Mobile application for automatic bacterial density estimation

Phongsatorn Taithong[1], Siriwan Wichai[2], Rattapoom Waranusast[1]
and Panomkhawn Riyamongkol[1]*

[1]Faculty of Engineering, Naresuan University, Phitsanulok Province, 65000, Thailand
[2]Faculty of Medical Science, Naresuan University, Phitsanulok Province, 65000, Thailand
* Corresponding author. E-mail address: panomkhawnr@nu.ac.th
Received: 5 August 2022; Revised: 27 December 2022; Accepted: 11 January 2023

Abstract

The traditional method for calculating the concentration of viable bacteria in a pure source is to use serial dilutions. This conventional method takes more than 72 hr and involves a series of complex steps that must be done by a microbiologist, including culturing the colonies. In contrast, this study utilizes a combination of image processing and machine learning developed into a mobile application that can estimate the concentration of viable bacteria by simply taking a picture, substantially reducing the time required. To create this new estimation model, a series of image processing techniques optimize and standardize a dataset of photographed test tubes containing pure bacterial suspension, culminating in the delimiting of the Turbidity Testing Zone (TTZ), which is uniform across all the test tube photos. Bacterial concentration is correlated with suspension turbidity, so statistical data from the pixels within the TTZ is analyzed using four machine learning algorithms to find the optimal estimating model. The finished model becomes the foundation of the Viable Bacteria Image Estimating System (VBIES) android application, which enables any user to easily and conveniently determine the concentration of viable bacteria in a test tube with an accuracy of 97.57%. In contrast to the several days required by the traditional methods, the VBIES application estimates the concentration of viable bacteria in only 3-5 seconds.

Keywords: estimating bacterial concentration, image processing, machine learning, mobile application, decision tree learning

Introduction

Bacteria have many commercial uses. The food industry uses bacteria to produce fermented foods. The garment industry uses bacteria to tan animal skins. Agriculture uses bacteria to produce fertilizers. In irrigation canals, bacteria are used to test water quality. In the pharmaceutical industry, bacteria are used to produce antibiotics (Cangliang & Yifan, 2022; Dong et al., 2022; Jayapal et al., 2022; Novik, Savich, & Meerovskaya, 2018; Renganathan & Olubukola Oluranti, 2021). The ability to determine the concentration of bacteria cells is vitally important. The traditional process to test the concentration of bacteria is Standard Plate Count (SPC) (Taylor, Allen, & Geldreich, 1983). SPC is also called by numerous other names, including Aerobic Plate Count (APC) (Yunxia, Lijuan, Cuizhi, Zhiyong, & Lijun, 2021), Total Viable Count (TVC) (Moore-Colyer & Fillery, 2012), and Total Viable Microorganisms (TVM) (Osano, Bessho, Asai, & Takei, 1979). However, all of these refer to the same workflow. SPC is reported as colony-forming units per milliliter (CFU/ml) (Prockop, Bunnell, Phinney, Pochampally, & Walker, 2008).

The traditional method for establishing the density of inoculum in a test laboratory is based on turbidity. To begin, pure bacteria are mixed with a diluent to form a bacterial suspension, and a spectrophotometer is used to measure its turbidity (Tao, Luze, & Long-Fei, 2014). The bacterial suspension undergoes approximately 7-9 serial dilutions to obtain a useful range of bacterial concentrations, and these various concentrations are then plated on culture media. The process to this point takes at least two hours, including media preparation, inoculum

preparation, serial dilution, and plating. After 24-48 hr of incubation, the colonies are visible and can be counted (Maturin et al., 2016). Great caution must be taken by the technician at this stage to avoid contamination of the dishes by other bacteria. Finally, the bacteria colonies in the cultured Petri dishes are hand counted to find a dish that falls in the range of 30-300 colonies (Tomasiewicz, Hotchkiss, Reinbold, Read, & Hartman, 1980), which is the required range for calculating the bacterial density in CFU/ml. This hand counting of the colonies represents an additional 1-10 minutes per Petri Dish (depending on the number of colonies) in the conventional method for determining the bacterial density and unfortunately, manual counting is inherently prone to errors. Thus, each step of the conventional method for determining bacterial density requires a lot of time to conduct the necessary procedures as well as expensive laboratory equipment, disposable material from the client and the skill of a trained microbiologist.

There have been previous studies that used image processing techniques to reduce the time consumed to count cells in a Petri Dish. Velier et al. designed The STEMvision™ system to count and mark hematopoietic stem cell (HSC) colonies that have been cultured for several days in a semisolid media. This system is the first semi-automated instrument using image processing for this purpose. Results show that the STEMvision™ system is faster than manual counting. However, STEMvision™ only counts hematopoietic progenitors and cannot be used for bacterial colonies. Another drawback was that the process was not fully automatic. The user had to manually select the area to be analyzed (Velier et al., 2019).

