ILKOM Jurnal Ilmiah Vol. 14, No. 1, April 2022, pp. 1-9
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Classification of coffee bean defects using gray-level co-occurrence matrix and k-nearest neighbor

Mila Jumarlis [a,1,*]; Mirfan [b,2]; Abdul Rachman Manga [c,3]
[a] *STAIN Majene, Jl Blk Kelurahan Totoli, Majene, 91415, Indonesia*
[b] *STMIK Handayani, Jl Adyaksa Baru No I, Makassar, 90231, Indonesia*
[c] *Univeristas Muslim Indonesia, Jl Urip Sumiharjo Km 5, Makassar, 90231, indonesia*
[1] *milajumarlis.mirfan@yahoo.com;* [2] *irfan_phapros@yahoo.co.id;*[3] *abdulrachman.manga@umi.ac.id*
** Corresponding author*

Article history: Received July 07, 2021; Revised November 20, 2021; Accepted November 20, 2021; Available online April 30, 2022

Abstract

Defects in coffee beans can significantly affect the quality of coffee production. Thus, defects in coffee beans result in decreased levels of coffee production. This study aims to implement the GLCM (gray-level co-occurrence matrix) and the K-NN (k-nearest neighbor) method on a web-based program and provided a website to detect coffee bean defects. This study uses the GLCM algorithm to extract the features of the coffee images and uses the K-NN algorithm to classify the defect level of coffee beans. The system development was built using Unified Modeling Language. The development of this website was utilized the programming structure of PHP, HTML, CSS, Javascript, Mozilla Firefox as a browser for the website and MySQL for the database management systems. The results show that the system can provide the output in the form of a classification level of the defect level of the coffee bean images. Then, the accuracy of the coffee bean defect rating was achieved by 90%. Finally, this study concluded that the proposed system could help the coffee farmers determine the defect level of the coffee beans using images input.

Keywords: Coffee beans; Digital Image; GLCM; Classification; K-NN

Introduction

Coffee is one of the plantation commodities with a relatively high economic value among other plantation crops. It also plays an essential role as a source of foreign exchange for the country. Coffee not only plays a vital role as a source of foreign exchange but is also a source of income for coffee farmers in Indonesia [1]. Based on statistical data from Sinjai Plantation and Agriculture Office in 2020, total coffee productivity in the last three years with an area of 1006 HA, including the average production in 2018 was 910 (KG/Ha), while in 2019 and 2020, it was 355 (Kg/Ha). The decline in production was due to poor quality coffee beans and many defective coffee beans. This is because the determination of the type of coffee bean defect is carried out by assessing the physical characteristics of the coffee bean using the human senses with applicable standards. However, under certain conditions, humans cannot properly determine the type of coffee bean defect, such as when they are sick or tired, which results in inaccuracies in determining the type of coffee defect. Other conditions, such as problems with the human senses, such as color blindness, coffee farmers do not understand the standard of quality coffee beans, so coffee farmers cannot classify good coffee beans according to standards. Thus, it is necessary to have a method to detect coffee bean defects so they can be appropriately classified [2].

In this study, the authors use GLCM for the feature extraction process, which is then classified using the K-NN Algorithm. The results of the classification will produce the output of the type of coffee bean defect that has been uploaded. The research of "A. TiaraSari and E. Haryatmi " with the detection of dried corn kernel image uses Convolutional Neural Network Deep Learning [3] so, it gets accurate results. The dry corn kernel image is used as data in the deep learning method before real-time implementation. This technique consists of 3 main stages. The first is preprocessing, wrapping, and cropping. The second is the model formation and system training, and the last is system testing. This study used 20 images of corn kernel as test data from 80 images of corn kernel used in the training dataset. The detection accuracy value of dry corn kernels is influenced by the size of the image and the

http://dx.doi.org/10.33096/ilkom.v14i1.910.1-9

position of the image taken from the smartphone camera. The use of 7 convolutional layers gives an accuracy value ranging from 80% - 100%, so the average value of data testing accuracy is 0.90296.

