Quality Classification of Cross-cut Cacao Beans using K-Nearest Neighbor Algorithm Vision Learning and Sale Price Calculator

Kevin A. Mendi[1], Angel Vonn A. Capuyan[2], Ma. Paula R. Estrada[3]
[1-3]Computer Engineering Program, College of Engineering Education, University of Mindanao
Davao City, Philippines
[1]kevin.mendi20@gmail.com, [2]vonnyanyan@gmail.com, [3]mpe22gal@gmail.com

***Abstract*— One of the factors taken into consideration when making chocolates is the quality of the beans based on its fermentation level. This device aims to classify the quality of the cross-cut beans after the fermentation and drying process is done. It uses image processing technology to classify an image of the beans and determine the classification of the beans: well-fermented, under-fermented or 'purple', over-fermented, slaty, and moldy. After determining the quality of the beans, the device also calculates the sale price for the beans based on a standard price matrix.**

***Index Terms*— Image Classification, OpenCV, Image Processing, Agricultural Technology, Color Classification**

I. Introduction

The Philippines is one of the producers of Cacao globally and was even able to produce 20% of the world's cocoa at one point in the 1990s. (Tupas, 2014) The country's climate and soil quality makes for a suitable environment for the Cacao farms to grow different varieties of cacao and the right weather conditions needed for the post-harvest process that would prepare the cacao for chocolate production. According to the Bureau of Agricultural Statistics in 2008, 80% of the Philippines' total cacao production comes from Mindanao, and 66% of the total is from the Davao region. (Peace and Equity Foundation, 2016)

The Philippines still has a long way to go when it comes to the global cacao trade, however, since the growth and development of the industry is held back by various problems such as fragmented market linkages, unavailability of more land for expansion, low farmer awareness and expertise on the proper practices of cultivation, and insufficient technology. (Peace and Equity Foundation, 2016)

In the process of chocolate making, the quality of the beans used would have to be controlled and should be able to pass the set standards to produce chocolates with just the right consistency and flavor. From the tree, cacao beans go through various stages and processes to ensure that they would produce the right combination of flavors to make the chocolate bar. One of the processes that the beans go through in the initial stages is the fermentation of the beans. This is a crucial stage as this is when the beans develop the flavors needed for the chocolate. However, during this process there would still be under fermented beans. The next process would then be to check whether the amount of under fermented beans in the batch are way over the amount suggested in the standards set to produce the right flavors for the chocolate. This process is usually done by manually checking the color of the cross-cut cacao beans.

Figure 1. Crossed-Cut Cacao Beans: Well-Fermented (Brown), Under-Fermented and Slaty (Purple)

The fermentation level of the beans can be determined by its color and the cracks developed by the bean. As seen in figure 1, well-fermented beans are brown in color and have well developed cracks due to the drying of the developed cotyledon before it was roasted. Under-fermented beans however are purple in color but may still have cracks. Slaty beans, on the other hand, are also purple in color but may not have cracks unlike the under fermented beans, this could be caused by harvesting and fermenting the pods before it has fully ripened. There are also the cases of deformed and moldy beans which are the really defective ones characterized by their non-oval shape – some of these are two beans that somehow ended up getting stuck together during the fermentation process.

The manual process of identifying the beans is done by judgment of the eyes of the one doing the classifying. The beans are crossed-cut, placed on white boards and classified in batches that are composed of 100-gram samples, as shown in figure 2. The number of the beans are counted and recorded – this would be the basis of the size of the beans; the lesser the number of beans per 100-gram sample the better. The next step is to then identify which of the beans are well and under fermented. This is done by checking for the color and the cracks on the beans. The next step would be to count the number of deformed and moldy beans are in the sample. The numbers taken from the identified beans are then taken into consideration and the percentage of each of these classifications calculated to get the overall rating of the beans based on the matrix of standard of the chocolate maker. The matrix is then used to determine the price to be given to the farmers for the beans sold.