Thungopakun et al. (2014) developed software to count and measure the size of colonies of *Staphylococci* bacteria, which can cause food poisoning and are therefore of concern in the food and pharmaceutical industries. This software, called Micros-Staph, was developed using image processing techniques to extract the features of the colonies, and these features were then used with a Support Vector Machine (SVM) model to differentiate the colonies from visual noise. The number of colonies could be counted easily in this way. Micros-Staph measured colony size using a colony bounding box, which is a transparent overlay image showing a known size scale. The counting results were very close to an outside expert's hand counts, with a correlation of 0.99 for images captured at 400x magnification and 0.98 for 1000x magnification. The results for colony size were 0.5 - 0.9 microns in diameter, which compared well with the actual known colony size of Staphylococci bacteria colonies (Thungopakun et al., 2014).

Zhu, Yan, Xing, and Tian (2018) developed an image processing method to count colonies on a Petri Dish consisting of four steps: eliminating optical noise outside the Petri Dish; removing the dish rim and wall; distinguishing between clustered colonies; and counting colonies using the connected region labeling, the Distance Transform, and Watershed Transform algorithms. The system had an accuracy of 82%. One disadvantage was that the process was not automatic. Users had to manually select the area where colonies should be counted. However, the above studies could only replace the step of counting colonies, saving only 2-3 hours of work. They did not eliminate the need for lengthy dilution and culturing steps (Zhu et al., 2018).

Beal et al. (2020) estimated the density of *Escherichia coli* cells in liquid culture using a monochrome spectrophotometer to measure optical density. The researcher compared optical density and bacterial concentration with low-cost, highly accessible OD calibration protocols across 244 laboratories, examining eight strains of *E. coli*. The results showed that cell density could be estimated from optical density with a very high precision rate of 95.5%. Beal's research thus suggested that bacterial density was directly proportional to solution turbidity.

That is significant because image processing techniques can be used to measure turbidity, as in the following studies (Beal et al., 2020).

Godoy et al., measured turbidity by designing a device comprised of a webcam and either an LED light or a laser of adjustable wavelength. Images of test tubes containing liquid of various turbidity were captured while the LED or laser was shining brightly. Each image was separated digitally into its color component layers, and the red, green, blue, and grayscale pixel values were then analyzed for correlation with turbidity. Linear regression analyses revealed distinct relationships among the various pixels and the turbidity value. It was found that the green layer pixels provided the most accurate measure of turbidity, and these results were obtained using a white LED (Godoy et al., 2018).

Kumar, Afzal, and Ahmad (2022). developed a system that used Machine Learning to predict the turbidity of seawater in Hong Kong from the chemical and physical properties of the water. Artificial Neural Network (ANN), Support Vector Regression (SVR), and Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) algorithms were all compared, and the results showed that the LSTM-RNN model outperformed both the ANN and SVR models, yielding results with an accuracy of 88.45% (Kumar et al., 2022).

The study of Godoy et al. (2018) predicted water turbidity using image processing (Godoy et al., 2018) and Kumar et al. (2022) applied machine learning for the same purpose (Kumar et al., 2022), but they do not deal with bacterial concentrations. The current study extends the concept of turbidity to bacterial concentration, which is then estimated using image processing and machine learning, which has not been done before. This study introduces a new method of estimating the concentration of viable bacteria in a pure source using an image. The image goes through a series of image processing techniques, including Color Transformation, Thresholding, and Projection Profiling to identify pixels in the turbid region. During the development of the model, the image pixel data was tested with four different machine learning algorithms to optimize bacterial concentration estimation performance: Regression Analysis (RA), Artificial Neural Network (ANN), Decision Tree Learning (DTL), and Support Vector Machine (SVM). The final combination of procedures has been combined into a convenient Android smartphone application called Viable Bacteria Image Estimating System (VBIES, "V-bees") that substantially reduces the time needed to estimate the viable bacteria in a pure source, compared to the conventional method.

Methods and Materials

The Viable Bacteria Image Estimating System (VBIES) introduced here was developed in two stages, which are shown in Figure 1. The first stage was the development of the VBIES Model when various theories and procedures were tested and optimized. The second stage was the development of the VBIES Application when the finished model was adopted.

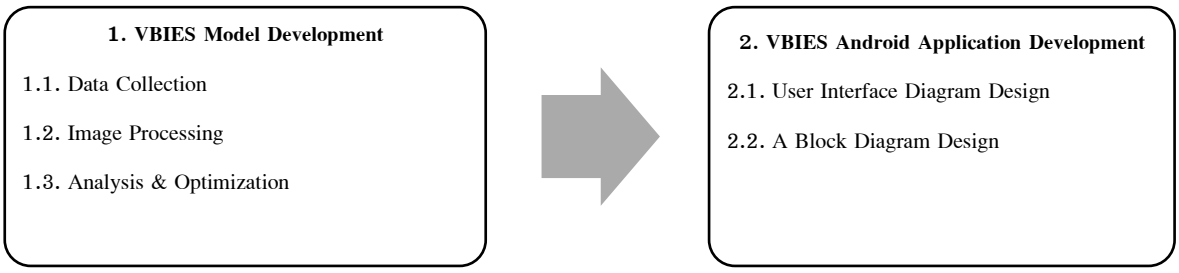


Figure 1 The two stages of VBIES development

1. VBIES Model Development

VBIES Model Development had three main components: data collection, image processing, and analysis & optimization, as shown in Figure 2. Each of these components is explained below.