Research conducted by "M. Rinandar Tasya et al" used the GLCM and Neural Network Methods to classify the quality of carrot ripeness. The GLCM algorithm can convert image data to produce numerical data. Then proceed with looking for accuracy calculations using the Neural Network algorithm, carried out on the RamidMiner application. Carrot classification will later assist in detecting the quality of carrots based on two classes, including carrots with "ripe" and "rotten" quality. In this study, the data used consisted of 10 training data and 40 test data. The results of this study indicate that the classification level of carrots gets an accuracy value of 72.50% [4].

Another study by "Febian Fitra Maulana et al" used Convolutional Neural Network to classify fruit images. In this study, the CNN architecture is used with a combination of 3 Convolutional Neural Networks and 2 Fully Connected Layers. At the stage of making a classification system that uses deep learning, there are several main process stages: data collection, system design, training, and testing. The processed dataset is a fruit image dataset derived from the Fruit-360 dataset. The data classes used are 15 classes from 111 classes in the fruit-360 dataset. The learning process results obtained a CNN model with 100% accuracy and a loss of 0.012. In testing the CNN model using 45 fruit image samples, an accuracy of 91.42% was obtained. So, it can be concluded that this CNN method can classify images well [5].

The implementation of the Convolutional Neural Networks method in agriculture is also carried out by [6], classifying the Pear Image Using Convolutional Neural Networks. This examination intends to overcome the issue of structuring organic products manually by utilizing one of the Deep Learning calculations to characterize an image, especially CNN. This investigation led to precision testing on two cycles, specifically preparation and testing, with the test results that the accuracy obtained was 100% for the preparation and testing using 100 tests of new information with a precision value of 98%. The use of CNN in agriculture was also carried out by Dhiya Mahdi Asriny et al. by implementing it on citrus fruits. The total dataset used is 1000 images of oranges, where each class consists of 200 images divided into 60% as training data, 20% as validation data, and 20% as test data. The level of accuracy obtained from this study is 93.8% [7].

Furthermore, research using CNN was also carried out by "Isna Wulandari et al." In this study, the data used are digital images of spices and herbs taken by crawling on the Google search engine. The digital image used consists of three categories: ginseng, ginger, and galangal. The total image collected for the sample is 300, with 100 images in each category. The data is divided into 2, including training data and testing data, with a comparison of training data and testing data of 80%: 20%. From the study results, the training data's accuracy value was 0.9875, and the loss value was 0.0769. The test data accuracy value is 0.85, and the loss value is 0.4773. Meanwhile, testing with new data, including three images for each category, resulted in an accuracy of 88.89% [8].

Another study that implements the CNN method is [9]. This study uses CNN with a model architecture using 8 Convolutional 2D layers with filters (16, 16, 32, 32, 64, 64, 128, 128). The first input layers are (20,20) and the following layers (5,5 and 3,3). The types of pooling used in this study are MaxPooling and Average Pooling. The fully Connected Layer used is (256, 128) and using Dropout (0.2). The dataset was obtained from the International Skin Imaging Collaboration 2018 with 10015 images. Based on the testing and evaluation results, an accuracy of 75% was obtained, with the highest precision and recall values found in the benign class, which were 0.80 and 0.82, respectively, and the f1_score value was 0.81. Another study in agriculture was conducted by [10] to detect wood defects using the Susan edge detection method and statistical feature extraction. In this research, a wood detection system has been designed to classify normal wood (without defects) and damaged wood using the SUSAN edge detection method and second-order statistical feature extraction, with an accuracy rate of 90.67% and a computation time of 2.5 seconds.

Various studies have been carried out regarding the detection of coffee bean defects. However, the implementation has not been seen in the agricultural practice for sorting coffee bean defects. In the agricultural practice, it is necessary to implement the detection of coffee bean defects. This is because coffee farmers do not understand and classify good coffee beans according to standards. This method's importance of detecting coffee bean defects is useful for detecting cracked, black, and broken coffee beans and Partial black. This study will use the Feature Extraction Method Using GLCM and the K-NN Classification Method to detect coffee bean defects based on existing research. The GLCM algorithm performs feature extraction from coffee sample images which will then be classified using the K-NN algorithm [11]. Based on that reason, this study aims to test the ability of the Feature

Extraction Method Using GLCM and the K-NN Classification Method in detecting coffee bean defects using only images.

