Image-based Automatic Counting of *Bacillus cereus* Colonies using Smartphone

Phongsatorn Taithong[1], Siriwan Wichai[2], Rattapoom Waranusast[3], Panomkhawn Riyamongkol[4]
Faculty of Engineering, Naresuan University, Phitsanulok City, 65000, Thailand[1, 3, 4]
Faculty of Medical Science, Naresuan University, Phitsanulok City, 65000, Thailand[2]

***Abstract*—Substantial amounts of *Bacillus cereus* bacteria present in food indicates that the food is unsafe to eat, so counting B. cereus colonies in food samples is a common test for food cleanliness. Manual counting of B. cereus bacteria colonies requires approximately 2-5 minutes per Petri dish, depending on the number of colonies present. This study presents a new smartphone-based method called Bacillus Cereus Image Counting System (BCICS, "B. kiks") for automatic counting of B. cereus colonies. BCICS uses image processing techniques including Projection Profiling, Circle Hough Transformation, Adaptive Thresholding, and Power-Law Transformation to achieve high image clarity and then uses the Connected-Component Labeling (CCL) technique to correctly count the colonies, including overlapping colonies. These techniques are built into a convenient Android smartphone application. Results of counting the colonies with BCICS were compared with results of hand counting the same dishes. The accuracy rate of each dish count was calculated, as well as the average dish accuracy across all dishes. BCICS counted total colonies with an accuracy of 90.14%, which is close to that of hand counting accuracy since hand counting itself commonly involves an error rate of 5-10%. Importantly, the application took only 3-5 seconds to count one Petri dish, which is more than 74 times faster than the time required for manual counting.**

Keywords*—Bacillus cereus bacteria; colonies; automatic counting; android phone application; image processing

I. Introduction

Bacillus cereus, or B. cereus, is an aerobic spore-forming bacterium that naturally inhabits soils and plants [1,2]. B. cereus is one of the most important foodborne pathogenic bacteria, and it can cause two types of food poisoning. The first, emetic syndrome, is characterized by nausea and vomiting and occurs within a few hours of consumption of contaminated food. The second, diarrheal syndrome, is characterized by abdominal pain and diarrhea [3] and occurs within 3-4 hours of consumption of contaminated food. Besides B. cereus, other members of the Bacillus group of bacteria can cause food poisoning outbreaks through consumption of raw fruits and vegetables, and this is a major concern in food safety [4-6].

B. cereus food poisoning most commonly occurs when prepared foods are left to sit without adequate refrigeration for several hours before serving. B. cereus concentrations reaching approximately 10^6 cells/gram of food will typically provoke abdominal pain and bloody diarrhea. There is an incubation period of 4-16 hours following ingestion, with symptoms lasting for 12-24 hours [4]. In Slovakia, B. cereus colonies in pasteurized milk at 30°C are not allowed to exceed a total dish count of 3×10^4 CFU/ml. When pasteurized milk is incubated for 5 days at 6°C, the total dish count must not exceed 5×10^4 CFU/ml. [7].

The Standard Petri Count (SPC) method is a conventional technique used to examine Petri dishes for microbiological growth that is large enough to be seen with the naked eye or with a magnifying glass. SPC can be used to test milk quality. To do this, a sample of the milk is diluted with distilled water and then spread onto a Petri dish that contains a medium where bacteria present in the food will start to grow. Once a Petri dish is inoculated and incubation begins, the bacteria colonies will usually be apparent and ready for counting within 24-48 hours [8-9]. When only a few colonies of B. cereus are found, this indicates that the milk meets common quality standards. Colony counting normally requires physically seeing and tallying the colonies by hand, and the accuracy of the total count for a dish can vary from one counter to another, because the B. cereus colonies can be numerous and appear in many sizes. An experienced counter can count the colonies in one Petri dish in approximately 2-5 minutes, depending on the number of colonies present in the dish.

The current study introduces a new method of B. cereus colony counting in which an image is taken of the Petri dish to be counted. The image goes through a series of image processing techniques, including Color Transformation, Thresholding, Projection Profile, Circle Hough Transform, Adaptive Thresholding, and CCL, which together count the colonies automatically. These procedures are built into a convenient Android smartphone application called Bacillus Cereus Image Counting System (BCICS, "B. kiks") that substantially reduces the time needed to count B. cereus colonies, compared to hand counting.

Deep learning and machine learning are not suited to this study, because they both require large amounts of data. All educational institutions in the region were under strict Covid-19 protocols when the study was conducted, so 20 cultured Petri dishes were obtainable and no more. Analysis was conducted on 420 images from these dishes. Brightness and contrast were variously increased and decreased (from -15% to +15%) on a portion of the images to test the method rigorously under a variety of conditions. Bacteria colonies were counted using image processing techniques.

The purpose of this study is to reduce the amount of time required to count colonies, which can benefit microbiologists, nutritionists, and others. The paper is organized as follows. Section I provides an introduction. Section II reviews relevant previous studies. Section III describes materials and methods used. Section IV reports the experimental results. Section V is a discussion of the results. Section VI summarizes the study in a conclusion. Section VII acknowledges various people who contributed support and assistance in the course of this study.

