

Refining Content-Based Segmentation for Prediction of Coffee Bean Quality

Suhendro Yusuf Irianto^{a1}

^aInstitut Informatika dan Bisnis Darmajaya
JI Zainal Abidin Pagar Alam No. 93 Gedong Meneng, Bandar Lampung, Indonesia
¹suhendro@darmajaya.ac.id (Corresponding author)

Abstract

Coffee has substantial economic value and is a key foreign exchange source for numerous nations, including Indonesia. Moreover, it is a primary livelihood for many of the country's farmers. Recently, there have been challenges in accurately predicting the quality of coffee beans, primarily due to time, inconsistency, and imprecision issues. Consequently, this study delves into the application of region-growing segmentation and content-based image retrieval (CBIR) techniques to enhance the prediction of coffee bean quality. The proposed hybrid approach, which combines region growing and CBIR methods, aims to improve the precision for forecasting cacao bean quality. Additionally, the research introduces an automated tool that employs these hybrid techniques for quality prediction. The study conducted experiments using a dataset of 400 premium and 400 low-quality coffee beans sourced from the University of Syiah Kuala in Indonesia. The results of the experiments demonstrate a commendable precision rate of 85.4%, showcasing significant improvement compared to certain previous studies.

Keywords: Segmentation, Region growing, CBIR, Coffee bean

1. Introduction

Coffee is a commodity with high economic value, and it plays a vital role as a source of foreign exchange for several countries, including Indonesia. Furthermore, it serves as a source of livelihood for most farmers in the country. One of the major problems affecting small-scale farmers is the low quality of products, which do not meet the standard requirements [1]. Several reports also stated that selecting coffee beans based on the specified standard is important because the price depends on the quality. However, an expert is needed to determine the grade, which can be classified into good, fine, and quality. Since 1987, Coffee Bean Quality Standards have been enforced using the Triage System. It was later changed to the Defects Value System based on the decision of the ICO (International Coffee Organization) on October 1, 1983. In this method, the higher the number of defects, the lower the quality of the coffee and vice versa [2].

The most straightforward technique in determining a bean deficiency involves direct observation. Still, the decision to choose manually has several drawbacks due to subjective selection and the lack of understanding of science. In determining quality, it is essential to have the right system to analyze the problem. The Content Based Image Retrieval (CBIR) method is an effective solution to the problem of selecting coffee beans based on the standardization requirements, and it can be used to detect the texture of the product. This is often done by taking texture feature values based on the Gray Level Co-Occurrence Matrix (contrast, correlation, homogeneity, and energy) obtained from seed images. The CBIR method helps produce coffee beans that comply with the standardization requirements for defects. Several studies have been carried out, such as [3] [4] [5], which used the CBIR technique with 90 image data and a precession rate of 55.20%. In their research [3], they also use Arabica coffee bean samples in several regions in Indonesia, one of which is the coffee beans from South Kalimantan, North Sumatra, East Java, and West Java, as used in our works. Our experiments indicate that the achieved precision or accuracy stands at 85.4%, surpassing the levels observed in certain prior studies. Additionally, a separate experiment demonstrates a precision rate of 55.20%. These findings suggest that our experiments have

yielded a precision rate of 85.4%, showcasing an improvement over specific earlier research endeavors. These results collectively underscore the advancements made in our study.

Three coffee bean types are used: Robusta, Arabica, and Liberika. However, in their work, we deployed Arabica coffee beans [6]. Furthermore, the weakness in this study was that a small amount of image data was used, namely 30 samples from each type of coffee; hence, the level of accuracy obtained was relatively low. Meanwhile, [7] used the AHP (Analytical Hierarchy Process) method and Arabica coffee, where the outcome was quality with an accuracy rate of 85%, while the input included moisture content, bean defects, and land height. The weakness of this study was the 15% error in the form of inaccuracy between manual calculation data and the total data obtained from the AHP method. [8] used the Backpropagation Artificial Neural Network (ANN) method with 238 data, where 138 and 100 were for training and testing of the samples, respectively. The results showed the best level of accuracy of 80% with a total of 10 test data. Therefore, this study aims to examine the accuracy of the prototype content-based image retrieval system combined with the segmentation method. The results of this study are expected to contribute to the development of digital image processing science and assist interested parties in identifying bean quality, especially coffee.

2. Research Methods

2.1. Data Collection

The dataset used in this study was taken from Syiah Kuala University (<https://comvis.unsyiah.ac.id/usk-coffee/>), which consisted of 400 low-quality (coffee bean defect (bad) images and 400 premium-quality (good) coffee beans. Furthermore, the quality of the samples obtained was determined using the defect value.