Figure 2. Manual process of classifying the quality of crossed-cut cacao Beans; crossed-cut beans are laid out on boards for inspection

However, the current method presents certain risks as not all things are immediately visible to the eyes. And the individual colors of the beans are too close to either purple or brown that there are times when it is indeed difficult to tell the colors apart by sense of sight alone. There are also the cases of biases when doing the classification of the beans. Considering the fact that we are humans, we tend to have our judgment easily swayed by our emotions, biases and subjective views which may affect the way one classifies the beans. There are times when the one classifying the beans would give the farmer the benefit of the doubt and just marks what should've been a substandard bean as standard and prices them as such. This doesn't just present a bias on the financial dealings of the chocolate maker but also on the quality of the beans they are purchasing. Also, some under fermented beans are classified as well fermented if the purple shade in the fruit is only up to 30% of the bean's color and the rest of the 70% is already brown. Proper identification of the ratio of purple to brown of the beans is a challenge when it's done manually.

These may not seem like much of an issue for some, but when it comes to keeping to the standards and the quality of the beans to be purchased, it in itself presents a problem.

II. Objectives of the Study

This study aims to provide a solution for a faster and more effective way of classifying quality of cross-cut cacao beans, and thus, answers the following problems:

1) Design a system that would classify the fermentation of Cacao Beans through image processing using a specific image processing algorithm
2) Create a system that automates the pricing of the cacao beans based on the results of the image analysis.
3) Conduct a functionality test.

III. Scope and Limitations

This study focuses only on identifying and classifying the fermentation level of the cacao beans through image processing and designing a software and hardware that would make it possible to do so. It is designed to only hold up to 100-gram of cacao bean samples and will identify whether the samples' color attributes fits that of well, under and over fermented cacao beans.

IV. Materials and Methods

This section shows the necessary procedures done to manufacture the hardware device and develop the system for the proposed study.

A. Design Procedures

The researchers followed the prototype life cycle model upon conducting the study and developing the software of the system. For this development model, the researchers had to plan the objectives of the study and prepare the details beforehand so that the researchers could identify the goals they need to achieve. For this study, the researchers were able to recognize and list down the initial objectives of the study and narrowed it down to three main objectives. The researchers then identified the list of additional data that they would need for the system and worked on getting said data needed. Once the researchers were able to determine a source where they could get the information they need, they then proceeded with the research on how to achieve the results they need and how to make use of the information and data that they have on-hand. Their research yielded a variety of algorithms, methods and processes that they can try out. The researchers assembled and put together the hardware device and tried the algorithms out to compare which would yield the kind of results that best serves the researchers' intentions. One at a time, source codes were prepared for each algorithm; starting with the one that would get the closest results according to the researchers' inquiries. After developing the source code for each algorithm and the data from the results taken into account, the researchers then proceeded with implementation and experimentation. Errors and any amendments in the system and the algorithm were considered and the code revised. After deciding on the best algorithm to use for the system, the researchers then proceeded with collecting data from the 100 batches of samples to train the device. This training process have been done to set-up the data that the system would use to properly and correctly classify the beans once the system is done. Once the training process was over, the system then underwent testing to check its functionality. Once all revisions were made and all the needs for the system were met, a final test is conducted to ensure full functionality and the percentage of error of the system.