II. Related Work

Previous studies on the counting of bacteria colonies have varied in both the type of bacteria counted and the methods employed. Hogekamp et al. (2020) developed an automated image-based system for counting Escherichia coli DH5-Alpha colonies. When each petri dish image was imported into Hogekamp's program, it was divided into 6 equal parts similar to slicing a pizza. The system randomly selected 1 of these 6 parts. After that, five image processing techniques were used to count the colonies: background removal, contrast increase, median filter (to reduce visual noise), brightness increase, and dark maxima detection of colonies. The resulting number of colonies was multiplied by 6 to make up for the unprocessed image parts. The main problem that caused counting errors was the random selection of image parts, because the parts did not all have similar numbers of colonies [10].

Ferrari, Lombardi, & Signoroni (2017) designed and tested two different machine learning approaches to counting colonies. The first method was based on extraction of a complete set of custom-designed morphometric and radiometric features used within a Support Vector Machine (SVM) solution. e second approach was based on the design and configuration of a Convolutional Neural Network (CNN), a kind of deep learning architecture. Both approaches used the same 28,500 images. The study found that the CNN method worked better and had a 91.5% accuracy rate. Ferrari et al. were able to use machine learning because of the large number of images available [11].

Siragusa et al. (2018) developed a semi-automated, image-based colony counting system named CoCoNut (Colony Counter developed by the Nutech department at the Technical University of Denmark). CoCoNut consists of an ImageJ macro program and a 3D-printed photo light box. V79 and HeLa colonies were tested. Results were compared to other available programs. CoCoNut succeeded in distinguishing between freestanding and overlapping colonies. CoCoNut software was also faster than other available programs. However CoCoNut required the user to adjustment program parameters according to colony size. Depending on users to do this correctly is by nature risky and best avoided when possible. CoCoNut's counting results were sometimes inconsistent because of the parameter input requirement [12].

Correa et al. (2020) developed semi-automated counting of the bacterium Bacillus pumilus and the yeast Meyerozyma guilliermondii, both isolated from diesel oil. Correa's method involved a series of steps. First, the original image is resized to a smaller image to save processing time, and this smaller image is then converted to grayscale. The grayscale image is then converted to a binary (black and white) image. A binary opening filter is applied to filter out most of the optical noise. Next, colonies are identified using the luminosity features, because a colony is more luminous at its core than at its edges. Using that fact, the identification method involves computing the points of local maximum luminosity. Correa's method was compared with regular hand counting, and the results showed accuracy of 94.64% and 93.41%, respectively. Correa's system delivers good accuracy, but it runs on computers, not on phones, so it is not particularly mobile [13].

III. Materials and Methods

A. *Materials*

Three kinds of software were used in this study. The tinkercad.com website was used to design models for 3D printing. Python 7.2 software was used to test the image processing techniques. Finally, Android Studio 4.2 software was used to write the BCICS phone application. Materials used consisted of a photo light box, a Petri dish box, an Android smartphone (OPPO Reno 2 with 48-megapixel camera), a phone stand, and Petri dishes with B. cereus colonies. The photo light box, the Petri dish box, and the phone stand were designed using tinkercad.com and the objects were then formed using a 3D printer (Flashforge Finder 2.0; Flashforge; Jinhua City, Zhejiang Province, China). Fig. 1 shows an actual representation the photo light box arrangement.

B. *Methods*

The new BCICS method to count B. cereus colonies in Petri dishes involves three stages: the input stage, the processing stage, and the output stage.

1) Input stage: B. cereus colonies from milk samples were cultured on Plate Count Agar (PCA) in Petri dishes in advance at a microbiology laboratory. All Petri dish images in this study were taken using the previously described photo light box and smartphone camera. The images had a focal length of 4.77 millimeters, an aperture of f/1.7, and an ISO in the range of 700-799.

Sample images showing the color, size, and shape of B. cereus colonies can be seen in Fig. 2. While some of the colonies are quite large, others are small enough to be difficult to see with the naked eye, so a magnifying glass is used for hand counting.

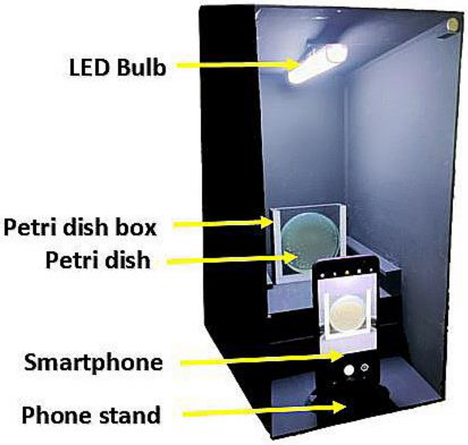


Fig. 1. Actual Representation of the Photo Light Box Arrangement.

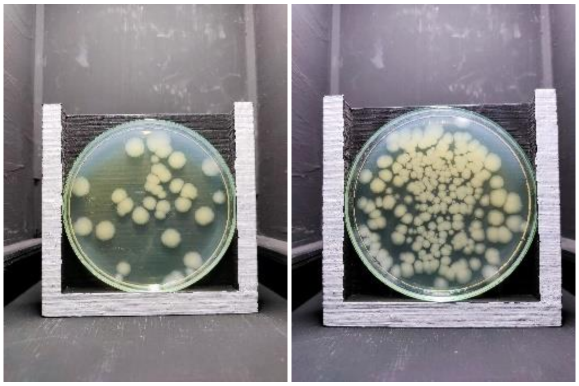

(a) (b)

Fig. 2. Sample Images of B. Cereus Colonies in Petri Dishes, showing how the Denseness of the Colonies has been Intentionally Varied, with (a) Sparsely Inoculated and (b) Densely Inoculated.