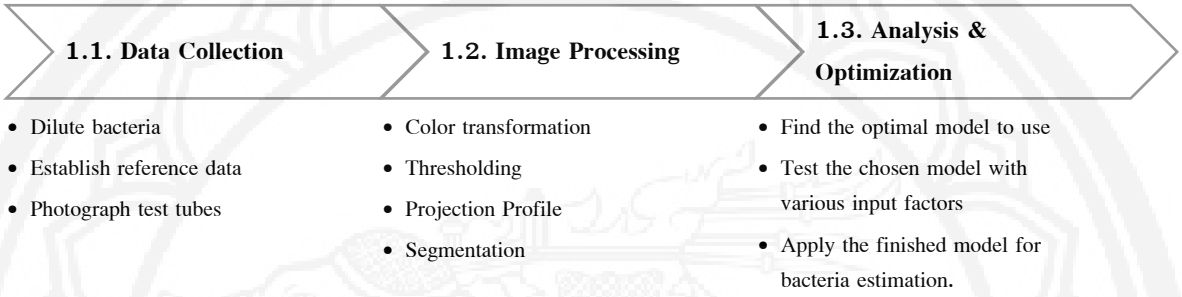


Figure 2 An overview of the three stages of the ABICS method

1.1 Data Collection

The dataset for the VBIES model came from four kinds of bacteria that are commonly tested for in raw milk quality tests: *Escherichia coli*, *Enterobacter aerogenes*, *Lactobacillus* sp., and *Bacillus cereus* (Agricultural Product Quality Standards, 2017). These Bacterial suspensions underwent 10-fold serial dilution using 0.85% sodium chloride (NaCl). *E. coli*, *E. aerogenes*, and *B. cereus* were inoculated onto Plate Count Agar (PCA) using the spread plate technique and then incubated at 35-37°C for 24-48 hr before being hand counted (Maturin & Peeler, 2001). *Lactobacillus* sp. was inoculated using the pour plate technique onto the same medium (PCA) and then incubated at 30°C for 24-48 hr. The Colony Forming Units per milliliter (CFU/ml) in each bacterial solution were determined, along with the turbidity of the solution. Higher concentrations of bacteria have higher turbidity. A summary of dilution specifications for the four types of bacteria is shown in Table 1.

Color images of these test tubes containing the diluted bacteria suspension were then collected. This was done using a photo light box, an Android smartphone (OPPO Reno 2 with 48-mega pixel camera), a phone stand, a test tube holder, and a test tube containing a suspension of bacteria. The photo light box, the phone stand, and the test tube holder were designed using www.tinkercad.com, and the objects were then formed using a 3D printer (Flashforge Finder 2.0; Flashforge; Jinhua City, Zhejiang Province, China). The photo light box is made of acrylic (40 cm high, 20 cm wide, and 30 cm long). The test tube holder is made of PLA plastic (105 mm high, 37 mm wide, and 38 mm long). 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. Figure 3 is a photograph of the photo light box arrangement.

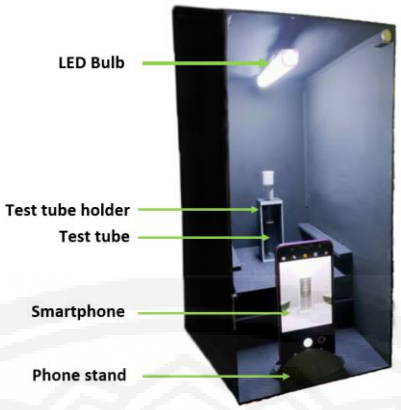


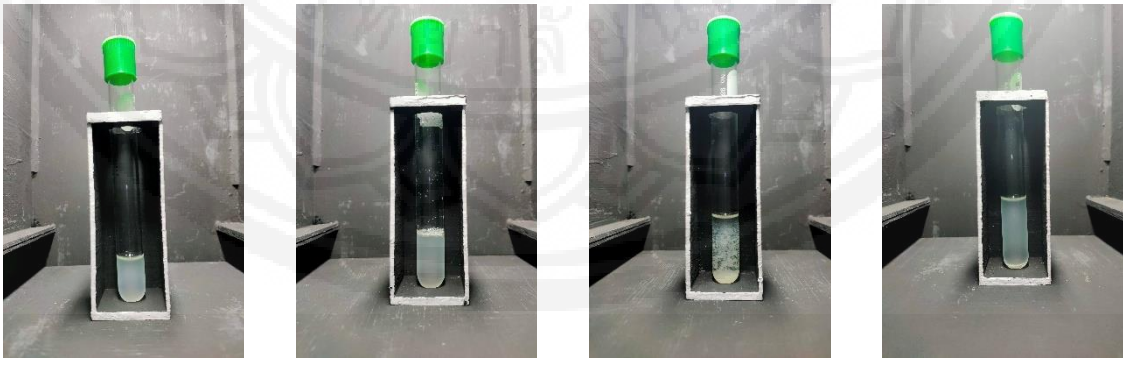
Figure 3 A photograph of the photo light box arrangement

