AN EFFICIENT REAL-TIME ALGORITHM USING SHAPE AND CIELAB COLOR SPACE FOR SORTING COFFEE BEANS

Nguyen Duc An[1], *Bui Quoc Bao*[1], *Tran Hoang Linh*[1], *Hoang Trang*[1]

Abstract

Sorting coffee beans is a crucial stage to achieve high quality and raise the value for the product. This work usually takes a short time to conduct with a large number of coffee beans, while sorting by hand is hard to respond to. And in some cases, appearances of bad coffee beans are nearly similar to good ones, this is hard to distinguish by eyes as sorting in bulk. So an efficient algorithm used particular standards to sort coffee is necessary. From existed issues, this paper presents an efficient approach used as a computer vision system to sort coffee beans based on the criteria about shape and color of the product. Geometric properties and a linear graph are used in this paper to analyze the features of the product. Coffee beans are categorized into two major groups: bad beans and good beans, corresponding to quality standards about specific color and shape. Our proposed method detects and covers the majority of types of bad beans, and get high at both the accuracy metric and F1-score metric with fast speed in sorting.

Index terms

Coffee bean, sorting coffee bean, shape and color for sorting, machine learning for sorting, CIELab color space.

1. Introduction

Coffee is one of the most widely consumed beverages around the world. With 500 billion cups each year, coffee holds the second position in consumption among all beverages after water [1]. And coffee consumption rates have increased by about 2% per year worldwide during the last decades [2]. With a high consumption, coffee is grown in over 80 countries in the tropical and sub-tropical regions of the world, contributing sustainably to their national economies [3]. Coffee flavor is one of the most important quality evaluation criteria employed for coffee commercialization and consumption and affects directly product price [4]. Unfortunately, defects of coffee beans after harvested might damage considerably to the flavor of the product, leading the value of coffee is reduced. Therefore, a pre-processing stage eliminating defect beans of coffee is necessary to reach a better worth for products and raise flavor for the consumer.

[1]Ho Chi Minh University of Technology

The criteria commonly used to evaluate the quality of coffee beans include color, shape, size, density, the number of defects, moisture, etc. [4], [5], and they are also employed in computer vision systems to recognize coffee defects after the harvested stage. A computer vision system might consist of an illuminated system, a camera used to acquire images, a computer, and software for processing input images [6], [7]. Features of beans from images are extracted and analyzed for defect detection of coffee beans, and a classification step is conducted afterward. There have been many studies in the field of sorting coffee beans were proposed. One of the approaches for this problem was represented by Pinto and co-worker [8], authors used a model of deep convolutional neural networks (CNN) to classify coffee beans with six types of different output defects. Although this method obtained high accuracy in distinguishing black and sour beans, it only reaches a proportion rather low in recognizing other coffee colors (72%) and broken coffees (67.5%). Another CNN model was also presented in [9] using for small datasets with high variance by Wallelign. This method divided coffee into 12 quality grades corresponding to 12 different output nodes, however, it also only obtains under 90% rates. Apart from CNN techniques, some other methods are also utilized to evaluate coffee quality. In [7] converted from RGB color space to a CIELAB space using an ANN model, and a Bayesian classifier was combined to classify based on the CIELAB color space converted. The results of this method are shown well with high accuracy in terms of color, but this approach is limited by it did not notice to shape criterion. This one is also similar to Arboleda's method [10], image processing techniques are applied and combined with analyzing RGB color space to identify coffee, however, this proposal also only focuses on identifying black coffee, and can't cover multiple different defects of shape criterion like broken coffee, sour coffee,... Generally, though [7] and [10] all obtained positive results in sorting coffee color, both still can't address thoroughly problems in classifying for multiple different defects, leading to substantial limits in sorting coffee to eliminate bad beans.

From the existed problems mentioned above, this paper proposes an efficient approach to evaluate the quality of green coffee beans with significant contributions. Firstly, our approach achieves high at both the accuracy metric and the F1-score metric. Secondly, our algorithm can detect and cover multiple types of defects of coffee bean which the previous algorithm did not tackle thoroughly. And the last, this method can get a high speed that responds to real-time systems. An image preprocessing stage is conducted to achieve object borders and an RGB color space matrix. The border will be utilized for checking the bean shape, and the RGB matrix is employed to convert to CIELAB color space used for bean color evaluation. A linear chart shows the relation between parameters L*, b*, Hue* in the color space with bean color quality will be analyzed and combined machine learning algorithms to evaluate product color quality afterward.

The paper is organized into three main sections. Section II will represent the materials used in our system, and the details of our proposal are also presented in this section. The achieved results are shown and discussed in section III, and in the final section, we will perform to conclude the proposed approach.

2. Materials and Methods

2.1. Materials

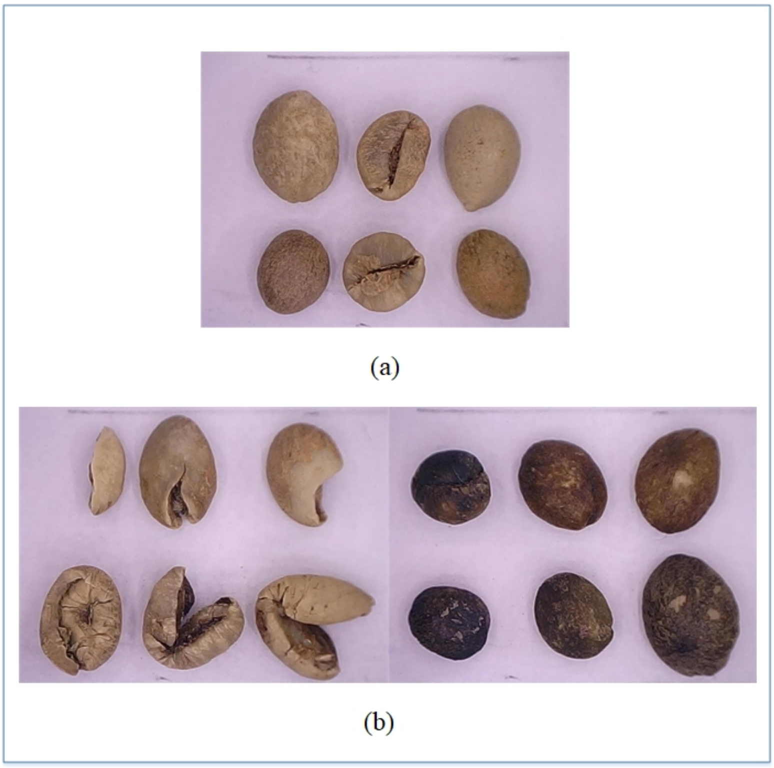


Fig. 1. Two main coffee bean groups.(a) Good bean (b) Bad bean (left: shape criterion, right: color criterion)

