , comparing LVQ with other classifiers such as SVM or Random Forest on the same features, to determine whether the errors originate from the feature extraction or the classifier, a comparative study is recommended. Fourth, comparing this fixed-filter approach with a pretrained CNN such as MobileNet or ResNet, to see whether automatically learned features separate Raksasa and Rawit better. Fifth, exploring feature selection or dimensionality reduction to reduce computational load without decreasing accuracy, so that this approach remains suitable for devices with limited resources.

REFERENCES

Affifah, D. D., Permanasari, Y., Matematika, R. P., Matematika, F., Ilmu, D., & Alam, P. (2022). Bandung Conference Series: Mathematics Teknik Konvolusi pada Deep Learning untuk Image Processing. *Bandung Conference Series: Mathematics*, *2*(2), 103–112. https://doi.org/10.29313/bcsm.v2i2.4527

Akbar, F., & Wiliani, N. (2025). Perbandingan Kinerja ANN dan CNN dalam Tugas. *Tekomatika*, *18*(1), 22–27.

Alhanannasir, Muchsiri, M., Yani, A. V., Amelia, K. R., & Sebayang, N. S. (2025). Pengaruh Formulasi Cabai Rawit (Capsicum Frutescens Linn) Terhadap Cuko Pempek The Effect of Cayyey Chill Formulation (Capsicum frutescens Linn) Against Cuko Pempek. *Jurnal Penelitian Pertanian Terapan*, *25*(1), 112–119.

Apriansyah, Y., Khairi, N., Siregar, Ha. H. S., & Lubis, A. H. (2025). Implementation of Edge Detection Using the Sobel Operator on Papaya Leaf Images. *Jurnal Ilmiah Informatika Dan Komputer (Informatech)*, *2*(2), 164–168.

Arif, M. F., & Pramana, A. A. (2022). Implementasi Metode Learning Vector Quantization (Lvq) Pada Pengenalan Bahasa Isyarat Yang Mengandung Kata Kerja. *JAMI: Jurnal Ahli Muda Indonesia*, *3*(1), 1–8. https://doi.org/10.46510/jami.v3i1.40

Aziz, A., Insani, F., Jasril, J., & Syafria, F. (2023). Implementasi Metode Learning Vector Quantization (LVQ) Untuk Klasifikasi Keluarga Beresiko Stunting. *Building of Informatics, Technology and Science (BITS)*, *5*(1), 12–21. https://doi.org/10.47065/bits.v5i1.3478

Bilolikar, D. K., More, A., Gong, A., & Janssen, J. (2025). How to out-perform default random forest regression: choosing hyperparameters for applications in large-sample hydrology. *Hydrology Research*, *57*(1), 61–77. https://doi.org/10.2166/nh.2025.075

Hidayat, E. Y., & Radiffananda, M. F. (2019). Pengenalan Tanda Tangan Menggunakan Learning Vector Quantization dan Ekstraksi Fitur Local Binary Pattern. *CogITo Smart Journal*, *5*(2), 123–136. https://doi.org/10.31154/cogito.v5i2.180.123-136

Hidayat, F. M., Crysdian, C., & Lestari, T. M. (2026). *Optimasi Deteksi Retakan Jalan Menggunakan Filter Sobel dan Klasifikasi Gaussian Naïve Bayes*. *11*(2), 155–168.

Kementerian Pertanian, I. (2024). *Outlook Cabai 2024*. Diakses Tanggal 03 Juni 2026 dari https://satudata.pertanian.go.id/assets/docs/publikasi/OUTLOOK_CABAI_2024_.pdf

Khoerunisa, R., Jheniar, R., Siti, N., Ayuanindya, D. G., Surtikanti, H. K., Priyandoko, D., Alam, P., Indonesia, U. P., & Pendidikan, U. (2024). Kandungan senyawa capsaicin dalam cabai (Capsicum Annuum L) sebagai anti hama pada sayuran: Kajian pustaka. *Holistic: Journal of Tropical Agriculture Sciences*, *1*(2), 138–148.