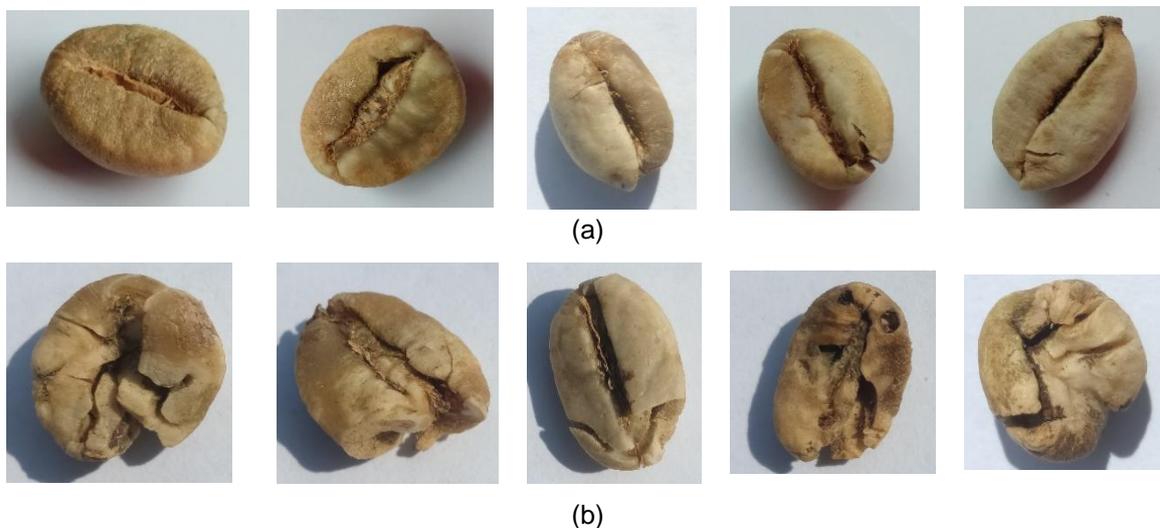


Figure 1. Examples of coffee beans based on the value of bean defects: (a) Good, (b) Bad

The dataset was a collection of interconnected images stored simultaneously and used to provide further information. Additionally, the system created in this study aimed to obtain similarities in coffee beans through the CBIR (Content Based Image Retrieval) and image segmentation processes. The first step involved segmenting the manually cropped image and separating it from the background. It was subjected to several resizing and grayscale processes in the preprocessing stage. The system then performed a feature extraction to obtain the characteristics of the input image, which were compared with the data stored in the training database. The process was carried out using the texture feature extraction with GLCM (Gray Level Co-occurrence Matrix). Feature vectors were then obtained in the database with specific closeness values to be retrieved. The system could distinguish between coffee's good and bad categories based on the results.

2.2. Region Growing Segmentation Algorithm

This study used the region segmentation method to separate the desired region from the unnecessary part. According to [9] and [10], region-growing segmentation is a computer vision and image processing technique used for segmenting an image into meaningful regions or segments based on specific similarity criteria. The main idea behind region growing is to start with a seed pixel or region and iteratively expand the region by adding neighboring pixels that satisfy specific similarity conditions. The process involves the following steps: seed selection, pixel similarity criterion, region growing process, and stopping conditions. The algorithm from region growing segmentation (RGS) can be explained in Figure 2.

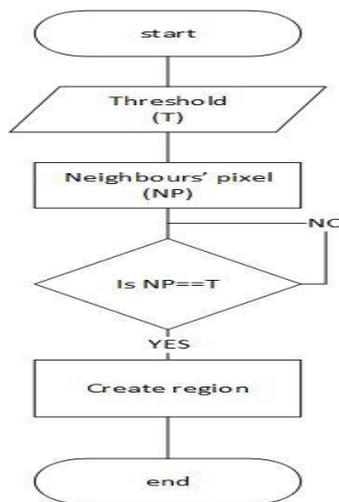


Figure 2. Segmentation of Region-Growing

Furthermore, as mentioned in [11] and in [11], the region-growing technique can be outlined through several stages: i) Initially, the threshold value is determined, establishing the seed point of the image as the initial phase of region or neighbor formation) Subsequently, each region of neighboring pixels is examined to assess their similarity with the seed pixel. iii) When similarity is identified, a region is promptly generated. Conversely, if no similarity is found, the process reverts to rechecking the entire region to identify pixels that meet the criteria. iv) The growth of a region concludes when no additional pixels fulfill the inclusion criteria, leading to the creation of a new region. v) The process is then considered complete.

2.3. CBIR System Design

As introduced by [12], it aims to describe the outline of the overall system workflow. The structural algorithm CBIR is presented in the following sentences. The Content-Based Image Retrieval (CBIR) architecture is a sophisticated framework designed to enhance the efficiency and precision of image searches based on visual content characteristics. At its core, CBIR relies on a structured approach, starting with the preprocessing and indexing images within a database. Visual features such as color, texture, shape, and spatial information are extracted from each image, forming the foundation for subsequent content-based comparisons. Users interact with the system through a query interface, inputting preferences or example images to initiate searches. The CBIR system then measures the similarity between the visual features of the query and those of images in the database, often utilizing distance metrics like Euclidean distance or cosine similarity. The top-ranked images are retrieved and presented to the user. Advanced CBIR architectures may integrate machine learning for adaptive learning, support multiple modalities, address scalability and efficiency concerns, and consider security and privacy considerations. Continuous evaluation and benchmarking ensure the system's competitiveness and effectiveness in delivering visually relevant results across diverse applications and domains.

2.4. Preprocessing Stage

The preprocessing stage was carried out to process image data and improve the quality. The first step was cropping, which helped select and trim the image of the coffee beans obtained from data acquisition. However, in this study, the cropping process was performed manually. The reason for performing the cropping process manually is to maintain quality and precision in selecting the parts to be cut or retained from an image or plant material. Manual cropping allows us to make decisions that consider visual composition, focus, and image context more accurately. Resizing was also done to equalize the sample size and speed up the calculation process. The final stage was grayscale, which involved changing the color to gray. By changing the RGB of each image pixel to the same value, each has the same value for the three-color elements, and a grayscale matrix value was obtained.