Method

A. Gray-Level Co-occurence Matrix

The Gray-Level Co-occurrence Matrix (GLCM) method is a second-order extraction for statistical texture features. It shows a statistical relationship between 2 pixels. GLCM is a matrix with the number of rows and columns proportional to the number of gray levels (G) in an image. The GLCM method uses a grayscale image [12]. The first step to calculating GLCM highlights is to convert an RGB image into a grayscale image. The next step is to create a co-occurrence grid and determine the spatial relationship between the reference pixel and the adjacent pixels depending on the point and distance d. The next step is to create a symmetrical grid by adding a co-event framework to the rendering lattice. Then standardize the symmetrical grid by ensuring the possibilities of each component of the framework. The final step is to calculate the GLCM highlights. Each component 14 is determined by a one-pixel distance [4]. GLCM has four angular directions (figure 5), which are commonly used to create a GLCM matrix, which is the angular directions of 0°, 45°, 90°, and 135° [13].

B. The procedure of Gray-Level Co-occurrence Matrix

The main demand scalable strategy is an element recovery technique that relies on the histogram attributes of the image. The histogram shows the possible occurrence of pixel dimming values in an image. Several first-demand trademark limits can be determined from the qualities in the subsequent histograms, including mean, skewness, fluctuation, kurtosis, and entropy. For this situation, we need to request a second request highlight. This methodology works by framing the co-occurrence of the image information, followed by deciding on the quality as a component of the grid. Co-event implies co-event, specifically the number of occurrences of one degree of pixel value adjacent to one degree of another pixel value within a certain distance (d) and point direction (θ) (tetha). Distance is expressed in pixels, and orientation is expressed in degrees. The orientation is formed in four angular directions, which are 0°, 45°, 90°, and 135°. The distance between pixels is usually set at 1 pixel. This composition calculates the number of pixel groups that meet the correlation [12].

C. K-Nearest Neighbor (K-NN)

K-Nearest Neighbor is a method for classifying objects based on learning data closest to the object. Learning data is projected into a multidimensional space, where each dimension represents a feature of the data [14]. The dimension space is divided into parts based on the classification of learning data. The best value of k for this algorithm depends on the data. In general, a high value of k will reduce the effect of noise on the classification but make the boundaries between each classification more blurred. A good value of k can be chosen by parameter optimization, for example, by using cross-validation. A special case where the classification is predicted based on the closest learning data (in other words, k = 1), which is usually called the nearest neighbor [14].

D. The procedure of K-Nearest Neighbor

The working principle of K-NN is to find the shortest distance between the data to be evaluated and its K nearest neighbors in the training data. This technique belongs to the nonparametric classification group. Here we ignore the distribution of the data we want to group [15]. This technique is very simple and easy to implement. Similar to the clustering technique, we group new data based on the distance of the new data to some data/neighbors. In the classification process, this algorithm does not use any model to be matched and is only based on memory [16].

Results and Discussion

A. Image Acquisition

At the image acquisition stage, it can be seen in **Figure 1**, the image will be taken from the sample that want to be classified.

Figure 1. Original Image

B. Pre-process

The existing image is then converted to grayscale, then resized to 150x150 Pixels. **Figure 2** shows a grayscale image.

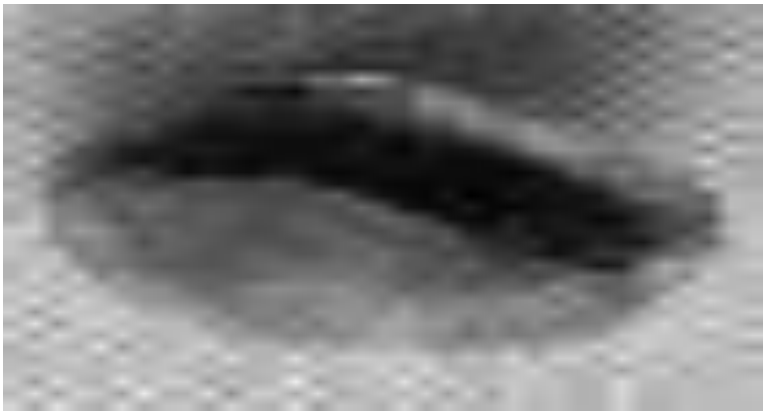


Figure 2. Grayscale Image

C. Segmentation

At this stage, the image that has been converted to grayscale will be extracted as shown in **Figure 3**, where the four GLCM features are energy, entropy, homogeneity, and contrast using the formulas for each of the four features.