Linse, C., Barth, E., & Martinetz, T. (2023). Convolutional Neural Networks Do Work with

Pre-Defined Filters. *Proceedings of the International Joint Conference on Neural Networks*, *2023-June*, 1–13. https://doi.org/10.1109/IJCNN54540.2023.10191449

Mansur, Umar, N., & Zuhriyah, S. (2025). Implementasi Algoritma Learning Vector Quantization untuk Deteksi Dini Penyakit Mata. *Jurnal Sains Dan Informatika*, *11*(1), 1–10. https://doi.org/10.34128/jsi.v11i1.907

Nurbuana, B., Nasrinatun, E., Arifin, M. P., Ayu, M., & Widya, D. (2024). Klasifikasi dan Pengenalan Pola Penyakit Cabai Dengan Metode CNN (Convolution Neural Network). *Prosiding seminar nasional teknologi dan sains*, *3*, 125–132.

Pasaribu, R. F. H., Zarlis, M., & Nababan, E. B. (2025). Performance Level Analysis On Learning Vector Quantization And Cohonen Algorithms. *Sinkron*, *9*(1), 267–282. https://doi.org/10.33395/sinkron.v9i1.14313

Pebrianto, R., & Haryanto, T. (2023). *IJCIT (Indonesian Journal on Computer and Information Technology) Optimasi Sistem Klasifikasi Biji Tanaman Cabai Menggunakan CNN : Pendekatan Inovatif dalam Agribisnis*. *8*(2), 121–129.

Pelletier, C., Webb, G. I., & Petitjean, F. (2019). Temporal convolutional neural network for the classification of satellite image time series. *Remote Sensing*, *11*(5), 1–25. https://doi.org/10.3390/rs11050523

Perlindungan, I., & Risnawati. (2020). Pengenalan Tanaman Cabai Dengan Teknik Klasifikasi Menggunakan Metode CNN. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)*, 15–22.

Ripa'i, A., Santoso, F., & Lazim, F. (2022). Deteksi Berita Hoax dengan Perbandingan Website Menggunakan Pendekatan Deep. *G-Tech : Jurnal Teknologi Terapan*, *6*(2), 295–305.

Safitri, R. (2024). Karakterisasi Morfologi dan Anatomi Beberapa Kultivar Cabai (capsicum annuum l .) Dalam Menanggapi Cekaman Kekeringan. *Integrative Perspectives of Social and Science Journal (IPSSJ) Ipssj.Com*, *1*(1), 156–164.

Sardogan, M., Tuncer, A., & Ozen, Y. (2018). Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm. *Computers and Electronics in Agriculture*, 382–385. https://doi.org/10.1109/UBMK.2018.8566635

Sello, M., & Lekatsa, T. (2023). Chili Pepper as a Functional Food: Relevance to Lesotho. *International Journal of Food Sciences*, *6*(1), 15–25. https://doi.org/10.47604/ijf.2085

Siswipraptini, P. C., Haris, A., & Sari, W. N. (2023). *Klasifikasi Citra Penyakit Daun Cabai Menggunakan Algoritma Learning Vector Quantization*. *16*(2), 119–125. https://doi.org/10.30998/faktorexacta.v16i2.15900

Supiyandi, Panggabean, T., Ramadhan, N., Dewi, S. D., & Yusra, S. (2024). Deteksi Tepi Sederhana Pada Citra Menggunakan Operator Sobel. *Repeater : Publikasi Teknik Informatika Dan Jaringan*, *2*(3), 43–56. https://doi.org/10.62951/repeater.v2i3.90

Tosi, R. B., Mbura, H. D., & Kaesmetan, Y. R. (2024). Implementasi CNN Dalam Mengidentifikasi Kematangan Cabai Berdasarkan Warna. *Indonesian Journal of Education And Computer Science*, *2*(1), 34–42. https://doi.org/10.60076/indotech.v2i1.385

Tsai, C. A., & Chang, Y. J. (2023). Efficient Selection of Gaussian Kernel SVM Parameters for Imbalanced Data. *Genes*, *14*(3). https://doi.org/10.3390/genes14030583

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. In *Artificial Intelligence Review* (Vol. 57, Issue 4). Springer Netherlands. https://doi.org/10.1007/s10462-024-10721-6