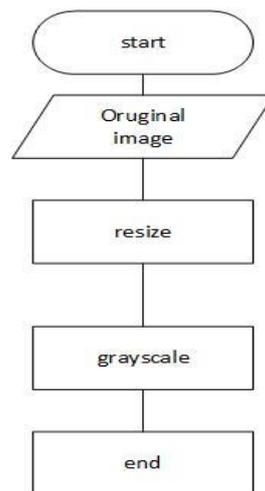


Figure 3. Preprocessing

The preprocessing stage is carried out to process image data to enhance image quality. The first process is cropping, which helps select and cut the acquired coffee bean images. However, in this study, the cropping process is done manually. Resizing is performed to make the image sizes uniform, which expedites the calculation process. The final stage is grayscale conversion, which transforms colors into shades of gray. By converting the RGB values of each image pixel into a single value, every pixel has the same value for all three-color components, resulting in a grayscale matrix value. In summary, this phase involves grayscale conversion, which alters colors into shades of gray. By transforming the RGB values of each pixel in the image into a single consistent value, uniformity is achieved across all three color channels. As a result, a grayscale matrix is obtained, representing the image in shades of gray.

Figure 3 can be explained as the initial phase of preprocessing; there are two sets of stages that the training image will go through, namely the "resize" stage, which aims to adjust the image size uniformly, and the "greyscale" stage, which functions to transform the RGB image into grayscale. Texture feature extraction: in this stage, the extraction of features is performed based on the texture of the seeds. At the same time, the database stores data resulting from the extraction of texture features.

2.5. Texture Extraction

After the preprocessing step, the next step in texture feature extraction is a pattern regularity formed from the arrangement of pixels in an image. An image is stated to have texture information when it has a pattern or characteristic in an area that appears repeatedly at intervals of a certain distance and direction [13]. Moreover [14] stated that there are several methods for extraction, such as:

- a. The first-order feature was based on the characteristics of the image histogram. They generally distinguished macrostructural textures (periodic repetition of local patterns). The first-order features include mean, variance, skewness, quotation, and entropy.

- b. The second-order feature was based on the probability of a neighbor relationship between two pixels at a certain distance and angular orientation. Furthermore, they were generally used to distinguish microstructural textures (local patterns and repeats are less pronounced). Second-order features include angular second moment, contrast, correlation, variance, inverse different moment, and entropy.
- c. Other texture descriptors are laws of texture measurement, wavelet, steerable pyramids, and local binary patterns (LBP).

GLCM described the number of pixel pairs concerning the frequency at a distance d and the variation of the inclination angle θ to calculate the feature value [15]. There were four static values commonly used in this matrix, namely Contrast, Correlation, Energy, and Homogeneity with angles of 0° , 45° , 90° , and 135° , [16][17].

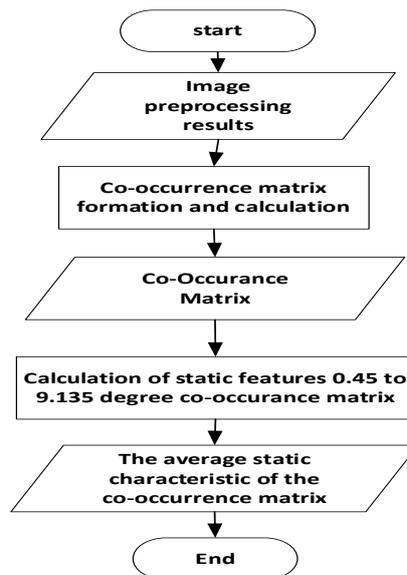


Figure 4. GLCM Diagram

The depiction in Figure 4 can be articulated subsequently: Following the completion of the image's preprocessing phase, the subsequent stride encompassed the extraction of GLCM texture features. After that, the primary phase in transforming the image to grayscale initiates with the matrix normalization process. Within this phase, steps are executed, including quantification and generation of the count matrix, transposition, combining the count matrix with its transposed counterpart, and culminating with the normalization procedure.

This intricate sequence of operations constitutes a pivotal stage in the overall process. Consequently, the outcomes stemming from the matrix normalization were effectively employed to compute notable attributes encompassing contrast, correlation, homogeneity, and energy. The subsequent step involved averaging the acquired values and angles, resulting in their amalgamation into a singular value. A combination of adept hardware and software configurations was employed in this research endeavor. A laptop with an Intel® Core™ i3-2330M CPU clocked at 2.20GHz, 2048MB of RAM, a 64-bit operating system, and a 128GB SSD was utilized.

3. Result and Discussion

This chapter explains the general workflow of the system. Some of the program functions or methods used in this study and their implementation into the simulation software have been described at the design stage in the previous chapter. The steps taken to translate the design were based on the results of the analysis in a language understood by the computer, as well as the application of the software in actual situations. In the process of looking for similarities, the images were resized and greyscaled using the function from Matlab: `Img_rsz =`

`imresize(Img,[200,200]);` and `Img_gray = rgb2gray(Img_rsz);`. Furthermore, the image was returned to RGB using the Matlab function `ImgRGB = cat (3 R, G, B).` Figure 5 illustrates the following MATLAB GLCM complete codes.

