Image classification of different clove (Syzygium aromaticum) quality using deep learning method with convolutional neural network algorithm

I Y Prayogi*, Sandra and Y Hendrawan
Department of Agricultural Engineering, Agricultural Technology Faculty, Universitas Brawijaya, Veteran St., Malang 65145, Indonesia

Corresponding author: istifaryogi_p@student.ub.ac.id

Abstract. The objective of this study is to classify the quality of dried clove flowers using deep learning method with Convolutional Neural Network (CNN) algorithm, and also to perform the sensitivity analysis of CNN hyperparameters to obtain best model for clove quality classification process. The quality of clove as raw material in this study was determined according to SNI 3392-1994 by PT. Perkebunan Nusantara XII Pancusari Plantation, Malang, East Java, Indonesia. In total, 1,600 images of dried clove flower were divided into 4 qualities. Each clove quality has 225 training data, 75 validation data, and 100 test data. The first step of this study is to build CNN model architecture as first model. The result of that model gives 65.25% reading accuracy. The second step is to analyze CNN sensitivity or CNN hyperparameter on the first model. The best value of CNN hyperparameter in each step then to be used in the next stage. Finally, after CNN hyperparameter carried out the reading accuracy of the test data is improved to 87.75%.

1. Introduction
In clove plant, clove flower is part that commonly used and dried according to its quality standards. The picked clove flowers normally dried by the farmers, until its texture become brownish. Following that process, then farmer sorting the dried clove flowers based on the quality using visual assessment according to its size and color [1].

Indonesia is ranked 5th in the clove export sector to Germany. However, from 2014 to 2015, there was a decline in the exports value of clove [2]. The standard quality for cloves has been set in SNI 01-3392-1994 and for the export market, ISO 2254-2004 is the standard for quality assessment.

To classify the dry cloves quality, image analysis can be used to make the work easier. Previously, the classification of dried clove flower quality based on size and color using the GLCM (Gray Level Co-Occurrence Matrix) method with 32 samples, give 92.5% accuracy [3]. Another study reported the use of HSV (Hue Saturation Value) method to classify the poor-quality of clove using 40 images as sample [4]. According to the previous studies mentioned, then we can assume that the clove quality classification can be done by distinguish its shape and color.

One of image analysis technique is deep learning with the CNN (Convolutional Neural Network) algorithm. The use of machine learning aimed to solve complex and large problems in short time by using CNN with feedforward and backpropagation stages [5]. The advantage of deep learning method are the data processing is automatic and more effective when compared to the Partial Least Square
(PLS), Artificial Neural Network (ANN), Support Vector Machine (SVM) also K-Nearest Neighbor (KNN) modelling methods [6]. The CNN algorithm can be used in various scientific fields, such as coffee quality classification in agriculture [7,8] and the classification of malaria virus blood infection in health sciences [5].

The purpose of this study is to build a deep learning model using the CNN algorithm by performing sensitivity analysis stages to get the best model value for analyzing the clove quality.

2. Materials and methods
In this section, we present the parts of the Convolutional Neural Network (CNN) architecture and the steps in performing a sensitivity analysis or Hyperparameter CNN to get the best model CNN.

2.1. Data collection
The raw material of this study is clove image with different quality which taken from PTPN12, Malang. The amount of the data is summarized in Table 1. In total this study uses 1,600 dataset, 900 data training, 300 data validation, and 400 data testing. The image size is 300×300 pixels, and the quality of cloves were divided into 4 classifications.

Table 1. Dataset of clove quality classifications

| Classification | Data training | Data validation | Data testing |
|----------------|---------------|-----------------|--------------|
| Quality 1      | 225           | 75              | 100          |
| Quality 2      | 225           | 75              | 100          |
| Quality 3      | 225           | 75              | 100          |
| Quality 4      | 225           | 75              | 100          |
| Subtotal       | 900           | 300             | 400          |
| Total          | 1,600         |                 |              |

2.2. System requirements
The test was carried out on a computer with specification using Windows 10, processor Core i3 4150 CPU 3.50 GHz, specifications of GPU HD graphics 4,400 and Double Data Rate 3 10 Gb.