2) Processing stage: The prepared images are processed in seven steps: isolate the Petri dish box, isolate the Petri dish, remove heightened brightness, reduce turbidity, remove the background, mark the colonies, and count the colonies. These steps are shown in Fig. 3.

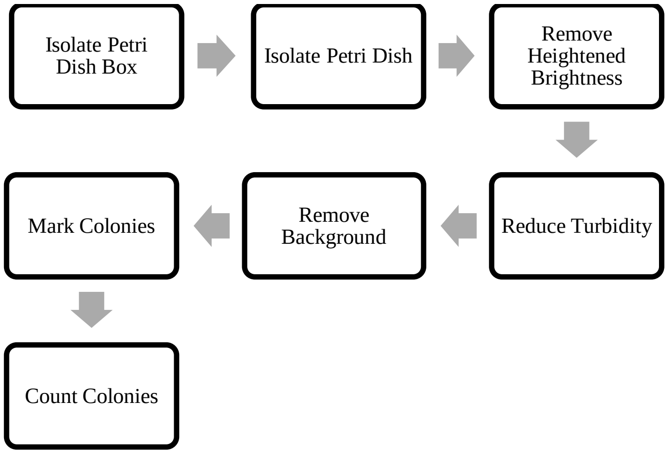


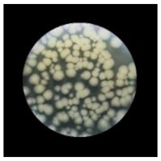
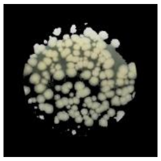
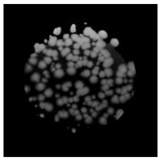
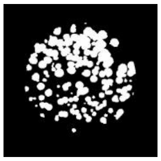
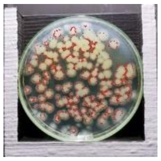
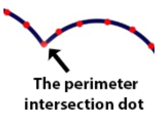
Fig. 3. The Seven Image Processing Steps Carried out by BCICS.

All seven steps involve image processing techniques. The first six steps prepare the image for colony counting, and the last step is where the actual counting occurs. Each of the seven steps in the processing stage is explained in more detail below in Table I.

a) Isolate Petri dish box: Whenever possible, it is good to remove irrelevant parts of an image that could potentially cause errors when looking for the B. cereus colonies. In this first step, the Petri dish box is detected and then everything outside the box is removed. Apart from reducing potential errors, this step also centers the Petri dish in the image. In order to isolate the petri dish box, the image will first be converted to grayscale. The grayscale image will then be converted to a binary image. Finally the Petri dish box can be isolated from the binary image.

Each of the original color images has three color components: red, green, and blue (RGB). Each pixel in the color image is defined by three integers in the range 0-255, which represent the intensity of the three colors red, green, and blue, respectively. The conversion to grayscale is carried using the luminosity method (also called the weighted method) rather than the average brightness method, because the luminosity method considers that the pixel value weight of each color component is different. This solves the problem of the image having three different color components (RGB) that have three different wavelengths. Therefore the luminosity method gives better results than the average method, which tends to produce a darker image. The luminosity method is also the most common option for conversion to grayscale [14-15]. After the image is converted to grayscale, each pixel is instead defined by a single integer in the same range, but with 0 representing black, 255 representing white, and each integer in between representing a corresponding shade of gray.

TABLE I. THE SERIES OF STEPS IN THE PROCESSING STAGE OF BCICS

Step	Action	How It Works	Image Result
1	Isolate Petri Dish Box	White pixels are summed horizontally and vertically to define the four corners of the Petri dish box and enable cropping.	
2	Isolate Petri Dish	The Petri dish is detected from its circular shape and then all the pixels outside the dish are turned to black.	
3	Remove Heightened Brightness	The V component of the HSV color model is used to isolate the band of heightened brightness and then the pixels in the band are turned to black.	
4	Reduce Turbidity	Contrast is increased to overcome the effect of turbidity.	
5	Remove Background	The difference between the colonies and the background is accentuated by converting all pixels to black or white.	
6	Mark Colonies	Red dots are plotted along the perimeter of the colonies.	
7	Count Colonies	Overlapping colonies are identified and all colonies are counted.	

After the grayscale image is ready, the conversion of the grayscale image to a binary image is accomplished using the thresholding technique. This means designating all pixel values as either 0 (black) or 1 (white). When a grayscale pixel value is below a chosen threshold value, that pixel is set to 0. When the grayscale pixel value is equal to or higher than the threshold value, that pixel is set to 1. An example of an image converted from color to grayscale and then to binary is shown in Fig. 4.

The Petri dish box is isolated in the binary image by using the projection profile technique. This means that the number of white pixels in the binary image is summed vertically and horizontally to define the corners of the Petri dish box. These coordinates in the form of pixel positions are then used to remove the irrelevant area outside the Petri dish box. Examples of the horizontal and vertical projection profiles of a Petri dish box image are shown in Fig. 5.