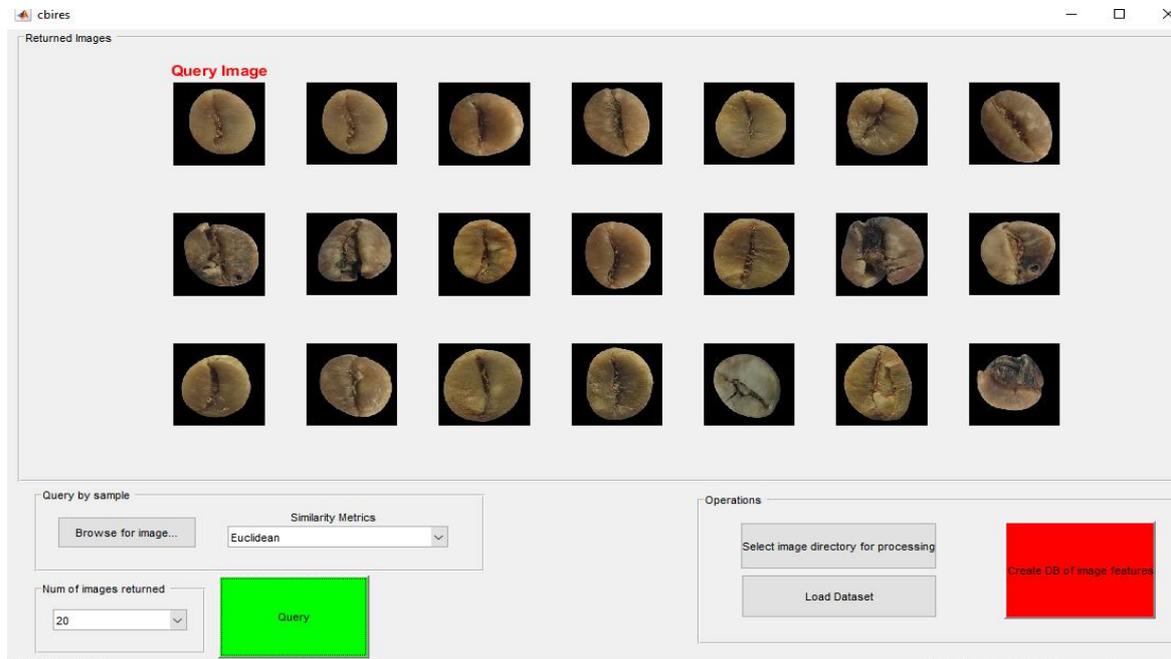


Figure 5. Example of Coffee bean image retrieved

The preprocessing in the study was carried out on the dataset, including changing the size to 200 x 200 pixels using the function `Img_rsz = imresize(Img,[200,200]` and `Img_gray = rgb2gray(Img_rsz)`. The next step was to retrieve texture features in each image database by using a function in Matlab: `pixel_dist = 1; GLCM = graycomatrix(Img_gray, 'Offset,' [0 pixel_dist]); -pixel_dist pixel_dist; -pixel_dist 0; -pixel_dist -pixel_dist]; stats = graycoprops(GLCM, {'contrast,' 'correlation,' 'homogeneity', 'energy'})`. Furthermore, the feature extraction was performed and stored as an image dataset. Resizing was implemented to speed up the search for parameters in the data using Matlab: `Img_rsz = imresize(Img,[200,200])`. After the image data was sized, the RGB matrix values were obtained for easy manual calculation. By taking a sample, the Matlab function used was: `red=rsz(:,:,1)`; `green=lmg_rsz(:,:,2)`; `blue=lmg_rsz(:,:,3)`. Based on the sample used, the values of R, G, and B were successfully extracted into matrix form, and the results are presented below:

	1	2	3	...	200
1	243	239	238	...	243
2	239	238	240	...	240
3	247	245	242	...	242
4	246	244	242	...	246
5	237	239	241	...	247
6	235	239	241	...	245
7	237	240	239	...	241
8	234	233	232	...	244
9	234	231	233	...	246
10	243	240	241	...	243
...
200	213	211	204	...	231

RED

	1	2	3	...	200
1	247	243	242	...	247
2	243	242	244	...	244
3	251	249	246	...	246
4	250	248	246	...	250
5	241	243	245	...	251
6	239	243	245	...	249
7	241	244	243	...	245
8	238	237	236	...	248
9	238	235	237	...	251
10	247	244	245	...	248
...
200	212	211	204	...	235

GREEN

	1	2	3	...	200
1	246	242	241	...	246
2	242	241	243	...	243
3	250	248	245	...	245
4	249	247	245	...	249
5	240	242	244	...	250
6	238	242	244	...	248
7	240	243	242	...	244
8	237	236	235	...	247
9	237	234	236	...	248
10	246	243	244	...	244
...
200	215	213	206	...	233

BLUE

Figure 6. RGB (Red, Green, Blue) matrix values

In Figure 6, the matrix value was R (Red), with a size of 200x200 pixels. At 1.1, the value of R = 243; at 200x200, it was 231. Furthermore, the matrix G (Green) was 200x200 pixels. In pixels (1.1), the value of R = 247; at 200x200, it was 235. Furthermore, it shows the value of matrix B (Blue) with a size of 200x200 pixels. At pixels (1,1), the value of R = 246; at 200x200, it was 233.