2.3. Convolutional Neural Network (CNN)
CNN is part of a deep learning technique that is inspired by the visual mechanism of living things, where CNN is widely used and very effective in analyzing an image [9]. The stages of CNN are convolutional layer, pooling layer, fully connected layer as we can see in Figure 1. For CNN architecture, the convolutional layer stages are built from the number of layers, kernel size, strides, padding and ReLu activation that must be regulated and optimized. Then the output of the convolutional layer will be subsampled by pooling with the aim of selecting image information that represents the input. The last stage is a fully connected network that connects many nerves to the output of the image analysis [6].

Figure 1. Architecture of CNN

2.3.1. Convolutional layers. Convolution layer is a process used to extract an image from the input into several parts or layers. As we can see in Figure 2, when the convolution process there are several
parts, like the number of filters, kernel size, strides, and padding. The function of the kernel itself is a sub window that will extract an image in a sliding window. Stride and padding used to control kernel movement and improve output accuracy [10].

![Convolutional Layer](image)

**Figure 2. Convolutional Layer**

In this study, the input image used is 128×128 pixels with RGB. While the feature maps used were 2 feature maps (32 and 64). In the convolution process, there is a kernel, stride, and padding to help the convolution process. Kernel size 3×3, Stride 1×1, and use padding in the convolution process.

2.3.2. Pooling layer. Represent the image (input) or from the previous layer to be a lower revolution through sub-sampling. There are two types of pooling that can be used, namely MaxPooling and AveragePooling. MaxPooling is taking the maximum value from each grid to compose the reduced image. Average Pooling is taking the average value of an image [11]. According to the previous report Scherer et al. [12], the use of MaxPooling is more widely used by users of the deep learning method because it only takes the largest value of the matrix, so the accuracy result is higher. Figure 3 is example for MaxPooling process.

Pooling settings in this research are made with a size of 2×2, where the type of pooling used is MaxPooling. For example, in Figure 3 the result of the convolution process with a size of 64×64, after going through the pooling process the image size will change to a new size of 32×32, because the pooling size used is 2×2.

![Pooling layer](image)

**Figure 3. Pooling layer**

![Fully connected layer](image)

**Figure 4. Fully connected layer**

2.3.3. Fully connected layer. This is the last process of the CNN modelling stage. At this stage, it aims to connect the results from the previous stage in the form of a multidimensional array to be connected to the output. The more layers that are passed, the more the number of parameters generated [13]. Figure 4 is a form of the fully connected layer process. With many parameters, it is possible for overfitting to occur in the classification process. So it is necessary to change some values or hyperparameters on the CNN architecture [14].

To overcome the problem of overfitting, a dropout is needed to reduce the effect of overfitting. The use of dropout in this study, we use a dropout value of 0.5. Because the dropout value has been proven to have the smallest error value in making the CNN model [15] and using a learning rate of 0.0001 [16], batch size 32 [17].
2.4. Hyperparameter CNN or analysis sensitivity
Creating a machine learning model requires testing more than one trial. This is because machine learning requires more learning processes to get accurate model results. The Hyperparameter CNN aims to determine the optimal point in the accuracy of the CNN model [18,19]. The use of the CNN hyperparameter is carried out in stages, where each stage will use the best value configuration from the results of the previous parameter test [20]. The testing phase starts from epoch, several layers (feature maps), image size, kernel, stride, padding, dropout, and learning rate.

| Parameters                        | Value                                |
|-----------------------------------|--------------------------------------|
| Epoch                             | 150; 225; 300; 500; 700; 1,000; 1,300; 1,600 |
| Number of Layer Feature Maps      | 2 Layer, 3 Layer, 4 Layer, 5 Layer   |
| Image Size                        | (32×32), (64×64), (128×128), (300×300) |
| Kernel                            | (1x1), (3x3), (5x5), (7x7)            |
| Stride                            | (1x1), (2x2)                          |
| Padding                           | “same”(0) dan “valid”(1)             |
| Dropout                           | 0.5, 0.4, 0.3, 0.2, 0.1               |
| Learning Rate                     | (0.00005), (0.0001), (0.001), (0.01)  |

3. Results and discussion
In this study CNN model was build using image of different quality of clove as input as shown in Figure 5. In the first stage, the CNN model using the proposed CNN architecture, image size 128×128, number of features maps 32 and 64, kernel 3x3, stride 1x1, using padding, max pooling 2x2, dropout 0.5, and learning rate 0.0001. From the results of the first model, the training accuracy value is 61.19%, training loss is 61.19%, 0.9329, and the validation accuracy is 54.6%, the validation loss is 1.064. Then the first CNN model was tested using 400 testing data, then the result gives 65.25% accuracy.