Specifically, the sum of white pixels on the horizontal axis is used to define the top and bottom edges of the Petri dish box, and the sum of white pixels on the vertical axis is used to define the left and right edge of the Petri dish box. On the horizontal axis, the top edge of the Petri dish box can be determined as the last row from the top at which the sum of white pixels is still zero, and the bottom edge of the Petri dish box is the last row from the bottom at which the sum of white pixels is still zero. Similarly, on the vertical axis, the left edge of the Petri dish box is the last column from the left where the sum of white pixels is still zero, and the right edge of the Petri dish box is the last column from the right where the sum of white pixels is still zero. Using these four reference points, the original color image is cropped. The binary image is discarded. A sample result is shown in Fig. 6.

b) Isolate Petri dish: If the Petri dish box is not removed from the image, errors will occur as a result of visual noise caused by the rough surface of the box. The circular shape of the Petri dish makes it suitable for use with the Circle Hough Transform (CHT) technique, which can detect circular objects in an image. The CHT technique accepts a radius range specified by the user and then scans an image, looking for circles with a radius that falls within the designated range [16]. In the current study, the radius range to seek was defined as 150-200 pixels, which corresponds to the radius of a Petri dish. Once the circle of the Petri dish was found by CHT, all pixels outside of the circle were set to 0 (black) in order to isolate the dish image from background, as shown in Fig. 7.

c) Remove heightened brightness: A heightened band of brightness just inside the perimeter of the dish, as seen in Fig. 8(a), can contribute to errors later when the background is removed, because the brightness makes it more difficult to distinguish the bacteria colonies from the background. This band can be removed using the HSV color model, which consists of three components: hue (H), saturation (S), and value (V) [17-18]. The current study, uses only the V component, shown in Fig. 8(b), which represents the degree of brightness of a pixel, ranging from 0 to 255, where 0 (black) is low brightness and 255 (white) is high brightness. The V component is used for thresholding here to create a binary image. Any V component pixel value that is greater than or equal to 60 is set to 1 (white), and values less than 60 are set to 0 (black). An example is shown in Fig. 8(c). The white pixels resulting from thresholding are then set to black in the original image to remove the band of heightened brightness. The Petri dish image after removal of the heightened brightness is shown in Fig. 8(d).

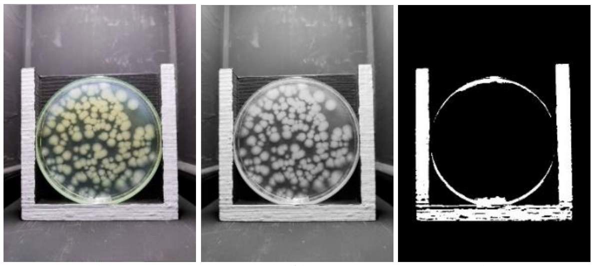

(a) (b) (c)

Fig. 4. A Sample Petri Dish Image Converted from (a) Color to (b) Grayscale and then to (c) Binary.

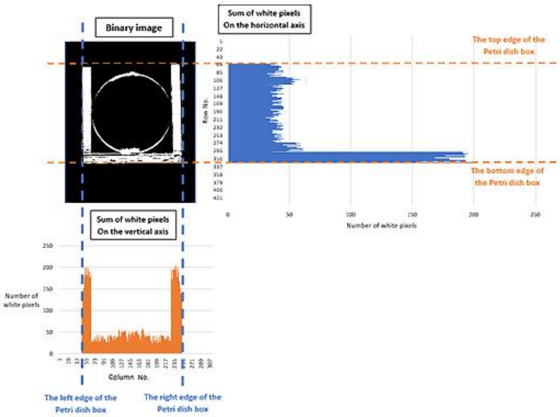


Fig. 5. Example of the Horizontal and Vertical Projection Profile of a Petri Dish Image.

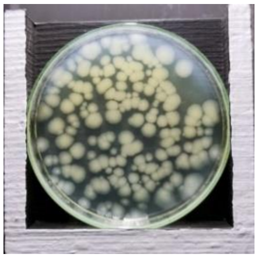


Fig. 6. After using the Projection Profile Technique, the Cropped Image Contains only the Petri Dish and the Petri Dish Box.

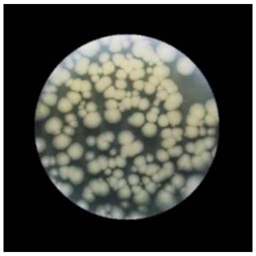
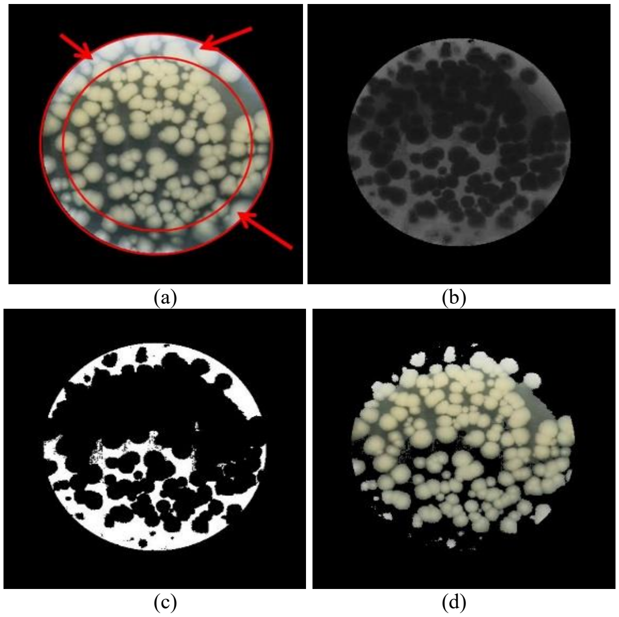


Fig. 7. After use of the CHT Technique, the Petri Dish is alone in the Image, with only Black Pixels outside the Dish Edge.