After going through the resizing stage, the next step is to change the RGB mode image to grayscale. `Img_gray = rgb2gray (Img_rsz);`

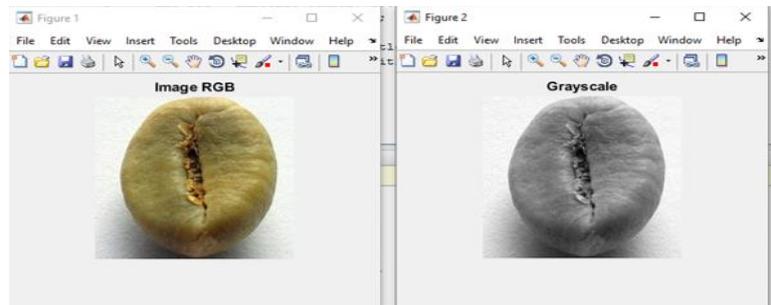


Figure 7. Resized image, (b) grayscale image

The function of converting RGB images to grayscale was to simplify the pixel values in an image. The initially had 3 pixel values, namely R, G, and B, and it only had 1 gray value after being converted to greyscale. Table 4.4 shows that pixel (1.1) has a value of R = 243, G = 247, B = 246, while 1.2 has a value of R = 239, G = 243, B = 242. For pixel 2.1, the value was R = 239, G = 243, B = 242, while 2.2 was R = 238, G = 242, B = 241. For the RGB to grayscale conversion, the value of each pixel was entered into the formula, and the results are presented below [18].

	1	2	3	...	200		1	...	45	46	47	...	200		1	...	45	46	47	...	200	
1	246	242	241	...	246	1	246	...	245	242	240	...	246	1	0	...	0	0	0	...	0	
2	242	241	243	...	243
3	250	248	245	...	245	45	240	...	245	244	246	...	247	45	0	...	0	0	0	...	0	
4	249	247	245	...	249	46	240	...	249	250	250	...	246	46	0	...	0	0	0	...	0	
5	240	242	244	...	250	47	245	...	243	243	249	...	245	47	0	...	0	0	0	...	0	
6	238	242	244	...	248	48	250	...	244	232	240	...	245	48	0	...	0	0	0	...	0	
7	240	243	242	...	244	49	246	...	247	238	243	...	243	49	0	...	0	0	0	...	0	
8	237	236	235	...	247	50	241	...	248	248	248	...	244	50	0	...	0	0	0	...	0	
9	237	234	236	...	249	51	239	...	248	251	247	...	246	51	0	...	0	0	0	...	0	
10	246	243	244	...	246	52	244	...	243	251	249	...	247	52	0	...	0	0	0	...	0	
...	53	244	...	243	251	249	...	247	53	0	...	0	0	0	...	0	
200	213	211	204	...	234	...	250	...	246	249	245	...	246	200	0	...	0	0	0	...	0	

RGB to grayscale conversion value Image matrix Segmentation results

Figure 8. RGB Image segmentation

In Figure 8, pixel (1.1) has a grayscale value of 246, pixel (1.2) is 242, pixel (2.1) is 242 and pixel (2.2) is x, and up to pixel (200 x 200) has a grayscale value of 234. The region-growing segmentation in this study is illustrated in Figure 8 and can be explained in 5 steps as follows:

- The first step involved the determination of the pixel point seed, and the selected value was missa100. Furthermore, this determines the region membership criterion, namely 100-199. A 10x10 matrix of grayscale images was used in this study as a sample table.
- Pixel analysis was carried out side by side based on the defined criteria. If adjacent pixels met the criteria, then the pixels were labeled as 1, while others were given 0.
- Save the pixels labeled 1 in the output image as a region.
- In the image $f(x, y)$, find the pixel labeled 0 and carry out an analysis as in points 2 and 3.
- Save the pixel labeled 0 as a region.

3.1. GLCM Feature Extraction

The sample image was converted into smaller pixels, namely 3x3 with an angle of 0° to facilitate manual calculations. At this stage, the first step was calculating the image matrix and normalizing the results. Furthermore, before normalizing the matrix, the working area was needed, and it was obtained from the function `glcm1 = graycomatrix(Img_gray, 'Offset',[0 1])`. The GLCM extraction stages can be explained in Table 1 [19].

- i. The Quantitation matrix stage: the matrix results obtained were quantized with 8 gray levels ranging from 0-255, as shown in Table 1.

Table 1. Gray value

Pixel Value	Gray Value
0 – 31	0
32 – 63	1
64 – 95	2
96 – 127	3
128 – 159	4
160 – 191	5
192 – 223	6
224 – 255	7

- ii. In the Count Matrix stage, the calculation was carried out in the horizontal direction; hence, the results are presented in figure 9:

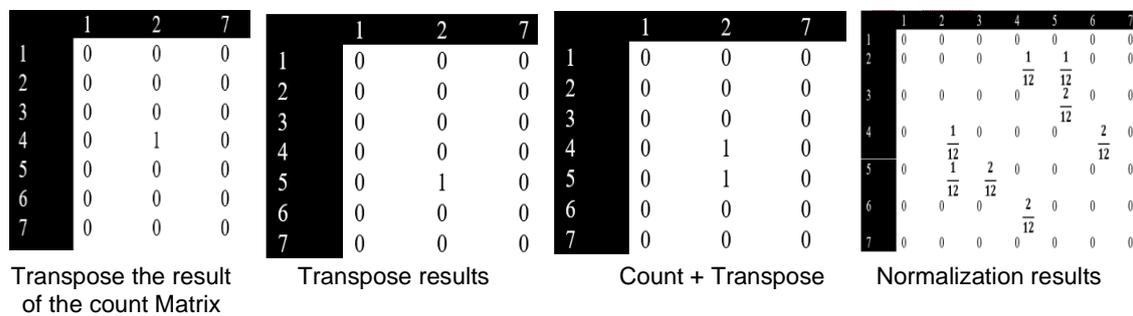


Figure 9. Feature extraction

- iii. The Transpose stage changes the rows in the count matrix to columns.
- iv. Count + Transpose stage. At this stage, the summation between the rows and columns of the count and transpose matrix was carried out.
- v. Normalization Stage. A summation of all the numbers contained in the count and transpose matrix was carried out: $1 + 1 + 1 + 1 + 2 + 2 + 2 + 2 = 12$. After the 3x3 matrix had been successfully normalized, the next step was calculating the features, as shown in Table 2.

Table 2. Results of texture feature retrieval

Feature	0°	45°	90°	135°
Contrast	4.8333	5	0.8333	5.7500
Correlation	-0.3843	-0.6000	0.9467	-0.7745
Homogeneity	0.3194	0.3750	0.5833	0.3333
Energy	0.1666	0.2500	0.2222	0.2500

Table 2 shows the results of retrieving texture features using Matlab functions with all angles in GLCM. The next step was calculating the average of all values from these angles using a function in Matlab. Hence, a single value was obtained from each feature to ease the matching stage.

Table 3. Performance Metrics Comparison for Good and Bad Seeds

Metrics	Good Seeds	Bad Seeds
Contrast	0.2606	0.2465
Correlation	0.9443	0.8900
Homogeneity	0.9424	0.9277
Energy	0.2924	0.3546
precision	0,90	1

Recall	0.04	0
--------	------	---

To describe the performance and analyze the characteristics of textures within an image using these metrics of image processing, such as contrast, correlation, homogeneity, and energy are commonly used texture metrics. Table 3, Illustrates of Cacao images analysis. It provided values representing various metrics used to assess specific data characteristics. In contrast, the value of 0.2606 indicates the measure of contrast within the data. Contrast measures the difference between the light and dark areas in an image. Correlation, with a value of 0.9443, correlation measures the linear relationship between different parts of the data. A higher correlation suggests that the data points have a strong linear relationship. Homogeneity, the value of 0.9424, signifies how uniform the data's texture is. High homogeneity indicates that the texture is consistent throughout the data. Energy: A value of 0.2924 represents the data's energy or "smoothness." Higher energy values suggest smoother data. Precision, a value of 90%, implies that out of the items classified as positive, 90% were indeed true positives. Recall the recall value is 4%, meaning that only a small portion (4%) of actual positives were correctly identified.

Moreover Contrast, the value of 0.2465 suggests a lower level of contrast in this scenario compared to the previous one. Correlation, with a value of 0.890, the linear relationship between different parts of the data is slightly lower compared to the previous classification. Homogeneity, the value of 0.9277, indicates that the texture's uniformity is somewhat lower than the earlier assessment. Energy this time, the energy value is 0.35464, which suggests a bit more variation or "roughness" in the data compared to the previous case. The classification achieved 100% precision, implying that all items classified as positive were indeed true positives. Recall: However, the recall is 0%, indicating that no actual positive instances were correctly identified, which is puzzling given the high precision.

Based on the testing conducted on the query data, where the test was repeated 50 times, the average precision was calculated to be 85.4. This means that, on average, about 85.4% of the positively classified instances were actually true positives. Additionally, the recall was determined to be 0.584. This indicates that a portion of actual positive samples (about 58.4%) were successfully identified through testing. These metrics collectively provide insights into the performance and characteristics of the data and the classification process used.

The table provided an analysis of two sets of seeds, categorized as "good seeds" and "bad seeds," along with their corresponding precision and recall metrics. Good seeds the precision for "good seeds" is 0.9. This means that out of the items classified as "good seeds," 90% were actually true positives. In other words, when the model classified seeds as "good," it was correct 90% of the time. Bad seeds, the precision for "bad seeds" is 1.0. This indicates that when the model classified seeds as "bad," it was correct 100% of the time. Additionally, good seeds, the recall for "good seeds" is 0.04. This suggests that out of all actual "good seeds," only 4% were correctly identified by the model. In other words, the model missed identifying 96% of the "good seeds. Bad seeds, the recall for "bad seeds" is 0.0. This means that the model did not correctly identify any of the "bad seeds" among the actual "bad seeds."

The model has relatively high precision for the "good seeds," indicating that when it classifies seeds as "good," it's quite accurate. However, the recall is lower, meaning that it missed a significant portion of the actual "good seeds. For the "bad seeds," the model has both perfect precision and recall. This suggests that when it identifies seeds as "bad," it is both accurate and thorough in its identification. In practical terms, a high precision is desirable when you want to avoid false positives, as in the case of "bad seeds" where you want to be certain that the classification is correct. On the other hand, a high recall is important when you want to make sure

you're capturing as many of the true positives as possible, as in the case of "good seeds," where you wouldn't want to miss potential positives. In summary, the model performs better in identifying "bad seeds" with high precision and recall. However, for "good seeds," while precision is relatively good, there is room for improvement in terms of recall to capture more of the actual positive instances.