A dataset of 2260 test tube images was thus produced, consisting of 600 *E. coli* images, 480 *E. aerogenes* images, 700 *Lactobacillus* sp. images, and 480 *B. cereus* images. All images were taken between Nov 2019 and Jun 2021. From the total 2260 images, 1800 (~80%) were randomly selected to be used to train the model. The remaining 460 (~20%) images were then used to test the developed model for accuracy.

Table 1 Dilution specifications for the four types of bacteria in 10-fold serial dilution and the number of tubes in each sample

Bacteria	Final concentration / Number of dilutions / Number of tubes			
	1:10000 (4 times)	1:100000 (5 times)	1:1000000 (6 times)	1:10000000 (7 times)
E. coli	100	150	150	200
E. aerogenes	60	120	180	120
Lactobacillus sp.	100	150	250	200
B. cereus	90	120	180	90

Examples of test tube images with the four types of bacteria are in Figure 4. The images show the faint color and sediment of bacterial suspensions. It is impossible to distinguish individual bacteria by the naked eye because of their microscopic size.



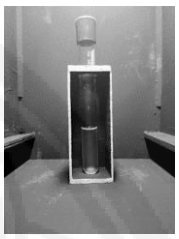
(a) (b) (c) (d)

Figure 4 Sample images of the four pure bacterial suspensions in test tubes, all at the same turbidity: (a) *E. coli*, (b) *E. aerogenes*, (c) *Lactobacillus* sp., and (d) *B. cereus*

1.2 Image Processing

In the VBIES model, four image processing techniques are used to prepare the test tube bacteria images for accurate measurement of the bacteria solution turbidity: color transformation, thresholding, projection profiling, and segmentation. The purposes of these techniques, along with samples of the images that result from them, are shown in Table 2, and each technique is explained in detail thereafter.

Table 2 The four techniques used in the image processing stage, along with their purpose and a sample image result

Step	Technique	Purpose	Image Result
1	Color Transformation	Convert color image to grayscale image	
2	Thresholding	Convert grayscale image to binary image	
3	Projection Profile	Sum white pixels along each axis and crop the image to remove the superfluous area outside	
4	Segmentation	Delimit the pixel area to analyze	

1.2.1 Color Transformation

The image processing in VBIES begins with color transformation because the later projection profile technique requires a binary image (Sarrafzadeh, Aghajari, & Shanbehzadeh, 2010), and conversion to a binary image first requires conversion to grayscale (Atmaja, Murti, Halomoan, & Suratman, 2016). The conversion to grayscale is carried out based on the luminosity method rather than on the average brightness method because the luminosity method (weighted method) takes into account that the pixel value weight of each

color layer is different. This solves the problem of having three different color layers that have three different wavelengths, which is not addressed using the average brightness method (Kanan & Cottrell, 2012). Each pixel in the incoming color image is defined by three integers in the range 0 – 255, which represent the intensity of the three colors red, green, and blue. After color transformation, each pixel in the grayscale image is 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 (Raveendran, Edavoor, Kumar, & Vasantha, 2018). The weighted method formula is Equation 1 (Bovik, 2009).

$$Grayscale = 0.299R + 0.587G + 0.114B \quad \text{Eq. 1}$$

Grayscale is a pixel value in the grayscale color model.

R is a pixel value in the red layer.

G is a pixel value in the green layer.

B is a pixel value in the blue layer.

1.2.2 Thresholding

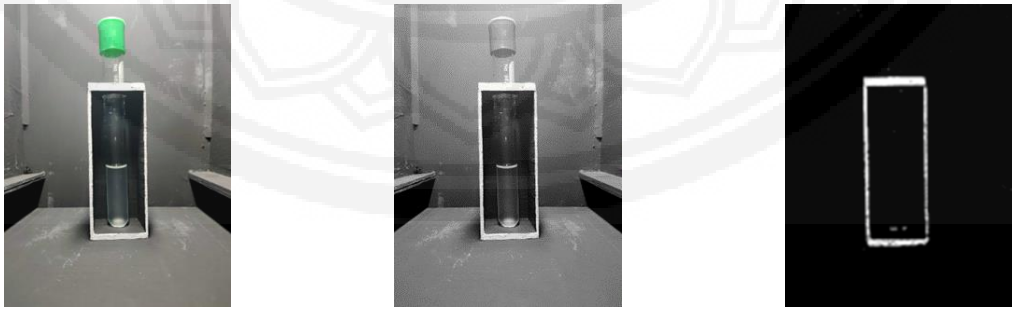
The grayscale image is now ready for conversion to a binary image, which will be used in the subsequent projection profile technique. The conversion to binary is carried out using the thresholding technique. Thresholding works by designating all pixel values as either 0 (black) or 1 (white), depending on a chosen threshold value. If a grayscale pixel value is below the threshold, that pixel is set to 0. If the grayscale pixel value is equal to or above the threshold value, that pixel is set to 1 (Pare, Kumar, Singh, & Bajaj, 2020). Equation 2 represents the thresholding formula (Bovik, 2009) which was determined to be 220 in this study. Figure 5 shows an example of an image converted from color image to a grayscale image and thence to a binary image.