![Figure 5. Image of different clove quality for dataset](image)

| Table 3. Analysis of epoch value
|----------------|-----------|----------|
| Epoch          | Accuracy  | Loss     |
|----------------|-----------|----------|
| 150            | 54.33%    | 0.9820   |
| 225            | 61.67%    | 0.8115   |
| 300            | 65.67%    | 0.7854   |
| 500            | 80.33%    | 0.5572   |
| 700            | 86.67%    | 0.4387   |
| 1,000          | 87.99%    | 0.3556   |
| 1,300          | 89.66%    | 0.3451   |
| 1,600          | 89.33%    | 0.3292   |
According to earlier study [18], machine learning requires more than one experiment to get accurate model results by changing and testing some parameters of the CNN architecture. Therefore we will use the parameters from Table 2 to do a sensitivity analysis or hyperparameter CNN to get the best model. The following are the results of the analysis.

3.1. Epoch
The first stage of variation testing results from the epoch value is obtained in Table 3, where the best epoch value is using the 1300 epoch value. With a validation accuracy value of 89.66% and a validation loss of 0.3451. The higher the epoch value will increase the accuracy of a model, also the longer the training time required [21].

3.2. Number of layer (feature maps)
The results of the second stage of testing the number of layers used are shown in Table 4. The number of convolution layers that have the best accuracy value is the use of 2 2D convolution layers. With feature maps sizes 32 and 64. This has a validation accuracy value of 89.66% and a validation loss of 0.3451.

| Layer                   | Accuracy | Loss  |
|-------------------------|----------|-------|
| 2 layer (32, 64)        | 89.66%   | 0.3451|
| 3 layer (32, 64, 128)   | 87.00%   | 0.3721|
| 4 layer (32, 64, 128, 256) | 81.33% | 0.4533|
| 5 layer (32, 64, 128, 256, 512) | 78.55% | 0.4815|

3.3. Input image size
The third stage is to test the image size variance as input. With input sizes of 32×32, 64×64, 128×128 and 300×300. The image size that has the best accuracy value is the image size of 128×128 pixels Table 5. With a validation accuracy value of 89.66% and a validation loss of 0.3451.

| Image Size           | Accuracy | Loss  |
|----------------------|----------|-------|
| 32 × 32 pixel        | 72.00%   | 0.9245|
| 64 × 64 pixel        | 82.99%   | 0.5130|
| 128 × 128 pixel      | 89.66%   | 0.3451|
| 300 × 300 pixel      | 87.00%   | 0.4572|

3.4. Kernel size
The next stage is to test the use of kernel size variants. The kernel sizes used are 1×1, 3×3, 5×5, and 7×7. The results obtained are the kernel size 5×5 has the highest accuracy value, namely 90.33% validation accuracy and a validation loss of 0.3083. The test results can be seen in Table 6.

| Kernel Size | Accuracy | Loss  |
|-------------|----------|-------|
| 1 × 1       | 85.33%   | 0.4275|
| 3 × 3       | 89.66%   | 0.3451|
| 5 × 5       | 90.33%   | 0.3083|
| 7 × 7       | 84.33%   | 0.4803|

3.5. Stride
The strides test used in this study is the size of 1×1 and 2×2, where it means the displacement of the kernel in every one pixel for the size of 1×1 and two pixels for 2×2. The model with the best value is
owned by the size of the stride of 1×1 in Table 7. With a validation accuracy value of 90.33% and a validation loss of 0.3083.