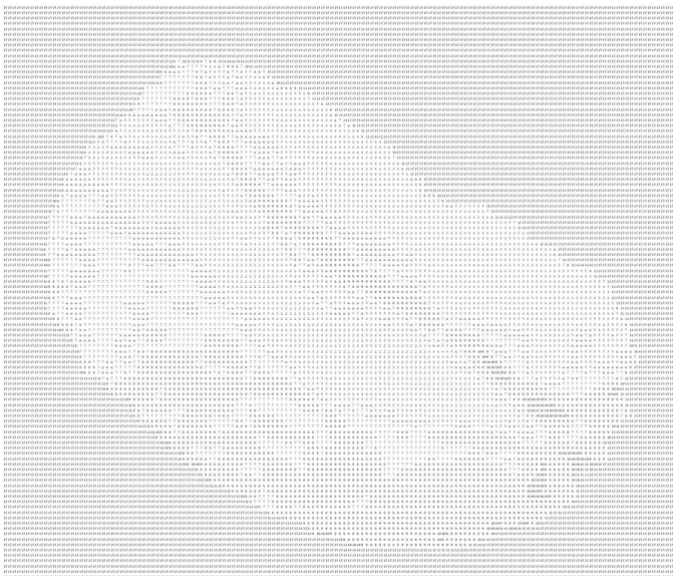


Figure 3. The matrix for which the features will be calculated

D. GLCM Feature Calculation

After the results of the grayscale image are used as a matrix, then the GLCM features will be calculated, including energy, entropy, homogeneity, and contrast. The calculation results can be seen in **Table 1** which is taken from four angles, they are 0^0, 45^0, 90^0, 135^0.

Table 1. GLCM Feature Calculation Results

Feature	Score
Energy	0.0997
Entropy	0.0413
Homogenity	0.6386
Contrast	3.1041

E. Image Classification with K-NN Algorithm

The image is divided into two types of data, training image data and test image data. Training image data is data that already exists and is used as a dataset in the classification process. While the test image data is the data used for testing the program [17]. The following is an example of training data with GLCM feature values, which can be seen in **Table 2**. Calculating the speed of the test data against each training data:

Table 2. GLCM Feature Calculation Results

Sample Id	Description of coffee beans	Features of GLCM			
		Energy	Entropy	Homogenity	Contrast
45	Black	0.36	0.11	0.95	0.10
46	Black	0.22	0.06	0.93	0.19
47	Black	0.31	0.09	0.90	0.29
48	Broken	0.10	0.04	3.10	0.64
49	Broken	0.17	0.05	0.90	0.26
57	Broken	0.60	0.18	0.94	0.50
58	Broken black	0.42	0.13	0.92	1.45
59	Broken black	0.51	0.15	0.91	0.84
60	Broken black	0.59	0.18	0.95	1.47
61	Partial Black	0.63	0.19	0.93	1.42
64	Partial Black	0.29	0.09	0.90	1.03
65	Partial Black	0.53	0.16	0.95	0.68
66	Partial Black	0.35	0.11	0.93	0.77
67	Normal	0.61	0.18	0.95	0.77
68	Normal	0.49	0.15	0.94	0.73

The previously generated test data will be classified according to the quality of the 10-training data closest to the test data with the formula (1).

$$D(a,b) = \sqrt{\sum_{k=1}^{d}(a_k - b_k)^2} \quad (1)$$

The following is the data that has obtained the value of its proximity to the test data presented in **Table 3**.

Table 3. Table of Proximity Values

Sample Id	Proximity Value	Description of coffee beans
45	0.95	Black
46	0.84	Black
47	0.76	Black
48	0.59	Broken
49	0.76	Broken
57	0.77	Broken
58	0.77	Broken black
59	0.58	Broken black
60	0.88	Broken black
61	0.86	Partial black
64	0.47	Partial black
65	0.66	Partial black
66	0.52	Partial black
67	0.69	Normal
68	0.61	Normal

For the value of proximity from small to large and determine the 10 closest data can be seen in **Table 4**.