$$Binary = \begin{cases} if\ Grayscale < Threshold\ value;\ Binary = 0 \\ else\ Grayscale \geq Threshold\ value\ ;\ Binary = 1 \end{cases} \quad \text{Eq. 2}$$

Binary is a pixel value in the binary image.

Grayscale is a pixel value in the grayscale image.

The threshold value is the parameter designated as grayscale pixel values, equal to either 0 or 1.



(a) (b) (c)

Figure 5 The sample test tube image converted from (a) color to (b) grayscale, and then to (c) binary

1.2.3 Projection Profile

The Projection Profile technique is now applied to the binary image to isolate the test tube holder from the background. A Projection Profile refers to a visual display of the sums of white pixels along either the horizontal or vertical axis of an image. An example of the horizontal and vertical projection profiles of a test tube image is shown in Figure 6(a). Information from the two resulting projection profiles will be used in the next processing step to define four reference points for detecting the outside edges of the test tube holder. This will enable cropping of the image to remove the irrelevant area outside the test tube holder and thereby center and enlarge the test tube in the image. The horizontal and vertical projection profiles formulae are Equation 3 and 4 respectively (Bovik, 2009).

$$\sum_{i=0}^{R-1}\sum_{j=0}^{C-1} if\ Binary_{(i,j)} = 1;\quad H_R = +1 \qquad \text{Eq. 3}$$

H_R is the sum of white pixels in row R

R is the number of rows of the image.

C is the number of columns of the image.

Binary(i,j) is a pixel value in binary image in the position (i,j)

$$\sum_{i=0}^{C-1}\sum_{j=0}^{R-1} if\ Binary_{(i,j)} = 1;\quad V_C = +1 \qquad \text{Eq. 4}$$

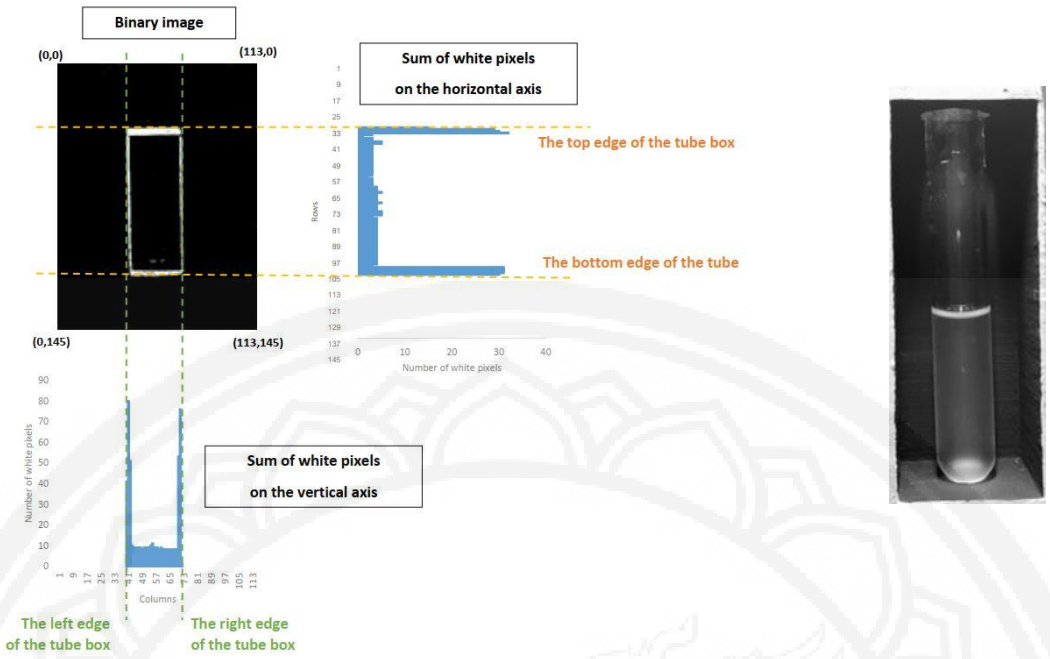
V_C is the sum of white pixels in column C

R is the number of rows of the image.

C is the number of columns of the image.