Table 7. Analysis of Stride

| Strides | Accuracy | Loss  |
|---------|----------|-------|
| 1×1     | 90.33%   | 0.3083|
| 2×2     | 77.66%   | 0.5449|

3.6. Effect of using padding
At this stage, we will test the effect of using padding on a CNN model. According to [22] through his research resulted in a difference in the value of better accuracy using padding in making CNN or RNN models. And suggest using padding on input images that already have the same size because they are more effective. The padding used is called zero paddings. The results obtained in Table 8 are the use of padding has a higher accuracy, namely 86.66% validation accuracy and a validation loss of 0.3083.

Table 8. Effect of using padding

| Padding | Accuracy | Loss  |
|---------|----------|-------|
| ‘Same’  | 90.33%   | 0.3083|
| ‘Valid’ | 88.99%   | 0.3284|

3.7. Dropout
At this stage, we will examine the effect of the dropout value on the CNN model. According to [23] the use of dropout can reduce the occurrence of overfitting and a dropout value that is too high will result in instability during the training process. In the Table 9 there are the results of testing some of the dropout values used. The dropout value of 0.4 got the highest accuracy value, namely with validation accuracy of 91% and a validation loss of 0.2953.

Table 9. Analysis of dropout

| Dropout | Accuracy | Loss  |
|---------|----------|-------|
| 0.5     | 90.33%   | 0.3083|
| 0.4     | 91.00%   | 0.2953|
| 0.3     | 85.66%   | 0.4328|
| 0.2     | 81.33%   | 0.5201|
| 0.1     | 77.6%    | 0.6233|

3.8. Learning rate
The last stage is to determine the value of the learning rate. According to previous study [24], the greater the value of the learning rate, the more unstable the accuracy value will be during the learning process. So, in this study using the learning rate value variant in Table 10. The results obtained are the model with the best accuracy value is owned by the learning rate value of 0.0001 with a validation accuracy value of 91% and a loss value of 0.2953. Because it has a smaller loss value. From these data, it shows that the greater the value of the learning rate will result in an unstable level of accuracy. At a learning rate of 0.001, the loss value is quite large, namely 2.9846 with an accuracy of 64.33%. While the 0.01 learning rate has an accuracy value of 81.99%, but the loss value is 1.1490.

Table 10. Analysis of Learning Rate

| Learning Rate | Accuracy Validation | Loss Validation | Accuracy Training | Loss Training |
|---------------|---------------------|-----------------|------------------|--------------|
| 0.00005       | 88.66%              | 0.3708          | 86.37%           | 0.3393       |
| 0.0001        | 91.00%              | 0.2953          | 91.15%           | 0.2270       |
| 0.001         | 64.33%              | 2.9846          | 99.73%           | 0.0258       |
| 0.01          | 81.99%              | 1.1490          | 99.69%           | 0.0004       |
3.9. The final result of sensitivity analysis or hyperparameter CNN

After analyzing the sensitivity or hyperparameter of CNN, it can be concluded that the best CNN architecture in classifying the quality of cloves, by setting the parameters in Table 11. The results of the model have a validation accuracy of 91%, validation loss of 0.2953 and training accuracy of 91.15%, training loss of 0.2270. Then tested CNN model using data testing as much as 400 testing data (Table 12) which resulted in CNN model accuracy of 87.75%. If compared to the first model, the second model has a better accuracy value. This shows that the use of the CNN sensitivity analysis stage or hyperparameter can provide model results with better accuracy values when compared to not using sensitivity analysis [18–20].

| Table 11. Architecture CNN after used sensitivity analysis or hyperparameter CNN |
| Parameter | Value |
|------------|-------|
| Epoch      | 1,300 |
| Number of Layer | 2 Layer (32, 64 feature maps) |
| Input Image Size | 128×128 pixel |
| Kernel Size | 5×5 pixel |
| Strides    | 1×1   |
| Padding    | ‘Same’ (using padding) |
| Dropout    | 0.4   |
| Learning Rate | 0.0001 |

| Table 12. CNN model accuracy with data testing |
| Classification | Quality 1 | Quality 2 | Quality 3 | Quality 4 |
|----------------|-----------|-----------|-----------|-----------|
| Actual Value   | Quality 1 | Quality 2 | Quality 3 | Quality 4 |
| Quality 1      | 85        | 10        | 1         | 1         |
| Quality 2      | 15        | 83        | 8         | 7         |
| Quality 3      | 0         | 7         | 91        | 0         |
| Quality 4      | 0         | 0         | 0         | 92        |
| Total          | 100       | 100       | 100       | 100       |