Our coffee samples are harvested from different seasons until 2019 and provided by a farm from Lam Dong, Viet Nam. These samples were acquired by a variety of different types of coffee (arabica, robusta,...) that plants commonly in the area. Aiming to group coffee beans into particular groups, we conducted to approach requirements at farms and through previous literature to categorize the dataset into two main groups that rely upon different properties about color grades and shapes of beans. According to National Quality Standards for coffee beans, defects of green coffee might be graded based on many descriptions like black beans, sour beans, shells beans, pulper-cut beans, broken beans, etc. [5]. However, from demands in reality surveyed, the central purpose at farms often only separate coffee beans into two types of coffee beans consists of good beans and bad beans. Bad beans will be eliminated while good beans will be retained after this stage to utilize for various destinations. Based on these realities, our method splits coffee beans into two principal groups: bad beans and good beans that aims to apply researched algorithms in this paper to invent a coffee the sorting machine.

Coffee beans might be evaluated by many different criteria, in which the appearance is one of the most important measures. From appearance criteria, our method distinguishes

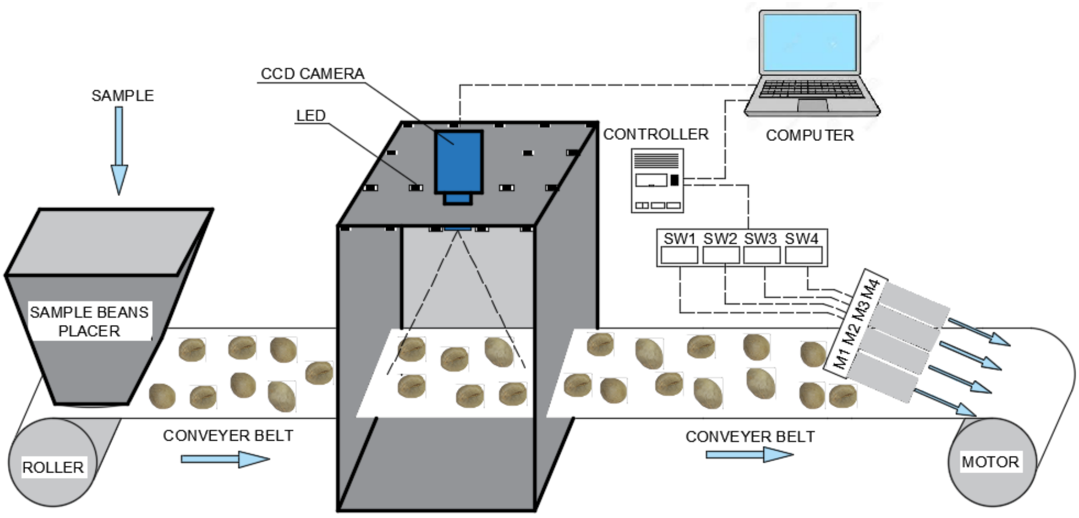


Fig. 2. Our system using to acquire image and sort bean

between bad beans and good beans by using factors about color and shape. A good bean must be ensured that both the color criterion and shape criterion is good. On the other hand, a coffee bean will be treated as a bad bean if any defect about shape or color exists on the bean.

Although there are a variety of defects about color like sour defect beans, black defect beans. . . , but generally the color of these defect beans often are darker than good beans, meanwhile, bad beans by shape are broken beans or deformed beans,... Using these properties to find out whether this is a good bean or not, our approach can cover multiple various defects bean into one group of bad beans. Two main groups consist of good beans and bad beans with kinds of different defect properties is shown in Fig. 1.

To collect the dataset, our system used a white background conveyor belt to move coffee beans through a camera that was placed at 20 cm above the surface in Fig. 2. An illuminated system with 20 single white LEDs is put around the digital camera to provide enough light for taking images. To avoid the distortion by rolling shutter camera when the conveyor belt is moving [11], a CCD digital camera is used in our system with resolution 5 MP, exposure time 1/60s, and ISO 200. The coffee bean samples are dropped along the conveyor belt and will be captured after each particular time by taking multiple objects at once into one image and give the coordinate of bad beans after the processing.

2.2. Our Methods

A flowchart for the entire algorithm is shown in Fig. 3. The images contenting coffee beans are acquired through capturing continuously after each 1/30s by a high-speed camera. Each obtained image with multiple objects shown in Fig. 4a will be segmented by using the Otsu threshold, and the stage eliminating noise will be conducted afterward to give the results represented in Fig. 4b,c. To achieve border pixels used for evaluating the shape of the object, our approach calculated the sum of neighbor pixels aiming to get the first border and used the Aparajeya thinning algorithm [12] to acquire the one-pixel border width as shown in Fig. 4d. Some of the earlier methods are used to detect the defect beans like the CNN method of Pinto [8] and Wallelign [9], or methods using color spaces in [7], [10],... These algorithms generally only provided good results at some type of defects. However, our proposed approach can cover multiple different defects by applying a simple method based on geometry properties to reduces considerable calculation costs, combining with CIELAB color space and machine learning algorithms to evaluate the coffee quality. Our approach utilized three parameters L*, b*, Hue* and analyzed the chart of these parameters to get a threshold based on their linear properties