Table 4. GLCM Feature Calculation Results

Order	Sample Id	Proximity Value	Description of coffee beans
1	64	0.47	Black
2	66	0.52	Black
3	59	0.58	Broken Black
4	48	0.59	Broken
5	68	0.61	Broken Black
6	65	0.66	Partial Black
7	67	0.69	Normal
8	49	0.76	Broken
9	47	0.76	Black
10	58	0.77	Black

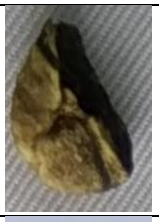
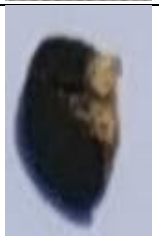
In **Table 4** it has been concluded that 10 data are closest to the existing test data, then from these 10 data, there are 5 descriptions of coffee bean defects with the following percentages: Partial Black $= 100\% \frac{20}{10} = 20\%$, Broken Black $= 100\% \frac{2}{10} = 20\%$,3. Broken $= 100\% \frac{1}{10} = 20\%$, Normal $= 100\% \frac{1}{10} = 10\%$, Black $= 100\% \frac{4}{10} = 40\%$.

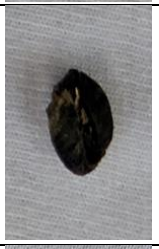
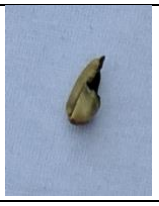
The presentation of coffee bean defects can be seen in **Figure 5** which is generated by the system using the GLCM and K-NN algorithms.

Figure 5. Results of GLCM and K-NN Algorithm Classification

Testing the level of accuracy of the results of the rating of coffee beans produced in the system is carried out by comparing the results of the actual rating of coffee beans with the results of the rating of coffee beans produced by the system. The following comparison results are presented in **Table 5**.

Table 5. Table of Accuracy Testing

No	Image	System-generated ratings	Actual Rating	Remarks
1		Broken beans (50%)	broken beans	Matched
2		Black and Broken beans (40%)	Black and Broken Beans	Matched
3		Broken beans (60%)	Broken beans	Matched

No	Image	System-generated ratings	Actual Rating	Remarks
4		Broken beans (40%)	Broken beans	Matched
5		Partial black beans (60%)	Partial black beans	Matched
6		Partial black Beans (60%)	Partial black beans	Matched
7		Black beans (60%)	Black beans	Matched
8		Broken and Black beans (50%)	Black beans	Matched
9		Broken beans (30%)	Partial black Beans	Not matched
10		Broken beans (50%)	Broken beans	Matched

Based on the comparison information in the table above, it is concluded:

$$\textbf{Accuracy Level} = \frac{\textbf{Σ Matched Test Data}}{\textbf{Σ Total Test Data}} \textbf{x 100}$$

$$= \frac{9}{10} \text{x } 100\ \% = \ 90\ \%$$

The level of accuracy of the system in classifying defects in coffee beans in this research is 90%.

Conclusion

From the test results of the design process for the implementation of the GLCM and K-NN methods to detect coffee bean defects in image-based programs, it is obtained that the accuracy of this system provides results in the form of a classification of the level of defects from the coffee bean image uploaded to the system, with the accuracy of the coffee bean defect rating that the system provides is 90%. This system can assist coffee farmers in knowing the level of defect in coffee beans from their production. It can be done by using images and the implementation of the GLCM algorithm image processing by changing the sample image which is still in RGB format (Red, Green, Blue) to a grayscale format with a gray standard of 15, then look for the value of the GLCM feature extraction with the existing formula.

References

[1] B. Marhaenanto, D. W. Soedibyo, and M. Farid, "Penentuan lama Sangrai Kopi Terhadap Variasi Derajat Sangrai Menggunakan Model Warna Rgb Pada Pengolahan Citra Digital (Digital Image Processing)," *J. Agroteknologi*, vol. 09, no. 02, pp. 1–10, 2015.