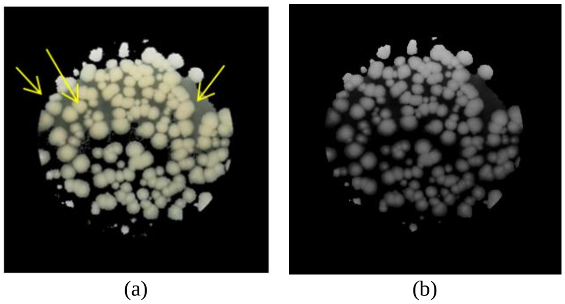


(a) (b) (c) (d)

Fig. 8. Removing the Band of Heightened Brightness from the Petri Dish Image: (a) Area of Heightened Brightness, (b) V Component of Previous Image, (c) Binary Image from V Component, and (d) Image after Removal of Heightened Brightness.

d) Reduce turbidity: The culture medium has a natural turbidity from the variety of chemicals it contains, as seen in Fig. 9(a). Turbidity, too, makes removal of the background more difficult later because it decreases contrast in the image. Power-Law transformation is therefore used to overcome the effect of the turbidity by increasing contrast in the image, as shown in Fig. 9(b).

e) Remove background: Removing the background involves setting the background pixels to 0 (black) and the colony pixels to 1 (white). However, while regular thresholding was a useful technique earlier to prepare for the projection profile, the level of visual noise it generates is too high for the current situation, which is more sensitive. Therefore, Adaptive Thresholding (AT) is used to remove the background. Unlike regular thresholding, AT has no set threshold value. The threshold value is dynamic and changes in order to adapt to different brightness environments in different parts of the image [19]. AT works in the following way. Every pixel in the image is processed one by one. At each pixel, a square block of surrounding pixels at which the target pixel is at the center are examined for their brightness. The average brightness of the block is calculated, and the threshold, which is inputted by the user only as a constant, is adjusted automatically to the average brightness of that block before determining if the target pixel will be set to black or white. When the target pixel is at or near the image edge, the missing part of its block that would theoretically extend outside the image is automatically ignored, and the average brightness is calculated from the remaining portion of the block that can still be constructed within the image. In the current study, the block size was set to 555 x 555 pixels, and the AT constant value was set to -25. An example of a Petri dish image after completion of this step is shown in Fig. 10.



(a) (b)

Fig. 9. Petri Dish Image (a) with Natural Turbidity in the Culture Medium and (b) after Turbidity was decreased using the Power-Law Transformation.

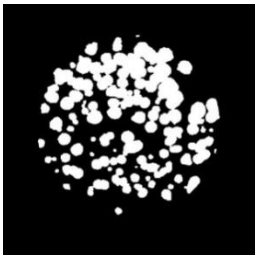


Fig. 10. A Petri Dish Image after use of Adaptive Thresholding to make the Colonies Stand out more clearly from the Background.

f) Mark Colonies: Colonies currently appear in a binary image as clusters (i.e. blobs) of white pixels against a black background. Since the colonies are of many sizes and often overlap, the Connected-Component Labeling (CCL) technique is now used. First CCL finds all the blobs of white pixels [20]. However this is not enough to distinguish between overlapping colonies. So CCL then marks the perimeter of each blob with red dots. Fig. 11 shows these red perimeter dots superimposed on the original petri dish image, with the red dots surrounding each colony or group of overlapping colonies.

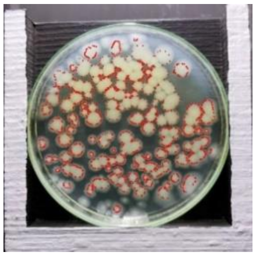


Fig. 11. Standalone Colonies and Overlapping Colonies Marked with Red Dots on their Perimeter after using the Connected-Component Labeling Technique.

g) Count Colonies: All of the previous steps have prepared the image for the actual counting of the *Bacillus cereus* colonies, which happens now. When there are overlapping colonies, CCL will always place a red dot at each intersection of the perimeters of the overlapping colonies. Most of the red perimeter dots are not at perimeter intersections, but wherever there is an intersection, CCL will put a dot there. CCL can use these perimeter intersection dots (PIDs) to distinguish and count overlapping colonies with the Convex Hull Detection function. Fig. 12 schematically represents two overlapping colonies, showing sample placement of red dots along the perimeter and the two PIDs for this configuration.

A total of 1-2 PIDs represents 2 cell colonies, 3 PIDs represent 3 cell colonies, 4 PIDs represent 4 cell colonies, and so on. The number of colonies in the whole Petri dish is summed for the final result.

3) Output stage: The previously described Input Stage and Processing Stage have been incorporated into a convenient and user-friendly tool, the BCICS android application. This is where the results of the first two stages are processed and displayed to the user. The BCICS app was built with Android Studio (Version 4.2.2) and OpenCV library (Version 3.4.14).

When the user launches BCICS, they are prompted to take a petri dish photo. If for any reason the user wishes to retake the photo, they can do so as many times as desired. As soon as the user confirms the photo, the application processes the image and displays the total number of colonies counted. At this last step, the user can choose either to photograph another Petri dish or to exit. A flowchart of the application is shown in Fig. 13.