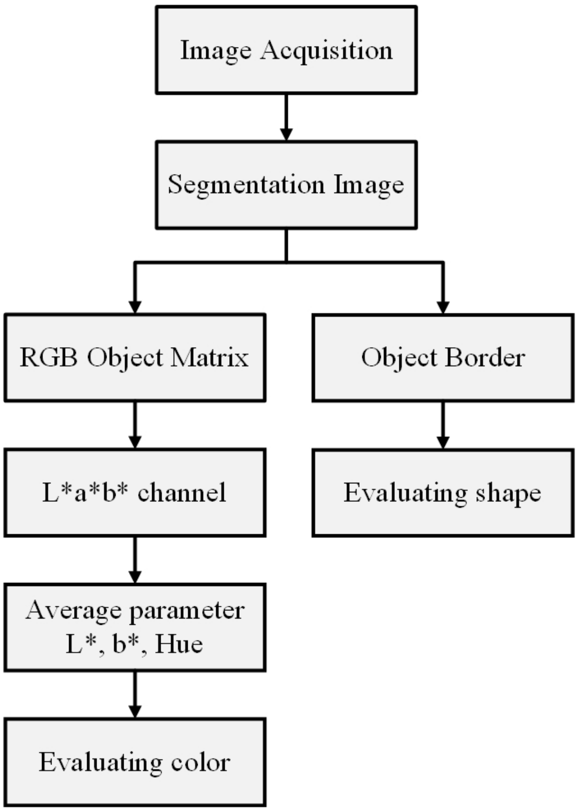


Fig. 3. Our approach flowchart

2.2.1. Object Extraction: A segmented image will be achieved in this section to maintain for the next stages by separating objects from the background. The border pixels and inside pixels of each bean are also archived that aims to use for evaluation with criteria regarding shape and color.

2.2.1.1. Segmentation: Let K is an accomplished automatic threshold after performing the Otsu algorithm [13], and T is the output binary image. The obtained binary

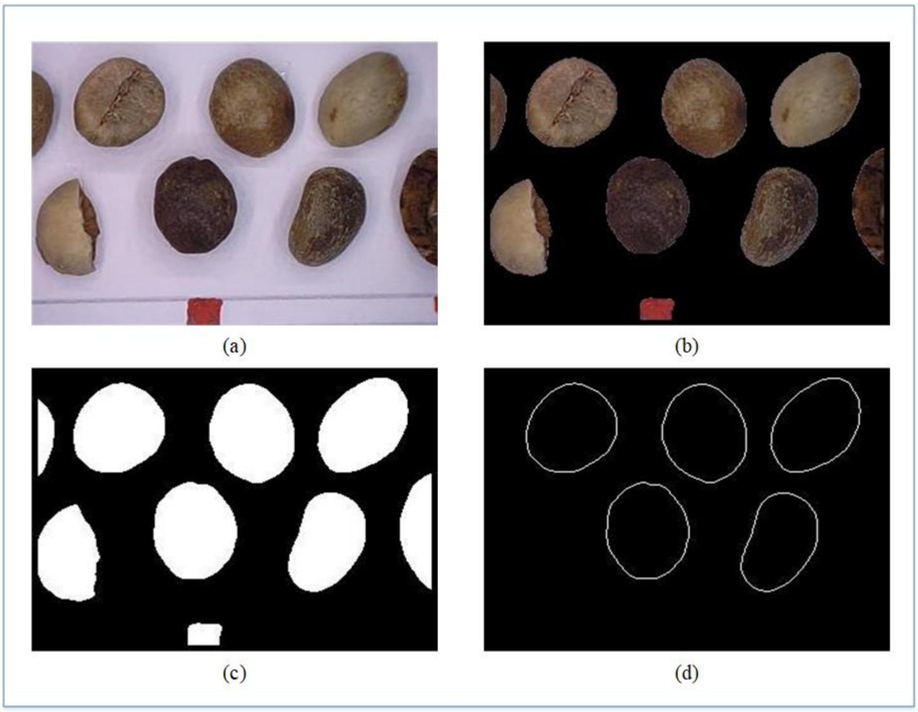


Fig. 4. Pre-processing image to get the border and RBG matrix. (a) Image accquired from camera (b) Segmentation (c) Binary image (d) The border of objects

image is given by

$$T(x,y) = \begin{cases} 1, & \text{if } IMG(x,y) \geq K + \delta \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where δ is the parameter to adjust deviation for threshold if the light from the source is too high or too small, and IMG is an original image acquired from the camera. By the interference from outside light, the acquired image is not homogeneous about light between areas, making the binary image will appear some noise after the binary step. According to our observation, the achieved results exist two types of noise, are salt noise and big holes. The Gaussian kernel was employed to remove salt noise and combined with the Hoshen-Kopelman algorithm [14] to eliminate big holes. The T_G image after eliminated salt noise with Gaussian window is obtained by using

$$T_G(x,y) = T(x,y) * W(x,y) \quad (2)$$

where $W(x,y)$ represents a Gaussian kernel. Basing on the survey through data set shows that the size 3x3 provides better results than using other kernels like 5x5 or 7x7. Fig. 4c shows the image after the binary step.

2.2.1.2. Object border: This stage aims to acquire the object border to check the shape of objects. The achieved border needs to ensure the one-pixel width utilized for evaluating object shape, so our algorithm conducted this stage consists of two steps. Firstly, we get the first border skeleton from objects by calculating the sum of neighbor pixels

$$B(x,y) = \begin{cases} 1, & \text{if } S \leq 7 \text{ and } T_G(x,y) = 1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where S represents the sum of 8-neighbors of the calculated point, T_G is the binary image as calculated above, and B is the border image. And secondly, we used Aparajeya thinning algorithm [12] to obtain the border width to one pixel. Let C_m is the m^{th} object in the image with multiple objects. The positions of pixels on the border belongs the each object C_m is denoted by $BP_m = \left\{(x_{ij}, y_{ij})\right\}_{k=1}^{N_m}$, with N_m is the number of pixels in C_m and k represents the k^{th} pixel in N_m. The value of BP is used for evaluating the shape and extracting each object in the image in the next stages.