3.2. Calculation of Euclidean Distance and CBIR

Previous research by [20] [21][22] According to Matching is a stage in image processing, which is used to find other similar images. One parameter representing the similarity degree between the samples is the Euclidean distance. The smaller the distance between two images, the higher the similarity. The image calculation was carried out using the Euclidean formula:

$$d(x - y) = \sqrt{\sum_{j=1}^i (x_j - y_j)^2} \quad (1)$$

Where $d(x-y)$ represents the Euclidean distance between x and y vectors in a multi-dimensional space, it quantifies the shortest path length between these two points in the multi-dimensional space. Essentially, it's a measure of how "far apart" the two vectors are from each other. X_i and y_i represent the components (values) of vectors x and y along the j^{th} dimension. Each vector has multiple components in multi-dimensional space, and this equation considers the differences between each corresponding component.

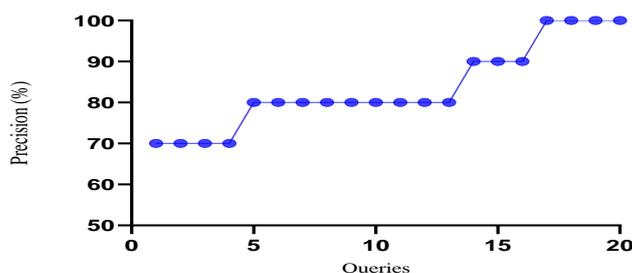


Figure 10. The image retrieval precision

Figure 10 shows an image precision graph matching the query based on the results obtained from the samples. Meanwhile, Figure 14 is a recall graph based on the test results of all query images. The data obtained were compared with the findings of previous studies, which served as references, and several studies related to the determination of quality have been carried out [23][24][12]. The table suggests that the model's optimistic predictions become more accurate as you progress from lower precision values to higher ones. The model performs very well in precision, particularly for instances with precision values of 90 and 100. Based on the precision values provided, it's clear that the model is performing well in terms of accurately identifying positive cases. This could be crucial in applications where false positives are costly or undesirable, such as medical diagnoses or fraud detection. High precision indicates that when the model predicts a positive outcome, it's likely to be correct. Though high precision is generally desirable.

However, it's important to consider whether the dataset has a class imbalance. If the positive class is significantly smaller than the negative class, achieving high precision might be more effortless because there are fewer positive instances to predict. A sudden increase in accuracy from 80 to 90 and then to 100 might indicate a change in the model, data, or problem-solving approach at those points. Investigating these changes could provide valuable insights into improving the model further. Finally, the table indicates that the model is performing well in precision. However, other metrics and factors need to be considered for a comprehensive assessment of the model's performance. Understanding the specific problem, dataset characteristics, and potential biases is crucial for drawing meaningful conclusions.

4. Conclusion and Future Works

Based on the comparison with previous studies, it can be concluded that this study obtained good results with a test accuracy rate of 85.4%. The implementation of the Region Growing algorithm in segmenting objects has been successfully carried out, and the system can separate objects from their backgrounds. The weakness of this study is that the system developed was not completely accurate. Hence, there were several errors during the assessment of image similarities. This was because a filter was not used to remove noise in the sample. Noise has been reported to affect segmentation, which greatly influenced the feature extraction results. This study can be continued by adding new image data types, such as Arabica, Gayo, and Liberika coffee, or by comparing fermented beans from mongooses/civets. Further studies are also advised to use a combination or comparison with other methods, such as the SVM algorithm, Artificial Neural Networks, and others, to compare their accuracy level.