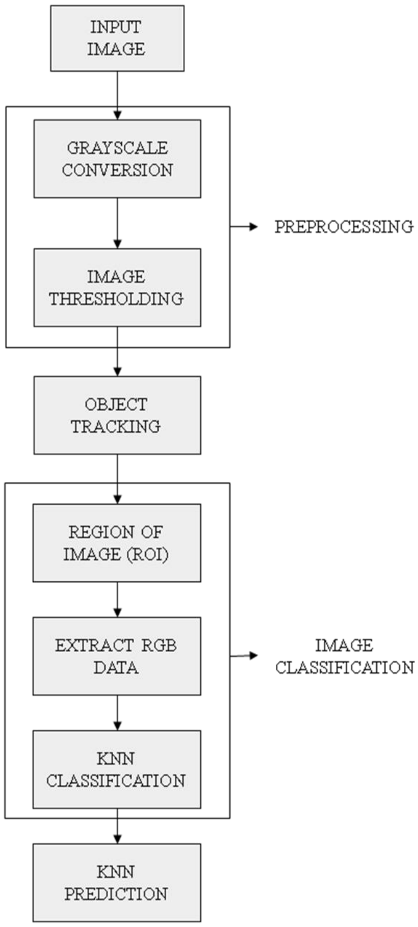

Figure 3. Data Flow Diagram

A data flow diagram was also developed by the researchers, as shown in figure 3, to better understand the flow of data throughout the system. The data flow diagram is designed with regard to the database of the system.

To be able to use the device and utilize its features, the user would have to set-up their own profile in the system to gain authorization to use the program itself. Once access to the system is enabled, the user can then analyze and calculate prices using the system, and access stored data from previous transactions and trades. Any transactions made with the system are recorded and matched to the user logged into the system at the time of the trade. All of the information taken are then stored to the system's database.

The system's image classifier was programmed with OpenCV and Python, while the interface was configured and designed using VisualBasic.NET. During image classification, the system transmits a message to the camera telling it to capture images and the camera sends the pictures back to the system or the program. Once the images are into the system, the classifier then analyzes them and compares the data to that of the data already in the database. Results of the classification and the sale price are then displayed for the user to see and the data, as well as the images and the results, are stored into the system with records as to who was logged in, who the trader was, the time of the trade and the price calculated for the said transaction.

B. SystemsDesign

The researchers designed the system flowchart which was the basis for the construction of the software as shown below. Figure4.2 shows the system flowchart for the design. The figure below shows the flow of process of the software.

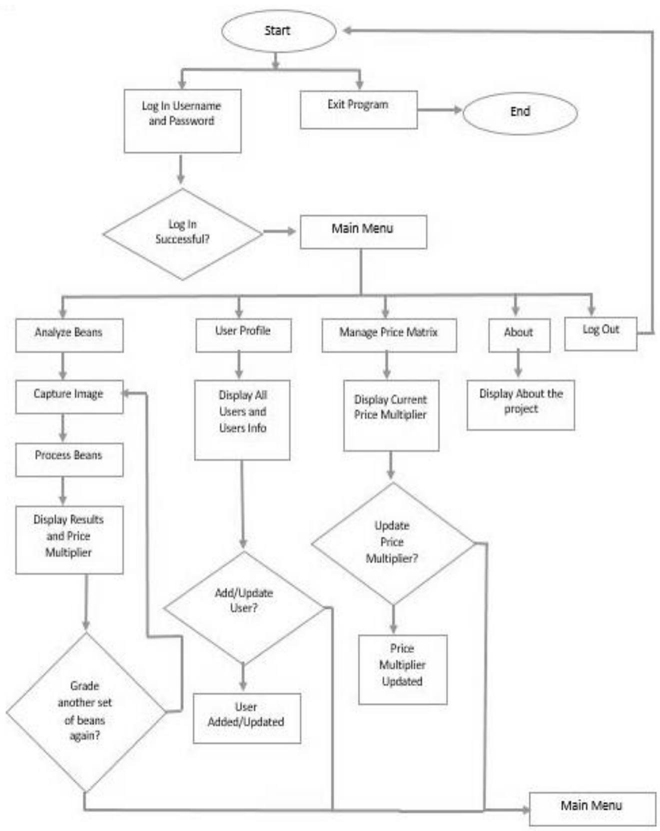

Figure 4. System Flowchart

The process of the whole software system shown in figure 4 is the software flowchart. After the user has logged in, Main menu will display with five available options for the user to select. First option is Analyze Beans which is the main function of the system. The system allows user to capture image of cacao beans from the chamber. The acquired image will be processed by the program. The device then detects which beans are well-fermented, under-fermented, moldy and deformed. The second option is the User Profile, the authorized user can add, update, and delete user information in the system. The third option is the Price Matrix, the user can view and update the price matrix especially if the global price of Cacao beans moves. The fourth option is the About button, when clicked, the user can view information about the project. The last option is the Logout button, when clicked, the user will be directed to the Login form.