After getting the border width to one pixel, broken lines are removed entirely to ensure the obtained image just contains borders of coffee beans. Object borders, which cut by the edge of the image, are also eliminated to ensure only full shapes exist in the image as Fig. 4d. Objects cut by image edge in the previous acquirement will be collected and evaluated in the next acquirement time.

2.2.1.3. Extracting each object from multiple objects image: Let R_{max}, R_{min}, C_{max}, C_{min} corresponding to outermost border coordinates of the objects presented in Fig. 5. A rectangle is used to surround the object aiming to separate the object with others. However, this covering might contain several parts of other objects around. So eliminating other objects residual parts is necessary to keep inside the rectangle only contains the considered object. Hoshen-Kopelman algorithm [14] is used in this step to find the labels contained in the rectangle. The considered object will correspond to the most sum of the label out of existed labels inside the rectangle.

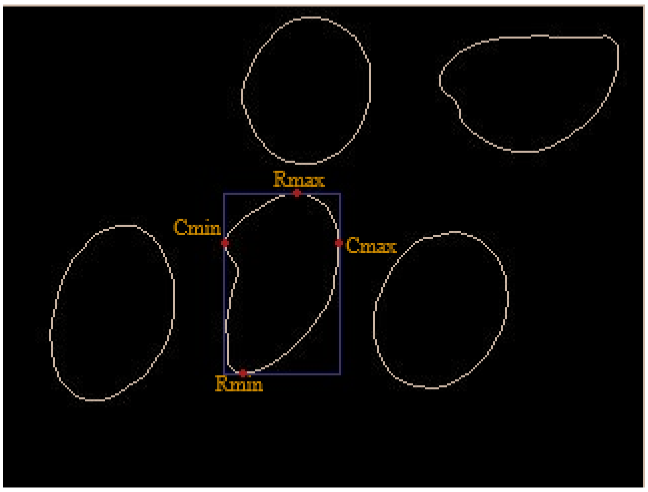


Fig. 5. The object is separated out background by using a rectangle

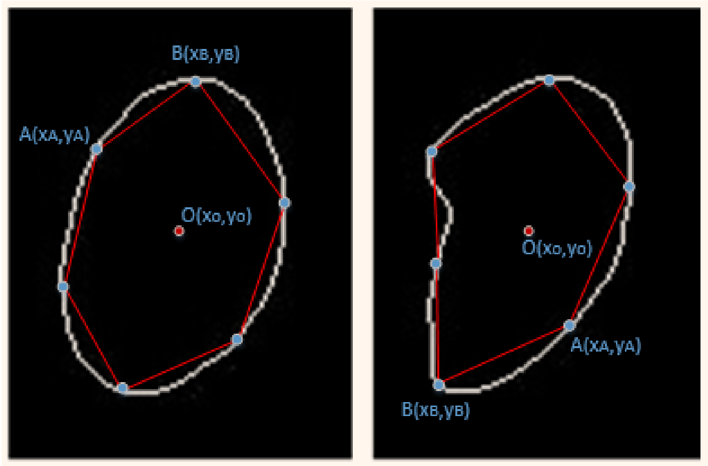


Fig. 6. The object border is seperated into P part (5 parts in our algorithm)

2.2.2. Evaluating shape and color of coffee beans:

2.2.2.1. Evaluating shape: This section represents an approach that appraises beans quality through shape criterion based on the geometry properties. Broken beans or distorted beans (defect beans by shape) usually appears defect positions on the bean surface and make the positions are concave or distorted with the bean surface. Meanwhile, these characteristics rarely appear on the surface of the good beans. Utilizing this property, our algorithm used the object border to explore the defect positions and considered the bean quality based on the number of defects found. If the number of concave positions on the border is greater than a determined threshold, the object will be evaluated as a bad coffee bean. On the other hand, it will be considered as a good bean if the number of concave positions does not exceed the threshold. By using this way, we separated the border of each object into P parts that show in Fig. 6, and evaluating each part relies on the object's cave positions scale.

Let $O(x_O, y_O)$ be center coordinate of object, and $M(x_M, y_M)$ is an arbitrary point in each part in BP_m. The $O(x_O, y_O)$ coordinate is obtained by

$$O(x_O, y_O) = \frac{\sum_{k=1}^{N_m} BP_m}{N_m} \tag{4}$$

where N_m is the number of pixels in C_m as mentioned above. Our approach determined the relative position between O and M with the line AB (A, B corresponds starting and ending point of each part) to calculate whether O and M are the same sides or opposite sides. The line AB in this calculation is represented by

$$f(x, y) = (y_A - y_B)(x - x_A) + (x_B - x_A)(y - y_A) = 0 \tag{5}$$

In the next step, the proposed method used an $r(i)$ array to contain the relative relations in each part of two considered points aiming to find the defect positions.

Value at i^{th} pixel of $r(i)$ get value "1" when O and M are same sides, otherwise, it will get value "0" when O and M are opposite sides

$$r(i) = \begin{cases} 1, & \text{if } f(x_O, y_O)(f(x_M, y_M) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Let thr_{block} is a normalized value of the number of pixels "1" in $r(i)$. This value is obtained by the following formula

$$thr_{block} = \frac{\sum_{i=1}^{N_p} r(i)}{N_p} \tag{7}$$

N_p is number of pixels in each part $N_p = \frac{N_m}{p}$. If the thr_{block} is greater than a threshold THR_{block}, the part of the object is considered not enough criterion, and this is the defect position found in the object. Depending on the level of the defect to evaluate the quality of the bean, our algorithm considers a bad bean if it exists any defected part in the object and give the object coordinate for the next steps