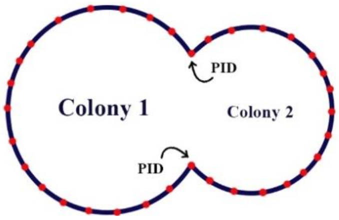


Fig. 12. A Schematic Representation of Two Overlapping Colonies, showing Red Dots Placed along the Perimeter and the Two Perimeter Intersection Dots (PIDs) for this Configuration.

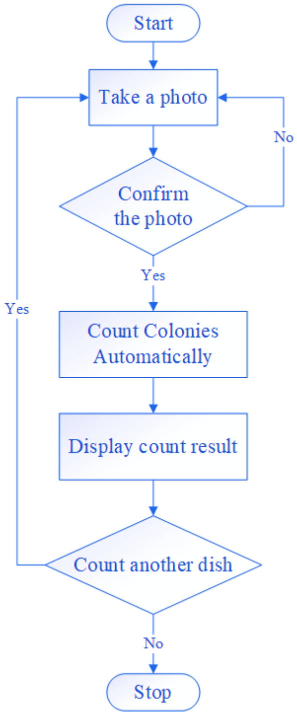


Fig. 13. Flowchart of the BCICS Smartphone Application.

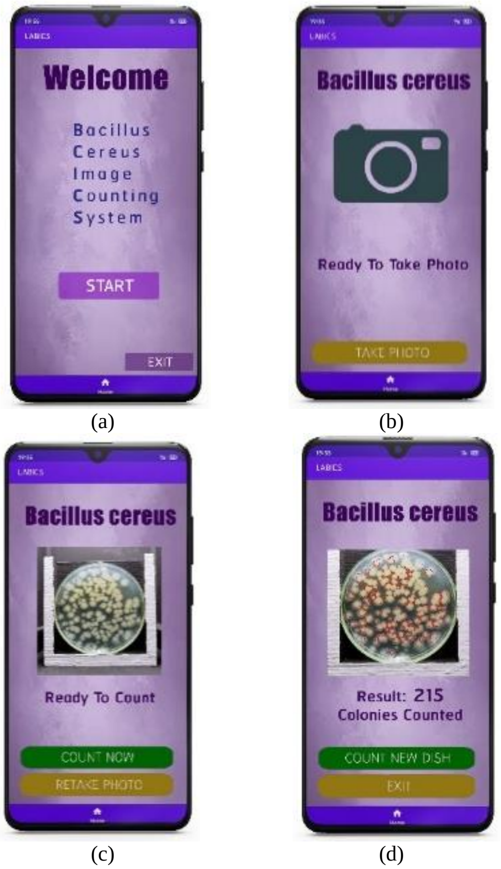


Fig. 14. Steps in using the BCICS Application: (a) Home Page, (b) Photography Page, (c) Photo Confirmation Page and (d) Result Display Page.

A sample usage of the BCICS app is shown in Fig. 14. The application opens with the home page in Fig. 14(a). Pressing Start takes the user to the dish photography page in Fig. 14(b). Taking a dish photo brings the user to the photo confirmation page in Fig. 14(c). Acceptance of the photo initiates the automatic counting of the colonies, and the user is taken to the result display page in Fig. 14(d), where both the total number of colonies in the dish and the colony locations are displayed.

IV. Results

The BCICS application counted the bacteria colonies in each Petri dish in an average of 4.44 seconds. The minimum time was 3.50 seconds, and the maximum was 5.41 seconds. Hand counting of the same dishes for comparison took an average of 5:32 minutes. The minimum time was 53.04 seconds, and the maximum was 6:51 minutes. Previous studies report that hand counting takes most people 2-15 minutes per dish. The percentage error was calculated according to the formula shown in Equation 1:

$$\frac{hand\ count - BCICS\ count}{hand\ count} \times 100 \tag{1}$$

Fig. 15 shows the distribution of various error rate ranges across all dishes. The error rate range with the most incidences is 0-3.6. The average error rate across all dishes is 9.86% and the standard deviation of the error rate is 7.64. In comparison, the error rate for hand counting falls in the range of 5-10%, because accurate hand counting depends on many factors [21].

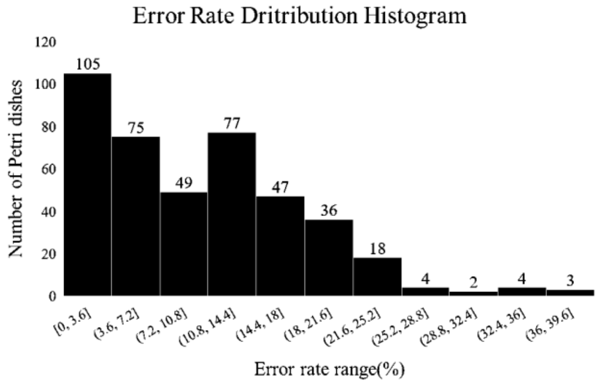


Fig. 15. Histogram Showing a Spectrum of Error Rate Ranges and the Number of Petri Dishes inside each Range.