V. Results and Discussions

A. Actual Image Analysis Result

Figure 5a. Original Image captured using the device

The device captures an image of the bean sample, as shown in figure 5a, and saves the image to an internal storage system. The camera is specifically set so as to control the image quality and make the image dynamics as uniform as possible.

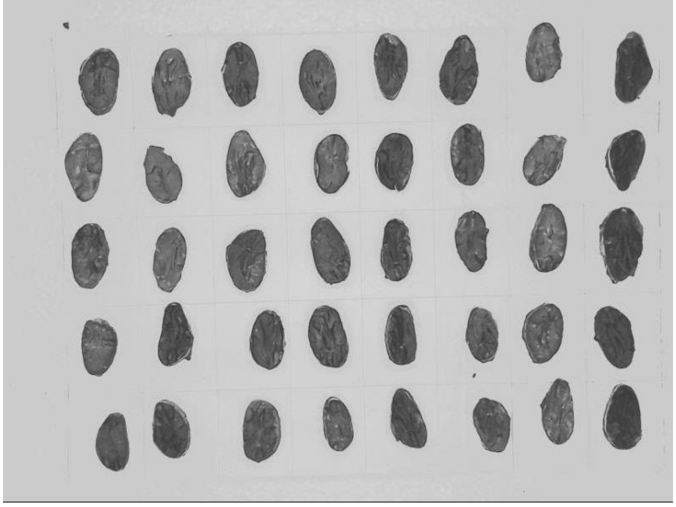

Figure 5b. Original RGB image converted to grayscale

Next, the captured image or the original RGB image is converted to greyscale, shown in figure 5b. This is to make it easier to do the thresholding process on the image. Converting the RGB image to grayscale lessens the pixel variations that the threshold value needs to be compared to. With the RGB values of the original image, there would be too much factors to consider upon determining the threshold value. But when the researchers convert the 24-bit RGB value of the image to its 8-bit grayscale counterpart, it is easier to set a threshold value that the researchers can compare the image’s pixels and lesser number of bits to work on in the image therefore making the thresholding process faster and easier.

Once the image has been converted to its grayscale equivalent, the image is now ready for thresholding. Image thresholding is a segmentation method in image classification. A threshold value is determined according to the needs of the programmer and is used as the basis or comparison of each pixel intensity value in the image which separates the pixels that the researchers want to examine further. Once the important pixels are separated or identified, the researchers can then set them with a determined value to identify them later on. The value assigned to the pixels could range from 0 (black) to 255 (white). The principle of thresholding is simple, if pixel value is greater than a threshold value, it is assigned one value (may be white), else it is assigned another value (may be black).

For the purposes of the study, the researchers used two thresholding techniques on the image: the inverted binary threshold and the Otsu’s binarization. The inverted binary threshold compares the pixels to the threshold value and assigns a set value to the pixel if it is less than the threshold value. Otsu’s binarization, on the other hand, determines the threshold value that is to be used to compare the pixels to. Both techniques need to be applied together. Figure 5c shows the results after both thresholding techniques are applied.