2.2.2.2. Evaluating color using CIE color: The CIELAB color space is an international standard developed by the CIE in 1976. It was considered the CIELAB uniform space in which two color coordinates, a* and b*, as well as a psychometric index of lightness, L^*, were measured [15], [16]. The parameter a^* ranges from green (negative value) to red (positive value), whereas parameter b^* ranges from blue (negative value) to yellow (positive value). And L^* is a qualitative attribute of relative luminosity, which is the property according to which each color can be considered as equivalent to a member of the grayscale, ranging between black (L^* = 0) and white (L^* = 100) [17], [18], [19]. The value of parameters L^*, a^*, b^* is given by

$$L^* = \begin{cases} 116(\frac{Y}{Y_n})^{\frac{1}{3}} - 16, & \text{if } \frac{Y}{Y_n} \leq 0.008856 \\ 903.3(\frac{Y}{Y_n}), & \text{otherwise} \end{cases} \tag{8}$$

$$a^* = [(\frac{X}{X_n})^{\frac{1}{3}} - (\frac{Y}{Y_n})^{\frac{1}{3}}] \tag{9}$$

$$b^* = [(\frac{Y}{Y_n})^{\frac{1}{3}} - (\frac{Z}{Z_n})^{\frac{1}{3}}] \tag{10}$$

where X_n, Y_n, Z_n are the tristimulus values of the reference white point (D65 white point is used in our system), and X, Y, Z are values received from RGB to XYZ conversion.

Another parameter, Hue (h^*), is also calculated in this section combined with other values to analyze a linear relation. Hue is considered the qualitative attribute of color, is the attribute according to which colors have been traditionally defined as reddish, greenish, etc., and it is used to define the difference of a certain color regarding grey

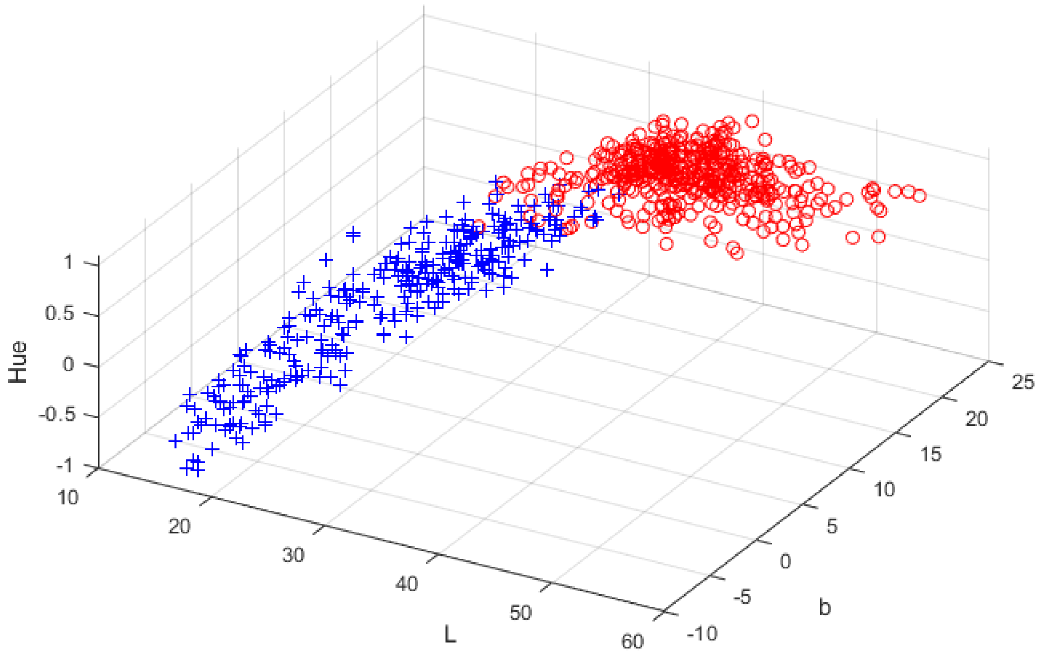


Fig. 7. The two data clusters represents the linear property about color criterion of good beans and bad beans (blue and plus: bad bean, red and circle: good bean)

color with the same lightness. This attribute is related to the differences in absorbance at different wavelengths. Hue is calculated by the following formula

$$h^* = \tan^{-1}\frac{a^*}{b^*} \tag{11}$$

To evaluate the color criterion, we observed the relation between color space and the quality of objects based on parameters together. Although using 1D or 2D space to evaluate the fruit quality might bring good results in several cases, however, the range of coffee bean color is wide and diverse, so 1D or 2D space selections might not cover multiple different color defects. Hence in our algorithm, triple parameter L*, b*, h* are used to appraise the coffee bean quality through a 3D graph. Values of L*, b*, h* in each object are calculated by