Binary(i,j) is a pixel value in binary image in the position (i,j)

The sum of white pixels on the horizontal axis is used to define the top and bottom edges of the test tube holder, and the sum of white pixels on the vertical axis is used to define the left and right edges of the test tube holder. On the horizontal axis, the top edge of the test tube holder can be determined as the last row from the top in which the sum of white pixels is still zero, and the bottom edge of the test tube holder is the last row from the bottom in which the sum of white pixels is still zero. Similarly, on the vertical axis, the left edge of the test tube holder is the last column from the left where the sum of white pixels is still zero, and the right edge of the test tube holder is the last column from the right where the sum of white pixels is still zero. Using these four reference points, the image is cropped. A sample result is shown in Figure 6(b).

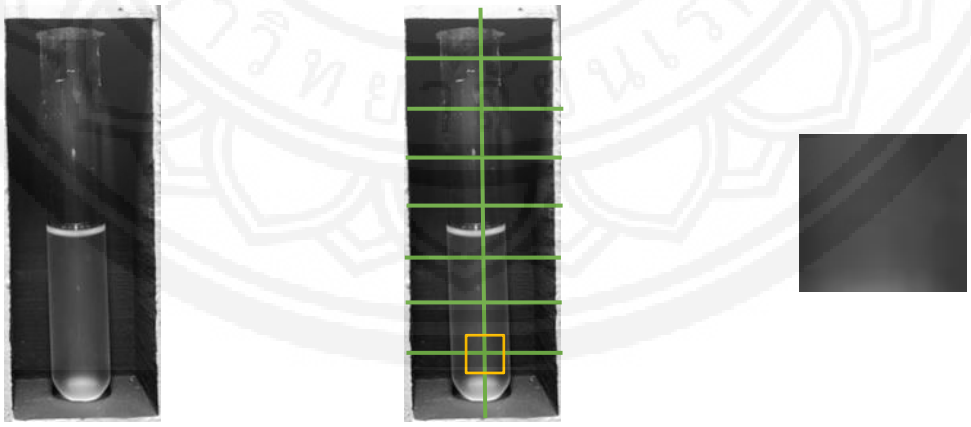


(a) (b)

Figure 6 (a) An example of the horizontal and vertical projection profiles of a test tube holder image (b) the resulting isolated test tube holder after the four reference points are used to crop the background

1.2.4 Segmentation

Since each capped test tube is shaken immediately before taking the photograph, turbidity within the suspension should be uniform. However, the level of the fluid in each test tube photo varies in different photos, as do the precise proportions and positions of elements within the photo. Therefore, in this step each photo is segmented and a specific size area in a specific position that will be applicable in all the photos is used for testing the turbidity. This area is called the Turbidity Testing Zone (TTZ). Each test tube image is segmented into 2 columns and 8 rows, yielding 16 segments. The square-shaped TTZ is 10 pixels x 10 pixels, and it is centered at the intersection of the bottom four segments of the image, as shown in Figure 7.



(a) (b) (c)

Figure 7 An example of test tube image segmentation showing (a) the test tube before segmentation, (b) the test tube after segmentation with the position of the Turbidity Testing Zone outlined in yellow, and (c) an enlarged view of the Turbidity Testing Zone.

The TTZ consists of 100 grayscale pixels, and each pixel value is in the grayscale range of 0–255. The following five numbers are extracted from the TTZ in each image: the minimum pixel value, maximum pixel value, mean pixel value, median pixel value, and mode pixel value (i.e. the most frequently occurring pixel value). These five descriptive statistics, referred to here as key statistics (KS), will next be used in the final component of VBIES model development: analysis and optimization.

1.3 Analysis & Optimization

The image processing component resulted in 2260 images, one for each test tube image. These were randomly divided into two groups. 1800 images (~80%) were used to train the model, and 460 images (~20%) were used to test the model. The training data in turn was divided into four parts to analyze the four different bacterial species: 480 *E. coli* datasets, 384 *E. aerogenes* datasets, 560 *Lactobacillus* sp. datasets, and 384 *B. cereus* datasets. To build the estimation model, four kinds of Machine Learning (ML) algorithms were trialed: Regression Analysis (RA), Artificial Neural Network (ANN), Decision Tree Learning (DTL), and Support Vector Machine (SVM). Each of these algorithms was applied to process the five key statistics in Weka 3.8.6 software. Each of the four different algorithms that were trialed to optimize the bacterial density estimation model processes data differently and the processing method of each is explained below.

The RA is a set of statistical methods used for the estimation of relationships between a dependent variable and one or more independent variables. It can be utilized to assess the modeling of the future relationship between them (Beck, 2017). The RA algorithm used the five KS, the bacterial concentration data, and one of five different mathematical functions to identify the function yielding the best results. All 30 possible combinations of algorithm and KS were tested using all the KS of the four bacterial types, for a total of 120 tests per 1 function. The purpose here was to find the best single combination of algorithm and KS that would give optimal estimation results for the four kinds of bacteria. The five functions used in this study were: Exponential, Logarithm, Linear, and Polynomial. Equations 5–8 show the resulting models for each of the four bacterial types.