V. Discussion

Although BCICS delivers high accuracy results, at present some errors are still unavoidable. Within the 8.53% error rate, overcounting represents 66.15% of the errors and undercounting represents 33.85%. Interestingly all errors can be traced back to only two steps: Step 3 (Remove Heightened Brightness) and Step 6 (Count Colonies). All the overcounts occurred in Step 6, while most of the undercounts occurred in Step 6, with some in Step 3 as well. The most common cause of overcounting is the presence of a gap in a colony group. An example of such a gap is shown in Fig. 16. These gaps create inadvertent extra PIDs that disrupt the formula for counting colonies based on the number of PIDs. This issue is a common limitation of CCL. Koomsap et al. (2014) also encountered this issue, and they developed a topological hierarchy-contour tracing algorithm to handle nests of interconnected contours [22]. It might be possible to adapt Koomsap's solution to BCICS as well.

The main cause of undercount errors is another kind of unusual formation of overlapping colonies that interferes with the system of counting PIDs to count colonies. Undercounting can happen when curves of two or more overlapping colonies are so similar and closely aligned to each other that they appear to be a single curve with no perimeter intersection and therefore no PID. This kind of missing PID leads to an undercount error. An example of such a configuration is shown in Fig. 17. Fig. 17(a) shows a common configuration of two overlapping colonies. The perimeters of the two colonies are distinguishable enough that CCL can sense the need for placement of two PIDs, and the colonies are counted correctly. However, Fig. 17(b) shows a less common configuration in which two colonies are even closer, and their perimeters have almost merged. In this unusual case, CCL may not sense the need for PIDs, so this configuration can generate an undercount error. The overlapping of colonies or other objects is a common issue that other researchers have tried to solve in various ways, including the erosion algorithm (a morphological image processing technique) [23], the watershed transform algorithm [24], and the distance transform algorithm [25]. Unfortunately none of these alternative techniques were found to give better results that CCL.

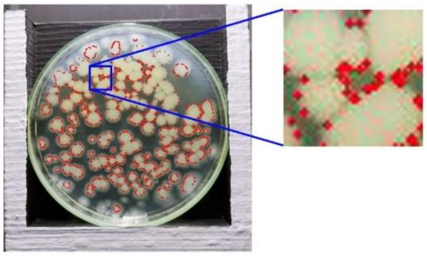


Fig. 16. An Example of a Gap inside a Colony Group, which Causes Creation of Inadvertent Extra PIDs that Disrupt the Formula for Counting Colonies.

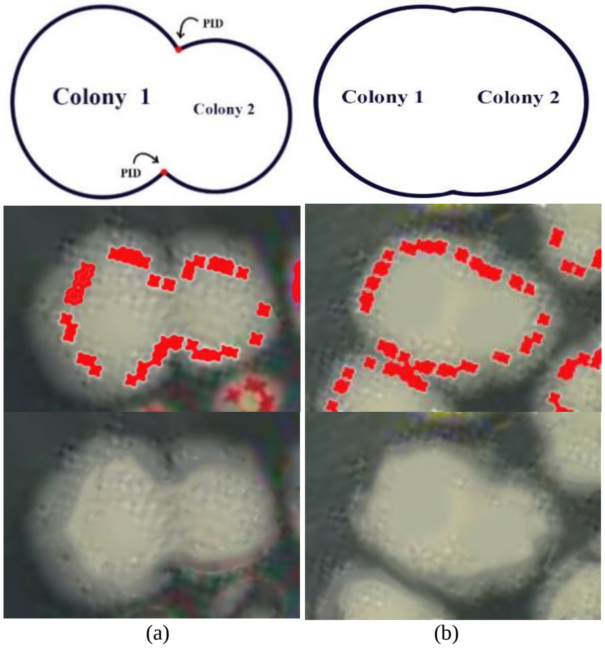


Fig. 17. Examples of Overlapping Colonies in which (a) the Perimeters are Sufficiently Distinguishable to Generate PIDs and be Counted Correctly and (b) the Perimeters are too Similar to Generate PIDs and an Undercount can occur.

A less common cause of undercount errors is unintended removal of a colony from the image during image processing step 3, when the HSV color model and thresholding are used to remove the band of heightened brightness. An example is shown in Fig. 18. Recently, Yifei et al. (2021) encountered an issue similar to this one when processing photos of car license plates. Reflection from another car's headlights created glare and thus areas of excessive brightness in the photos. Yifei applied tone mapping, and this technique was effective in enhancing the local contrast and preserving the image details and structures [26]. It is possible that tone mapping might also be useful in the current study.

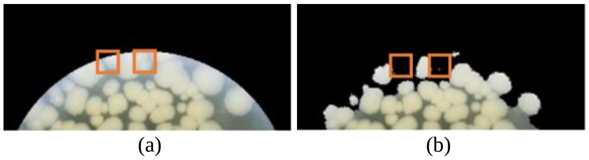


Fig. 18. Examples of Colonies Inadvertently Lost from an Image during Processing Step 3 are shown (a) before Removal of the Heightened Brightness Band and (b) after Removal of the Band.

VI. Conclusion

The BCICS application developed in this study successfully reduces the time needed to count *Bacillus cereus* bacteria colonies in a Petri dish from the average of 5:32 minutes required by hand counting to an average of 6.04 seconds in this automated method. By integrating and automating the seven steps involved, BCICS makes the counting process intuitive and easy to carry out. Because the application utilizes image processing rather than less nimble methods such as densitometric software, it can handle any number of Petri dishes images, even if there are only a few. Although BCICS makes counting B. cereus much faster than hand counting, its accuracy rate of 90.14% is high and comparable to hand counting. In future research, use of the morphological operation called erosion might enable highlighting of inconspicuous perimeter intersections of overlapping colonies and thereby raise the performance of BCICS yet further.