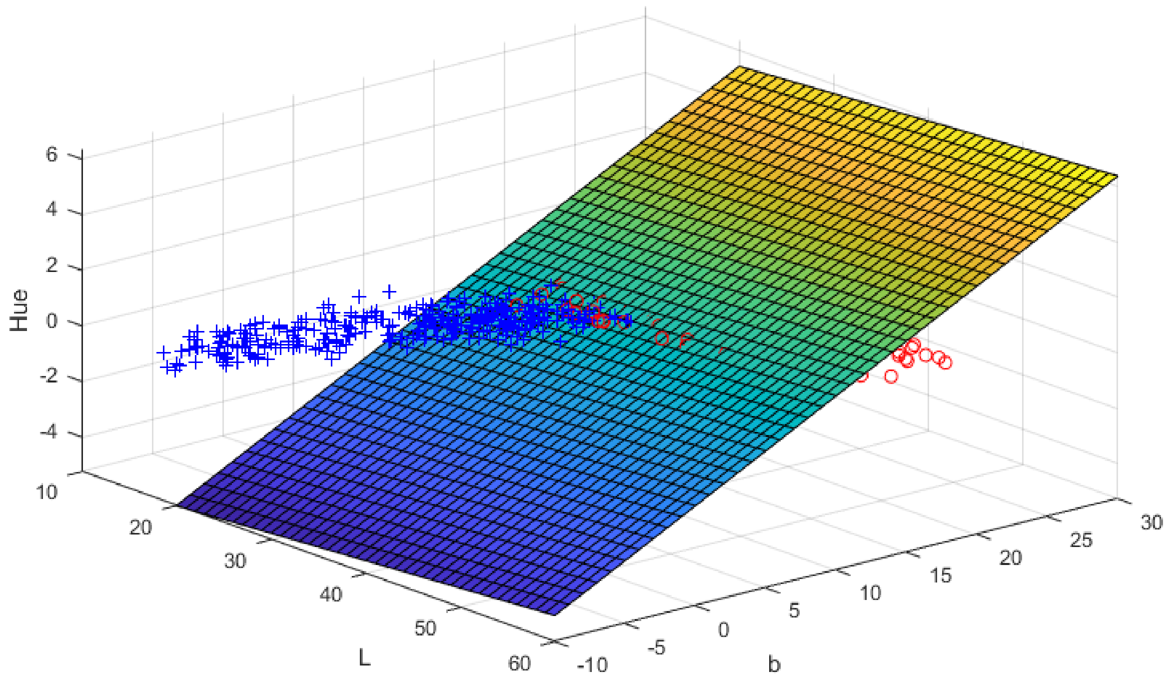


Fig. 8. The plane seprarates two data clusters using pocket algorithm

$$A_{L^*} = \frac{\sum_{i=R_{min}}^{R_{max}} \sum_{i=C_{min}}^{C_{max}} L(i,j)}{N} \tag{12}$$

$$A_{b^*} = \frac{\sum_{i=R_{min}}^{R_{max}} \sum_{i=C_{min}}^{C_{max}} b(i,j)}{N} \tag{13}$$

$$A_{h^*} = \frac{\sum_{i=R_{min}}^{R_{max}} \sum_{i=C_{min}}^{C_{max}} h(i,j)}{N} \tag{14}$$

Here L, b, h represents pixel matrices of L*, b*, h* converted from the formulas above. To analyze the relations of these parameters with the product quality, the proposed approach separated collection into two groups included good beans and bad beans, and label for each group. These data clusters are visualized by the 3D graph drawn through triple parameters as shown in Fig. 7. Red points correspond to the data cluster of good beans while the blue ones are the cluster of bad beans.

From the graph, two data clusters with two different linear characteristics are showed clearly, good beans are clustered separately with color defect beans. However, as presented above, the skin of some types of color defects is similar together, and there are not border clearly between good beans and bad at color criterion in several cases. This leads some noises appeared between two data cluster in the graph and broke the linear property of data. So Pocket Learning algorithm [20] is appropriate in this research to address the problem that aims to find a plane that separates two data classes. This algorithm is a supervised learning algorithm modified from the Perceptron algorithm makes perceptron learning well behaved with non-separable training data, even if that data is noisy and contradictory [21]. Assuming the plane separating two data clusters is given by

$$f_w(L, b, h) = w^T x = w_1 L + w_2 b + w_3 h + w_0 \tag{15}$$

Here $f_w(L, b, h)$ is the output value for classifying two data clusters, $w = [w_0\ w_1\ w_2\ w_3]^T$ is the weight vector and $x = [L\ b\ h\ 1]^T$ is the feature vector corresponding to average values $A_{L^*}, A_{b^*}, A_{h^*}$ calculated above. Let y is the defined label for output data, the label will be attached value "1" if it is a good bean meanwhile a bad bean will be attached value "-1". For each data point x_i, a lost function is represented by

$$e(w; x_i; y_i) = -y_i w^T x_i \tag{16}$$

The weight vector w is updated after each epoch to count the number of misclassified points. If misclassified points currently are smaller than previous epochs, the weight vector will be achieved. This processing is conducted by the number particular epochs and obtained weight vector with the smallest misclassified point. The weight vector w for each data point is updated through

$$w = w - \eta \nabla_w e(w; x_i; y_i) \tag{17}$$

Our algorithm split two data clusters included 450 good beans and 330 bad beans, using 106 epochs and η = 0.1 to train data and find the plane separating two data clusters. The output data will be assigned good bean or bad bean by using the sign function

$$label = sign(w^T x) \tag{18}$$

where

$$sign(w^T x) = \begin{cases} 1 & \text{if } w^T x > 0 \\ -1 & \text{if } w^T x \leq 0 \end{cases} \tag{19}$$

From the acquired weight vector after the training, a plane is found shown in Fig. 8 separated two data clusters into two areas. The accuracy is achieved shown in table 1 with 97.49% and 98.17% for good beans and bad beans.

Table 1. Training data and accuracy at color criterion

Types of bean	Training data	Misclassified data	Training accuracy
Good beans	450	11	97.49%
Bad beans	330	6	98.18%

3. Result

As mentioned above, our algorithm categorizes the dataset into two main groups: bad beans and good beans that are conducted by C language on Asus computer X450LDV (core I3, 1.90GHz). Through the camera, 198 image samples are captured and split according to particular criteria, which consist of 513 good beans and 651 bad beans. A bean will be treated as a bad bean if it exists any defect about color or shape, meanwhile, a good bean must satisfy that don't have any defect on both criteria.

Table 2. The confusion matrix of the proposed algorithm

		Predict	
		Good beans	Bad beans
Actual	Good beans	477	36
	Bad beans	33	618

Aiming to assess the effectiveness of the proposed algorithm, a confusion matrix is used that shows in the table 2 with two main groups, include actual values and predicted values. The outcome shows 36 out of 513 good beans are misidentified, and this number is also low in the bad group with only 33 are misclassified. Utilizing the confusion matrix, two major metrics: accuracy and F1-score are also calculated to measure the effectiveness of the algorithm. The accuracy metric is used to determine how many coffee beans are distinguished true while the F1-score is a balancing metric used for considering the capacity to sort both bad beans and good beans with a trade-off. These formulas are represented as follows

$$Accuracy = \frac{TP + TN}{Total} \tag{20}$$

$$F1\text{-}score = (\frac{precision^{-1} + recall^{-1}}{2})^{-1} \tag{21}$$

where TP stands for true-positive that good beans are predicted as good beans, and TN stands for true-negative that bad beans are predicted as bad beans. Two other metrics are precision and recall are given by

$$precision = \frac{TP}{TP + FP} \tag{22}$$

$$recall = \frac{TP}{TP + FN} \tag{23}$$

with TP already mentioned above, FP stands for false-positive that good beans are predicted as bad beans, and FN stands for false-negative that bad beans are predicted as good beans. Although precision and recall can be utilized to evaluate the algorithm, however, to achieve the number of true classification is highest while still retaining the highest accuracy, F1-score is an appropriate metric for measurement in our algorithm. Table 3 shows the consequences of metrics, with a high accuracy metric of 94.07%, our algorithm is robust in distinguishing exactly bad beans and good beans. Apart from high at accuracy metrics, our approach also ensures the balance of both precision and recall at a high proportion of 93.25%.