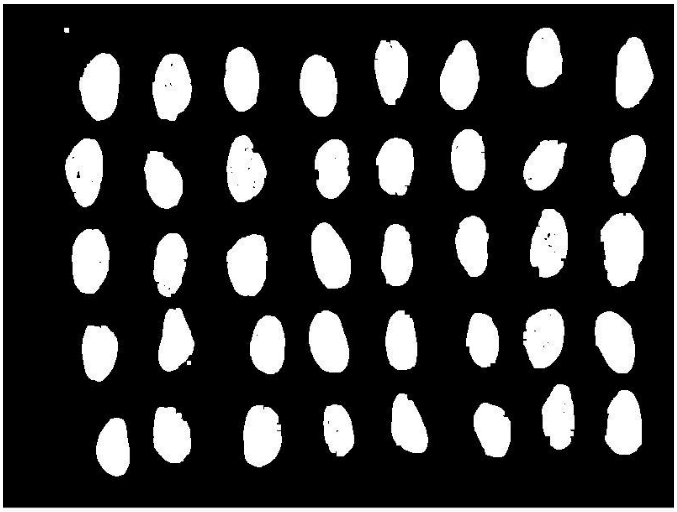

Figure 5c. *Resulting image after Image Thresholding (cv2.thresh_binary_inv + cv2.thresh_otdu)*

After image thresholding, the resulting image would now feature a clearer shape of the objects in it, which in the case of our study are the cacao beans. With the new state of the image, it is now easier for the program to identify the area of each object or bean.

Once image thresholding is done, it is now possible to track the objects in the image. In the process of object tracking, a bounding rectangle is applied on every object or bean detected in the image. These rectangles set the value of the area of the objects or the cacao beans. In technical terms, this process could also be tagged as finding the contours of the object. After the areas of the objects are set, the ROI or region of the image is then taken. This value is based on the 4 end-points of the bounding rectangle for each object in the image.

Canny detection is also applied on the image. The average number of cracks per object is then extracted and the data is stored.

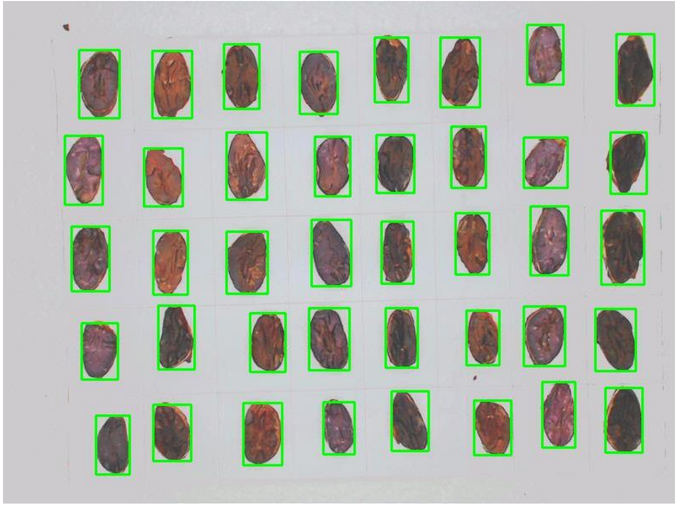

Figure 5d. Original Image with bounding rectangle

The ROI value taken from the bounding rectangle are then used on the original image to get the average color (RGB) based on the region enclosed by the bounding rectangle, shown in figure 5d.

After extracting the features from the RGB image, the values are then applied to the KNN classifier. The classifier then analyses the data from the captured image with respect to the system's database. Once it is done with comparing the data, the prediction will then determine the number of well-fermented cacao beans in the image, as well as the number of under-fermented beans, over-fermented beans and moldy or defective beans.

B. Confusion Matrix

TABLE 1
TRUTH DATA

	C1	**C2**	**C3**	**CO**	**PA (Precision)**
C1 (WF)	73	0	1	74	98.649%
C2 (UF)	0	27	5	32	84.375%
C3(OF)	0	1	13	14	92.857%
Truth Overall	73	28	19	120	
User Accuracy (Recall)	100%	96.429 %	68.421 %		

Overall Accuracy (Observed Accuracy)	94.167 %
Kappa[1]	0.893

Legend:
C1 = Class 1 – Well-Fermented Beans
C2 = Class 2 – Under-Fermented Beans
C3 = Class 3 – Over-Fermented Beans
CO = Classification Overall
PA = Producer Accuracy

ACKNOWLEDGMENT

The researchers would like to acknowledge the University of Mindanao – Matina Campus, specially their adviser, Engr. Randy E. Angelia, who gave guidance, constructive criticism and support – financially and emotionally, to the researchers in order for this study to be complete.