E. coli model

$$CFU_E = 1.278 \times 10^{10} - 3.457 \times 10^8 \times min_E - 2.034 \times 10^8 \times max_E + 4.365 \times 10^4 \times min_E^2 + 6.494 \times 10^6 \times min_E \times max_E + 1.443 \times 10^4 min_E^3 - 3.294 \times 10^4 \times min_E^2 \times max_E in \quad \text{Eq.5}$$

Where:

CFU_E is the concentration of viable *E. coli* in the test tube, expressed in CFU/ml.

min_E is the minimum pixel value in the TTZ of *E. coli*.

max_E is the maximum pixel value in the TTZ of *E. coli*.

E. aerogenes model

$$CFU_{En} = 1.921 \times 10^{10} + 3.305 \times 10^8 \times min_{En} - 7.128 \times 10^8 \times max_{En} + 4.891 \times 10^5 \times min_{En}^2 - 6.046 \times 10^6 \times min_{En} \times max_{En} + 7.963 \times 10^6 \times max_{En}^2 + 1.871 \times 10^4 \times min_{En}^2 \times max_{En} - 94.92 \times min_{En} \times max_{En}^2 - 1.883 \times 10^4 \times max_{En}^3 \quad \text{Eq.6}$$

Where:

CFU_{En} is the concentration of viable *E. aerogenes* in the test tube, expressed in CFU/ml.

min_{En} is the minimum pixel value in the TTZ of *E. aerogenes*.

max_{En} is the maximum pixel value in the TTZ of *E. aerogenes*.

Lactobacillus sp. model

$$CFU_L = -2.079 \times 10^9 - 1.739 \times 10^7 \times min_L + 7.702 \times 10^7 \times max_L - 2.284 \times 10^6 \times min_L^2 + 3.316 \times 10^6 \times min_L \times max_L - 1.833 \times 10^6 \times max_L^2 + 1.364 \times 10^4 min_L^2 \times max_L - 1.911 \times 10^4 \times min_L \times max_L^2 + 8667 \times max_L^3 \quad \text{Eq.7}$$

Where:

CFU_L is the concentration of viable *Lactobacillus* sp. in the test tube, expressed in CFU/ml.

min_L is the minimum pixel value in the TTZ of *Lactobacillus* sp.

max_L is the maximum pixel value in the TTZ of *Lactobacillus* sp.

B. cereus model

$$CFU_B = 7.55 \times 10^9 - 8.468 \times 10^7 \times min_B - 2.248 \times 10^8 \times max_B + 1.686 \times 10^6 \times min_B \times max_B + 2.243 \times 10^6 \times max_B^2 - 7651 \times min_B \times max_B^2 - 7631 \times max_B^3 \quad \text{Eq.8}$$

Where:

CFU_B is the concentration of viable *B. cereus* in the test tube, expressed in CFU/ml.

min_B is the minimum pixel value in the TTZ of *B. cereus*.

max_B is the maximum pixel value in the TTZ of *B. cereus*.

The ANN is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. ANN can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of mechanical systems (Zgurovsky, Sineglazov, & Chumachenko, 2021). An example of the ANN algorithm applied to *E. coli* in this study is shown in Figure 8. The ANN algorithm used the five key statistics as the input layer, also called the input factors, which are shown in green in Figure 8. The input layer is followed by the hidden layers, of which there are four in this example, shown in red in Figure 8. Each node in the hidden layers holds one neuron. The job of each neuron is to receive information from all the previous layer nodes and perform a calculation based on a linear function assigned to it by the ANN algorithm and a weighted value that is part of the assigned linear function. Each neuron then sends its calculation result (the same result) to all nodes in the next layer. The input factors in the input layer do not hold a neuron. They simply pass their value into the first hidden layer. The nodes of the last hidden layer send their calculations to the sole node in the final layer, the output layer (Zgurovsky et al., 2021). In the case of this study, this output node contains the calculated concentration of the bacterial solution of one test tube image. This study used a multilayer perceptron ANN algorithm with learning rates of 0.1, 0.2, and 0.3, momentums of 0.3, 0.4, and 0.5, and a training time of 2,000 cycles. Based on these parameters and the KS data of *E. coli* fed to the multilayer perceptron ANN algorithm, the algorithm itself determines that this model should have four hidden layers of 4, 6, 3, and 5 nodes, respectively.