Table 3. Measured metric in algorithm

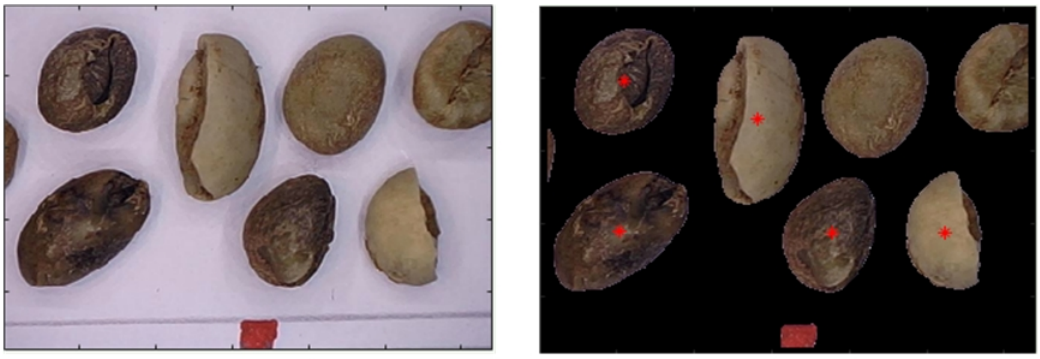
Metric	Percent (%)
Accuracy	94.07
Precision	92.98
Recall	93.53
F1-score	92.25

A good algorithm should ensure to be able to detect multiple defects to achieve high overall accuracy. From accomplished results, our algorithm can achieve a good overall performance that early algorithms did not include. Table 4 shows the comparison between algorithms together. In the table, four methods that consist of Arboleda algorithm [10], Oliveira algorithm [7], Pinto algorithm [8], and our proposed algorithm are compared with the ability at classifying coffee relies on both shape criterion and color criterion of product. Through the table, methods of Arboleda and Oliveira is capacity at detecting defect beans based on the bean color, however, it's not enough to get high at F1-score metric and accuracy metric when it only focuses on sorting for product color, and not noticing that shape is also one of the important criteria to evaluate the product quality. A similar one also appears in the Pinto method, although this algorithm noticed to shape criterion, it only obtains a low proportion in identifying defect beans by shape. In real applications, eliminating entire bad beans is a tough one but it's necessary, our algorithm tackles the problems that previous methods did not resolve, and accomplish high performance at all.

Table 4. Criteria about shape and color in algorithms

Algorithm	Shape criterion	Color criterion
Arboleda algorithm	No	Yes
Oliveira algorithm	No	Yes
Pinto algorithm	Yes (low accuracy at 67.5%)	Yes
Our algorithm	Yes	Yes

Apart from reaching high performance, the proposed approach also achieved results exceeding our expectations in the processing speed of the algorithm. With each image acquired from the camera as shown in Fig. 9a, our algorithm take about 0.03s to process per frame, and this speed can respond to real-time systems well. This is a strength in our algorithm to achieve a coffee machine in reality. The position of bad beans is given to use for the next stages is shown in Fig. 9.



(a) (b)

Fig. 9. The result of algorithm. (a) Image is acquired from camera (b) the position of bad coffee is given with red center

4. Conclusion

In this study, we conducted to classify coffee beans into two groups consisted of bad beans and good beans, bad beans are detected bases on standards about color and shape. A pre-processing is performed that aims to extract the necessary information utilized for evaluating the shape and color of the product. The shape criterion is evaluated relied on geometric properties while color criterion is evaluated based on a combination between CIE color space and machine learning algorithms. The accuracy of over 94% and the F1-score over 93% demonstrated the strength of the proposal. The proposed method can detect the majority of defects of coffee beans to identify the quality of beans (bad or good). Our method also responds to real-time systems when the speed up to about 0.03 frames per second.

Acknowledgement

This research is funded by Ho Chi Minh City University of Technology-VNU-HCM under grant number T-ĐĐT-2019-30.

References

[1] M. S. Butt and M. T. Sultan, "Coffee and its consumption: benefits and risks," *Critical reviews in food science and nutrition*, vol. 51, no. 4, pp. 363–373, 2011.

[2] D. Giacalone, T. K. Degn, N. Yang, C. Liu, I. Fisk, and M. Münchow, "Common roasting defects in coffee: Aroma composition, sensory characterization and consumer perception," *Food quality and preference*, vol. 71, pp. 463–474, 2019.

[3] N. S. Prakash, M.-C. Combes, N. Somanna, and P. Lashermes, "Aflp analysis of introgression in coffee cultivars (coffea arabica l.) derived from a natural interspecific hybrid," *Euphytica*, vol. 124, no. 3, pp. 265–271, 2002.

[4] P. D. M. Agresti, A. S. Franca, L. S. Oliveira, and R. Augusti, "Discrimination between defective and non-defective brazilian coffee beans by their volatile profile," *Food chemistry*, vol. 106, no. 2, pp. 787–796, 2008.

[5] "National quality standards," International Coffee Council, London, United Kingdom, Standard, 2018.