Acknowledgment

The authors are grateful to Ms. Nootsara Yinyom, master's degree student in the Faculty of Medical Science, Naresuan University, for her invaluable assistance in the microbiology laboratory. Thank you also to Mr. Paul Freund of Naresuan University Writing Clinic, Division of International Affairs and Language Development (DIALD) for his help proofreading and editing this manuscript.

References

[1] Sever, B., U.S. Food and Drug Administration (FDA). 2020, Salem Press.

[2] United States, F. and M. Drug Administration. Division of, Bacteriological analytical manual. 1969, Washington.

[3] Can, H.Y., et al., Psychrotrophic properties, toxigenic characteristics, and PFGE profiles of Bacillus cereus isolated from different foods and spices. Propriedades psicrotróficas, características toxigênicas e perfis PFGE de Bacillus cereus isolado de diferentes alimentos e especiarias., 2022. 52(4): p. 1-11.

[4] Antequera-Gómez, M.L., et al., Sporulation is dispensable for the vegetable-associated life cycle of the human pathogen Bacillus cereus. Microbial Biotechnology, 2021. 14(4): p. 1550.

[5] Jovanovic, J., et al., Bacillus cereus food intoxication and toxicoinfection. Comprehensive Reviews in Food Science & Food Safety, 2021. 20(4): p. 3719-3761.

[6] Jessberger, N., et al., *The Bacillus cereus Food Infection as Multifactorial Process.* Toxins, 2020. 12(11).

[7] Valik, L., F. Gorner, and D. Laukova, Growth dynamics of Bacillus cereus and shelf-life of pasteurised milk. 2003, MINISTRY OF AGRICULTURE CZECH REPUBLIC: Czechoslovakia. p. 195-202.

[8] Pisuttilap, N., & Saengswetmaneengam, N., Study of Escherichia coli analysis in frozen seafood by MPN technique / Suksa kan wikhro Escherichia coli nai ahan tha-le chae khaeng duai withi MPN technique. Bulletin of the Department of Medical Sciences (Thailand), 2001. 43(2): p. 95-101.

[9] Puchalt, J.C., et al., Active backlight for automating visual monitoring: An analysis of a lighting control technique for Caenorhabditis elegans cultured on standard Petri plates. PLoS ONE, 2019. 14(4): p. 1-18.

[10] Hogekamp, L., S.H. Hogekamp, and M.R. Stahl, Experimental setup and image processing method for automatic enumeration of bacterial colonies on agar plates. PLoS ONE, 2020. 15(6): p. e0232869.

[11] Ferrari, A., S. Lombardi, and A. Signoroni, Bacterial colony counting with Convolutional Neural Networks in Digital Microbiology Imaging. Pattern Recognition, 2017. 61: p. 629-640.

[12] Siragusa, M., et al., Cell colony counter called CoCoNut. PLoS ONE, 2018. 13(11): p. 1-18.

[13] Correa, C., et al., Use of digital images to count colonies of biodiesel deteriogenic microorganisms. Journal of Microbiological Methods, 2020. 178.

[14] Kanan, C. and G.W. Cottrell, Color-to-Grayscale: Does the Method Matter in Image Recognition? PLoS ONE, 2012. 7(1): p. 1-7.

[15] Wu, T. and A. Toet, Color-to-grayscale conversion through weighted multiresolution channel fusion. Journal of Electronic Imaging, 2014. 23(4).

[16] Gonzalez, R.C. and R.E. Woods, Digital image processing. 4th ed., Global edition ed. 2008: Pearson.

[17] Shamshad, A., Building Computer Vision Applications Using Artificial Neural Networks : With Step-by-Step Examples in OpenCV and TensorFlow with Python. 2020, Berkeley, CA: Apress.

[18] Ding, C., et al., Traffic Image Dehazing Based on HSV Color Space. 2021, IEEE. p. 5442-5447.

[19] Prasad Reddy, P.V.G.D., *Blood vessel extraction in fundus images using hessian eigenvalues and adaptive thresholding.* Evolutionary Intelligence, 2021. 14(2): p. 577-582.

[20] Damayanti, F., Y.K. Suprapto, and E.M. Yuniarno, *Segmentation of Javanese Character in Ancient Manuscript using Connected Component Labeling*. 2020, IEEE. p. 412-417.

[21] Jarvis, B., Statistical aspects of the microbiological examination of foods. 3rd ed. ed. 2016: Elsevier.

[22] Koomsap, P. and N. Chansri, Topological hierarchy-contour tracing algorithm for nests of interconnected contours. International Journal of Advanced Manufacturing Technology, 2014. 70(5-8): p. 1247-1266.

[23] Glagoevs, J. and K. Freivalds, Statistical Method for Object Counting. 2017. p. 61-64.

[24] Arámbula Cosío, F., et al., Automatic analysis of immunocytochemically stained tissue samples. Medical and Biological Engineering and Computing, 2005. 43(5): p. 672-677.

[25] Zhang, J., et al., A comprehensive review of image analysis methods for microorganism counting: from classical image processing to deep learning approaches. Artificial Intelligence Review: An International Science and Engineering Journal, 2021: p. 1-70.

[26] Yifei, W., et al., High-Brightness Image Enhancement Algorithm. Applied Sciences, 2021. 11(11497): p. 11497-11497.