[2] E. R. Arboleda, A. C. Fajardo, and R. P. Medina, "An image processing technique for coffee black beans identification," *2018 IEEE Int. Conf. Innov. Res. Dev. ICIRD 2018*, no. May, pp. 1–5, 2018.

[3] A. TiaraSari and E. Haryatmi, "Penerapan Convolutional Neural Network Deep Learning dalam Pendeteksian Citra Biji Jagung Kering," J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 5, no. 2, pp. 265–271, Apr. 2021, doi: 10.29207/resti.v5i2.3040.

[4] M. R. Tasya, B. S. W. A, and E. T. Luthfi, "Klasifikasi Kualitas Kematangan Wortel Menggunakan Metode GLCM (Gray Level Co-Occurrence Matrix) Dan Neural Network," *J. FATEKSA J. Teknol. dan Rekayasa*, vol. 5, pp. 1–10, 2020.

[5] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 01, pp. 104–108, 2019.

[6] S. Juliansyah and A. D. Laksito, "Klasifikasi Citra Buah Pir Menggunakan Convolutional Neural Networks," *J. Telekomun. dan Komput.*, vol. 11, no. 1, p. 65, 2021.

[7] D. M. Asriny, S. Rani, and A. F. Hidayatullah, "Orange Fruit Images Classification using Convolutional Neural Networks," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 803, no. 1, 2020.

[8] I. Wulandari, H. Yasin, and T. Widiharih, "Klasifikasi Citra Digital Bumbu Dan Rempah Dengan Algoritma Convolutional Neural Network (Cnn)," *J. Gaussian*, vol. 9, no. 3, pp. 273–282, 2020.

[9] Luqman Hakim, Z. Sari, and H. Handhajani, "Klasifikasi Citra Pigmen Kanker Kulit Menggunakan Convolutional Neural Network," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 2, pp. 379–385, 2021.

[10] P. D. Wananda, L. Novamizanti, and R. D. Atmaja, "Sistem Deteksi Cacat Kayu dengan Metode Deteksi Tepi SUSAN dan Ekstraksi Ciri Statistik," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 6, no. 1, p. 140, 2018.

[11] C. J. Kuo *et al.*, "A Labor-Efficient GAN-based Model Generation Scheme for Deep-Learning Defect Inspection among Dense Beans in Coffee Industry," *IEEE Int. Conf. Autom. Sci. Eng.*, vol. 2019-Augus, pp. 263–270, 2019.

[12] E. Alvansga, "Pengenalan Tekstur Menggunakan Metode Glcm Serta Modul Nirkabel," *Comput. J.*, pp. 70–75, 2019.

[13] R. Ocsan, I. Safitri, F. T. Elektro, U. Telkom, G. L. C. Matrix, and K. Neighbor, "Dan Knn Classification of Kangkung Vegetables and Detection of Chemical Exposure Using Glcm and Knn," vol. 8, no. 2, pp. 1481–1489, 2021.

[14] Isman, A. Ahmad, and A. Latief, "Perbandingan Metode KNN Dan LBPH Pada Klasifikasi Daun Herbal," vol. 1, no. 10, pp. 557–564, 2021.

[15] I. T. Sitorus, D. R. Simarmata, and I. Christinawati, "Pengenalan Biji Kopi Arabika varietas Sigarar Utang Lintong Nihuta Berdasarkan Parameter Tekstur Menggunakan Machine Learning (Studi Kasus : KSU POM Humbang Cooperative)," pp. 6–8, 2020.

[16] C. Pinto, J. Furukawa, H. Fukai, and S. Tamura, "Classification of Green coffee bean images basec on

defect types using convolutional neural network (CNN)," *Proc. - 2017 Int. Conf. Adv. Informatics Concepts, Theory Appl. ICAICTA 2017*, 2017.

[17] A. Prabowo, D. Erwanto, and P. N. Rahayu, "Klasifikasi Kesegaran Daging Sapi Menggunakan Metode Ekstraksi Tekstur GLCM dan KNN," *Electro Luceat*, vol. 7, no. 1, pp. 74–81, 2021.