[6] D. Wu and D.-W. Sun, "Colour measurements by computer vision for food quality control–a review," *Trends in Food Science & Technology*, vol. 29, no. 1, pp. 5–20, 2013.
[7] E. M. de Oliveira, D. S. Leme, B. H. G. Barbosa, M. P. Rodarte, and R. G. F. A. Pereira, "A computer vision system for coffee beans classification based on computational intelligence techniques," *Journal of Food engineering*, vol. 171, pp. 22–27, 2016.
[8] C. Pinto, J. Furukawa, H. Fukai, and S. Tamura, "Classification of green coffee bean images basec on defect types using convolutional neural network (cnn)," in *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications (ICAICTA)*. IEEE, 2017, pp. 1–5.
[9] S. Wallelign, M. Polceanu, T. Jemal, and C. Buche, "Coffee grading with convolutional neural networks using small datasets with high variance," 2019.
[10] E. R. Arboleda, A. C. Fajardo, and R. P. Medina, "An image processing technique for coffee black beans identification," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*. IEEE, 2018, pp. 1–5.
[11] E. Ringaby, "Geometric computer vision for rolling-shutter and push-broom sensors," Ph.D. dissertation, Linköping University Electronic Press, 2012.
[12] P. Aparajeya and S. Sanyal, "An effecient parallel thinning algorithm using one and two sub-iteration," in *12th IASTED International Conference on Computer Graphics and Imaging (CGIM-2011)*.
[13] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
[14] J. Hoshen and R. Kopelman, "Percolation and cluster distribution. i. cluster multiple labeling technique and critical concentration algorithm," *Physical Review B*, vol. 14, no. 8, p. 3438, 1976.
[15] D. Granato and M. L. Masson, "Instrumental color and sensory acceptance of soy-based emulsions: a response surface approach," *Food Science and Technology*, vol. 30, no. 4, pp. 1090–1096, 2010.
[16] P. B. Pathare, U. L. Opara, and F. A.-J. Al-Said, "Colour measurement and analysis in fresh and processed foods: a review," *Food and bioprocess technology*, vol. 6, no. 1, pp. 36–60, 2013.
[17] J. R. M. Sandoval, M. E. M. Rosas, E. M. Sandoval, M. M. M. Velasco, and H. C. De Ávila, "Color analysis and image processing applied in agriculture," *Colorimetry and Image Processing*, p. 61, 2018.
[18] D. MacDougall, *Colour in food: improving quality*. Woodhead Publishing, 2002.
[19] I. L. Weatherall and B. D. Coombs, "Skin color measurements in terms of cielab color space values," *Journal of investigative dermatology*, vol. 99, no. 4, pp. 468–473, 1992.
[20] S. I. Gallant, "Optimal linear discriminants," in *Eighth International Conference on Pattern Recognition*. IEEE, 1986, pp. 849–852.
[21] I. Stephen, "Perceptron-based learning algorithms," *IEEE Transactions on neural networks*, vol. 50, no. 2, p. 179, 1990.

Manuscript received: 15-07-2020; Accepted: 15-11-2020.

■

Nguyen Duc An graduated from the HCM City University of Technology in 2019, Currently, he is a Master Student at HCM City University Of Technology. Research Interest: Image processing,AI. Email: ducanbk13@hcmut.edu.vn

Bui Quoc Bao graduated from the HCM City University of Technology in 2003, received a Master Of Engineering Degree from The University Of Leeds, UK in 2010. Currently, he is a PhD Student at HCM City University Of Technology. Research Interest: Embedded System, AI, IoT. Email: buiquocbao@hcmut.edu.vn

Tran Hoang Linh received the B.S. degree in Electrical and Computer Engineering from University of Illinois, Urbana – Champaign (2005), M.S. and PhD. in Computer Engineering from Portland State University (2006, 2015). Currently, he is working as lecturer at Faculty of Electrical-Electronics Engineering, Ho Chi Minh City University of Technology. His research interests include low power, high speed integrated circuit design, quantum/reversible circuit and data-mining. Email: linhtran@hcmut.edu.vn

Hoang Trang was born in Nha Trang city, Vietnam. He received the Bachelor of Engineering, and Master of Science degree in Electronics-Telecommunication Engineering from Ho Chi Minh City University of Technology in 2002 and 2004, respectively. He received the Ph.D. degree in Microelectronics-MEMS from CEA-LETI and University Joseph Fourier, France, in 2009. From 2009–2010, he did the postdoctorate research in Orange Lab-France Telecom. Since 2010, he is lecturer at Faculty of Electricals–Electronics Engineering, Ho Chi Minh City University of Technology. His field of research interest is in the domain of FPGA implementation, Speech Recognizer, IC architecture, MEMS, fabrication. Email: hoangtrang@hcmut.edu.vn

MỘT THUẬT TOÁN THỜI GIAN THỰC HIỆU QUẢ SỬ DỤNG HÌNH DẠNG VÀ KHÔNG GIAN MÀU CIELAB CHO PHÂN LOẠI HẠT CÀ PHÊ

Tóm tắt

Phân loại cà phê là một giai đoạn để đạt nâng cao chất lượng và nâng cao giá thành cho sản phẩm. Công đoạn này trong thực tế thường được thực hiện thông qua cách phân loại thủ công bằng tay. Điều này dẫn đến sự kéo dài về thời gian trong công đoạn khi xử lý, trong khi lượng cà phê mỗi lần cần phân loại thường lớn. Ngoài ra, trong quá trình phân loại bằng tay này, một số loại hạt cà phê khiếm khuyết có thể bị bỏ xót khi phải phân loại lượng lớn hạt bằng mắt thường. Vì vậy, một thuật toán hiệu quả sử dụng những tiêu chí cụ thể nhằm đánh giá chất lượng để phân loại hạt là một điều cần thiết. Từ những nhu cầu cấp thiết này, chúng tôi đề xuất một phương pháp sử dụng hệ thống Thị giác máy tính để phân loại hạt dựa trên các tiêu chí về hình dạng và màu sắc của sản phẩm. Những tính chất hình học và một biểu đồ tuyến tính được sử dụng trong phương pháp của chúng tôi để phân tích những đặc trưng của hạt cà phê. Các hạt này sẽ được chia làm 2 nhóm chính gồm cà phê tốt và cà phê xấu dựa trên các tiêu chí về màu sắc và hình dạng cụ thể. Phương pháp của chúng tôi có phát hiện phần lớn các loại hạt xấu và đạt một tỉ lệ cao ở cả thông số accuracy và F1-score với một tốc độ cao trong phân loại.