4. Conclusion

The use of deep learning methods with the CNN algorithm can be used in the classification of clove quality. The image analysis is able to perform classification with large amounts of data, without the need for pre-processing on the input image. By adding a sensitivity analysis method or hyperparameter, it can help in determining the value of the architecture created. Moreover, sensitivity analysis can be done by taking best model in the previous stage then to be used in the next stage. This proved in this study, where the first model without sensitivity analysis obtained a reading accuracy of 65.25%. Meanwhile, after the CNN sensitivity or hyperparameter analysis was carried out, the accuracy of the test data reading was increased to 87.75%.

Acknowledgements

Thanks to PT. Perkebunan Nusantara XII, Pancusari Plantation, Malang Regency for helping to obtain research data.

References

[1] Setiawan R R and Rosman R 2015 Perspektif 14 27–36
[2] Ministry of Trade Republic of Indonesia 2015 Market Brief Cengkeh di Jerman-ITPC Hamburg (Jakarta: Ministry of Trade Republic of Indonesia)
[3] Yaspin Y N, Widodo D W and Setiawan A B 2020 SEMNAS INOTEK 4 149–54
[4] Pesik P A L, Poekoel V C and Putro M D 2018 J. Tek. Elektro dan Komput. 7 161–6

7
[5] Liang Z, Powell A, Ersoy I, Poostchi M, Silamut K, Palaniappan K, Guo P, Hossain M A, Sameer A and Maude R J 2016 Proc. IEEE Int. Conf. Bioinform. Biomed. 493–6

[6] Zhou L, Zhang C, Liu F, Qiu Z and He Y 2019 Compr. Rev. Food Sci. Food Saf. 18 1793–811

[7] Rivalto A, Pranowo and Santosos A J 2020 AIP Conference Proceedings 2217 30014

[8] Saputra M, Kusrini K and Kurniawan M P 2020 Inspir. J. Teknol. Inf. dan Komun. 10 27–35

[9] Han J, Zhang D, Cheng G, Liu N and Xu D 2018 IEEE Signal Process. Mag. 35 84–100

[10] Rajasegaran J, Jayasundara V, Jayasekara S, Jayasekara H, Seneviratne S and Rodrigo R 2019 Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 10725–33

[11] Abdel-Hamid O, Deng L and Yu D 2013 Interspeech 11 73–5

[12] Scherer D, Müller A and Behnke S 2010 Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition Artificial Neural Networks - ICANN 2010 ed K Diamantaras, W Duch and L S Iliadis (Berlin: Springer Berlin Heidelberg) pp 92–101

[13] Basha S H S, Dubey S R, Pulabaigari V and Mukherjee S 2020 Neurocomputing 378 112–9

[14] Xu Q, Zhang M, Gu Z and Pan G 2019 Neurocomputing 328 69–74

[15] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 J. Mach. Learn. Res. 15 1929–58

[16] Georgakopoulos S V and Plagianakos V P 2017 Proc. EANN. 327–36

[17] Bengio Y 2012 Practical recommendations for gradient-based training of deep architectures In: Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science vol 7700, ed Montavon G, Orr G B, and Müller KR (Springer) pp 437–78

[18] Zhang Y and Wallace B 2015 arXiv: Computer Science 1510 03820

[19] Bergado J R, Persello C and Gevaert C 2016 IEEE IGARSS 1516–9

[20] Adi D, Santosos L W and Tjondrowigunoo A N 2020 J. Infra. 8 302–5

[21] Khamparia A, Gupta D, de Albuquerque V H C, Sangaiah A K and Jhaveri R H 2020 J. Supercomput. 76 8590–608

[22] Nam N T and Hung P D 2019 ACM Int. Conf. Proceeding Ser. 138–42

[23] Poernomo A and Kang D-K 2018 Neural networks 104 60–7

[24] Nurfita R D and Ariyanto G 2018 Emit. J. Tek. Elektro 18 22–7