References

- [1] E. D. Martauli, "Analysis Of Coffee Production In Indonesia," *Journal of Agribusiness Sciences*, vol. 01, no. 02, pp. 112–120, 2018.
- [2] B. Raharjo and F. Agustini, "Metode Forward Chaining pada Sistem Pakar Penilaian Kualitas Biji Kopi Berbasis Web," *International Journal of Natural Science & Engineering (IJNSE)*, vol. 4, pp. 73–82, 2020.
- [3] Y. Prastyaningsih, A. Noor, and A. Supriyanto, "Identifikasi Jenis Biji Kopi Menggunakan Ekstraksi Fitur Tekstur Berbasis Content Based Image Retrieval: Identification Types Of Coffee Beans Using Texture Feature Extraction Based On Content Based Image Retrieval," *Computer Science and Informatics Journal*, vol. 3, no. 2, pp. 105–116, 2020.
- [4] J. Bai, D. Qin, P. Zheng, and L. Ma, "Image Retrieval Method Based on Visual Map Pre-Sampling Construction in Indoor Positioning," *ISPRS International Journal Geo-Information*, vol. 12, no. 4, 2023, doi: 10.3390/ijgi12040169.
- [5] J. Jin *et al.*, "Machine learning based gray-level co-occurrence matrix early warning system enables accurate detection of colorectal cancer pelvic bone metastases on MRI," *Frontier Oncology*, vol. 13, no. March, pp. 1–10, 2023, doi: 10.3389/fonc.2023.1121594.
- [6] M. B. Kuyu, M. Meshesha, and C. Diriba, "Grading Ethiopian Coffee Raw Quality Using Image Processing Techniques," *Research Square*, August 2022, doi: 10.35248/0970-1907.23.39.560-566.
- [7] W. M. Kurniawan, "Penentuan Kualitas Biji Kopi Arabika Dengan Menggunakan Analytical Hierarchy Process (Studi Kasus Pada Perkebunan Kopi Lereng Gunung Kelir Jambu Semarang)", *Jurnal Teknik Mesin, Elektro, dan Ilmu Komputer*, Vol. 8, No. 2, Pp. 519–528, 2017.
- [8] M. Olivya, E. Tungadi, and N. B. Rante, "Klasifikasi Kualitas Biji Kopi Ekspor Menggunakan Jaringan Saraf Tiruan Backpropagation," *Jurnal Sains dan Teknologi*, vol. 3, no. 2, pp. 299–308, 2018.
- [9] M. Merzougui and A. El Allaoui, "Region growing segmentation optimized by evolutionary approach and maximum entropy," *Procedia Computer Science*, vol. 151, pp. 1046–1051, 2019, doi: 10.1016/j.procs.2019.04.148.
- [10] C. L. Kang, F. Wang, M. M. Zong, Y. Cheng, and T. N. Lu, "RESEARCH on IMPROVED REGION GROWING POINT CLOUD ALGORITHM," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Arch.*, vol. 42, no. 3/W10, pp. 153–157, 2020, doi: 10.5194/isprs-archives-XLII-3-W10-153-2020.
- [11] H. Jiang, B. He, D. Fang, Z. Ma, B. Yang, and L. Zhang, "A region growing vessel segmentation algorithm based on spectrum information," *Computational and Mathematical Methods in Medicine*, vol. 2013, 2013, doi: 10.1155/2013/743870.
- [12] R. Kaur, "Improving Efficiency of CBIR by using Color, Texture and Fusion Features with Bit Pattern," *International Journal of Computer Applications*, Vol. 178, no. 15, pp. 18–25, 2019.
- [13] V. Lusiana *et al.*, "Ekstraksi fitur tekstur menggunakan matriks glcm pada citra dengan variasi arah obyek 1,2,3," *Seminar Nasional Multi Disiplin Ilmu Dan Call For Papers*, Pp. 978–979, 2019.
- [14] S. H. Jadhav, "Content Based Image Retrieval System With Semantic Indexing and Recently

- Retrieved Image Library," *International Journal of Computer Applications*, vol. 4, no. 1, pp. 40–48, 2020.
- [15] G. T. Situmorang, A. W. Widodo, and M. A. Rahman, "Penerapan Metode Gray Level Cooccurrence Matrix (GLCM) untuk Ekstraksi Ciri pada Telapak Tangan", *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 5, pp. 4710–4716, 2019.
- [16] B. S. V, "Grey Level Co-Occurrence Matrices: Generalisation and Some New Features," *International Journal of Computer Science, Engineering and Information Technology*, vol. 2, no. 2, pp. 151–157, 2012, doi: 10.5121/ijcseit.2012.2213.
- [17] S. Kusi-duah and O. Appiah, "Performance Evaluation of State-of-the-Art Texture Feature Extraction Techniques on Medical Imagery Tasks," *Journal Current Research in Vaccines Vaccination*, vol. 2, no. 1, pp. 1–23, 2023, doi: 10.33140/crvv.02.01.03.
- [18] A. Susanto, "Matematika Citra Digital Untuk Ekstraksi Area Plat Nomor", *Jurnal Pseudocode*, vol. VI, no. 1, pp. 49–57, 2019.
- [19] F. M. Sarimole and A. Syaeful, "Classification of Durian Types Using Features Extraction Gray Level Co-Occurrence Matrix (GLCM) and K-Nearest Neighbors (KNN)", *Journal of Applied Engineering and Technological Science*, vol. 4, no. 1, pp. 111–121, 2022, doi: 10.37385/jaets.v4i1.959.
- [20] N. Rajlaxmi and L. S S, "Content Based Image Retrieval using Spectral Feature Extraction Methods," *International Journal Electronic Communication Engineering*, vol. 2, no. 4, pp. 10–13, 2015, doi: 10.14445/23488549/ijece-v2i4p104.
- [21] S. Hamad, A. Iqbal, S. Naz, N. ul, M. Imran, and B. AlHaqbani, "Content-Based Image Retrieval Using Texture Color Shape and Region," *International Journal Advanced. Computer Science Applications*, vol. 7, no. 1, pp. 418–426, 2016, doi: 10.14569/ijacsa.2016.070156.
- [22] C. Palai, P. K. Jena, S. R. Pattanaik, T. Panigrahi, and T. K. Mishra, "Content-based Image Retrieval using Encoder based RGB and Texture Feature Fusion," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 3, pp. 245–254, 2023, doi: 10.14569/IJACSA.2023.0140328.
- [23] A. Ghita and R. T. Ionescu, "Class Anchor Margin Loss for Content-Based Image Retrieval," 2023, [Online]. Available: <http://arxiv.org/abs/2306.00630>
- [24] S. Kumar, M. K. Singh, and M. Mishra, "Efficient Deep Feature Based Semantic Image Retrieval," *Neural Processing Letters*, vol.55, no.3 January 2023, doi: 10.1007/s11063-022-11079-y.