REFERENCES

[1] Cocoa World Foundation (2015), History of Cocoa, retrieved fromhttp://www.worldcocoafoundation.org/about-cocoa/history-of-cocoa/retrieved on May 31,2017

[2] CIDAMI (2017),About Cacao, retrieved from http://www.cidami.org/about-cacao/retrieved on May 29,2017

[3] Peace Equity Foundation (2016), A Primer on PEF's Priority Commodities: Industry Study on Cacao, retrieved from http://pef.ph on March 26,2017.

[4] ECA-Caobisco-FCC Cocoa (2015), Cocoa Beans: Chocolate and Cocoa Industry Quality Requirements

[5] Bradly (2017),Flavors of Cacao, retrieved from http://flavorsofcacao.com/flavor.html retrieved on May 15,2017.

[6] Reyes, J., Bohorquez, J., Alama, W. (2015) Hyperspectral Analysis Based Anthocyanin Index (ARI2) during Cocoa Bean Fermentation Process,Vol. 00, no. , pp. 169-172, 2015, doi:10.1109/APCASE.2015.37

[7] Guehi, T. (2010) Effect of Turning Beans and Fermentation Method on the Acidity and Physical Quality of Raw Cocoa Beans. Advance Journal of Food Science and Technology 2(3). Retrieved from http://maxwellsci.com/print/ajfst/v2-163-171.pdf

[8] Klette, R. (2014). Concise Computer Vision. London: Springer. DOi: 10.1007/978-1-4471-6320-6

[9] Szeliski, Richard. Computer vision: algorithms and applications. London: Springer, 2011. Print.

[10] Tupas, J. (2014). Mindanao typhoon victims' futurelooks sweet.http://www.ucanews.com/news/mindanaotyphoon-victims-future-looks-sweet/71953.

[11] Jabbar, M., Deekshatulu, B., & Chandra, P., (2013). Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm. Retrieved form https://arxiv.org/ftp/arxiv/papers/1508/1508.02061

[12] Imandoust, SB., Bolandraftar, M., (2013). Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background. Retrieved fromhttp://www.ijera.com/papers/Vol3_issue5/DI35605610

[13] Khamis, HS., Cheruiyot, K., & Kimani, S., (2014). Application of k-Nearest Neighbour Classification in Medical Data Mining. Retrieve fromhttps://www.researchgate.net/publication/270163293_Application_of_k-Nearest_Neighbour_Classification_in_Medical_Data_Mining

[14] Kumar, S., (2009). Canny Edge Detection Implementation on TMS320C64x/64x+ Using VLIB . Retrieve fromhttp://www.ti.com/lit/an/sprab78/sprab78

[15] Agaian, S., Almuntashi, A., (2009). An Improved Canny Edge Detection Application for Asphalt Concrete. Retrieve fromhttps://pdfs.semanticscholar.org/7881/f977b456ec75d60362f8818db88912eb8f82.pdf

[16] Chandrashekar N.S., and Nataraj K.R., (2012). A Distributed Canny Edge Detector and Its Implementation on FPGA. International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 7.

[17] School of Mathematical Sciences. (n.d). Retrieved Febuary 2018, fromhttp://www.math.tau.ac.il/~turkel/notes/threshold

[18] Bradski, G., Kaehler, A., (2008).LearningOpenCV. Retrieve from http://www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV

[19] Vala, M.H., Baxi, A., (2013). A Review on Otsu Image Segmentation Algorithm. Retrieved from https://pdfs.semanticscholar.org/ee8b/aed71d46b4d8747c8f2c73ba73f7c4a06130

[20] OpenCV 2.4.13.5 documentation, (2018). Basic Thresholding Operations. Retrieved from https://docs.opencv.org/2.4/doc/tutorials/imgproc/threshold/threshold.html