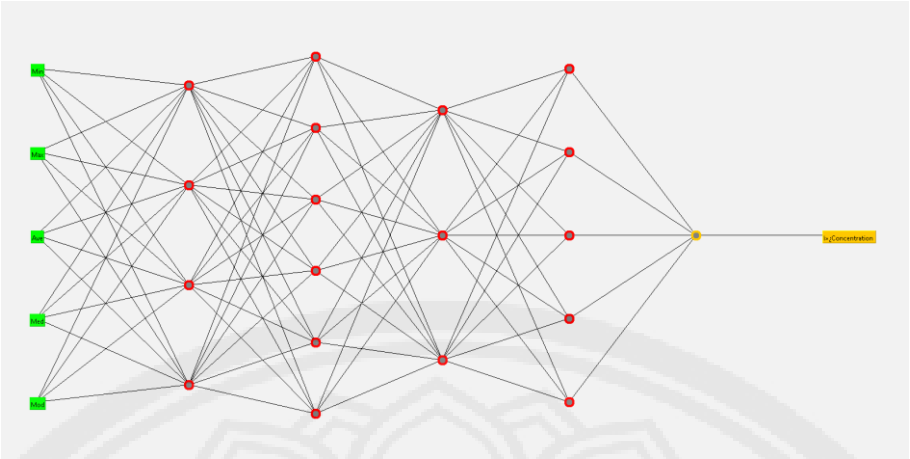


Figure 8 An example of ANN algorithm layers for *E. coli*

The DTL is a flowchart-like structure in which each internal node represents a test on a feature, each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class labels. The paths from the root to the leaf represent classification rules (Cheewaprakobkit, 2019; Lior & Oded, 2008). An example of the DTL algorithm applied to *E. coli* in this study is shown in Figure 9. The DTL algorithm uses the five KS to set a series of condition nodes (shown as gray ellipses in Figure 9, beginning at the top and moving downward) with each condition answerable as only true or false. The result of each condition (true or false) will lead to either a new condition node or a result node, which represents an end to that light of questioning. The first condition is called the root node. The level below the root node is called Level 0, and the level below that is called Level 1, etc (Cheewaprakobkit, 2019; Lior & Oded, 2008). This study used a random DTL algorithm set at a maximum depth level of 10. The resulting model for *E. coli* has four levels containing a total of nine nodes. Each of the five result nodes in Figure 9 gives the concentration of the bacterial solution estimated using a different combination of KSs. Note that even though the mode KS was input to the software, not all KS is used by all bacteria. For example, here the *Escherichia coli* bacteria did not use the mode KS.

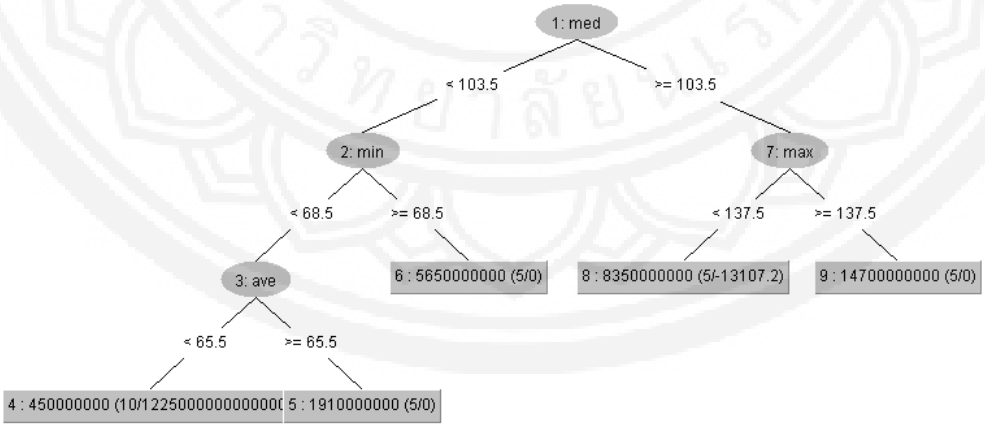


Figure 9 An example of DTL algorithm nodes for *E. Coli*

The SVM is a linear model for classification and regression problems. It can solve linear and non-linear problems and works well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. At first, an approximation of what SVMs do is to find a

decision boundary line (or hyperplane) between data of two classes. SVM is an algorithm that takes the data as an input and then outputs a line that separates those classes if possible (Boyle, 2011). The SVM algorithm shares some similarities with the RA algorithm. SVM also receives the five KS and the bacterial concentration data, and SVM also plots a graph in response. However, the SVM graph does not find trends in mathematical functions like the RA graph but rather SVM classifies the data to separate it with what is called a decision boundary, thus grouping similar data (Boyle, 2011), in this case, similar bacterial densities. The position of known bacterial concentrations on the graph can then be used to predict the unknown bacterial concentrations based on their KS data. This study uses a regression SVM algorithm with the number of kernel evaluations set at 10 and a complexity parameter of 2.

2. VBIES Android Application Development

The completed VBIES model was then developed into a convenient and user-friendly smartphone application that can run on any Android 9.0 (Android Pie) smartphone or earlier version. The VBIES application was built with Android Studio (Version 4.2.2) and OpenCV library (Version 3.4.14).

2.1 User Interface Diagram Design

The user interface diagram of the application that was designed is shown in Figure 10. The application opens with the home page. Pressing Start takes the user to the microbe selection page. Selecting the desired microbe takes the user to test tube photography. Taking a test tube photo brings the user to the photo confirmation page. Acceptance of the photo initiates automatic estimation of the viable bacteria and the user is taken to the result display page, which delivers the estimated concentration of viable bacteria in the test tube.

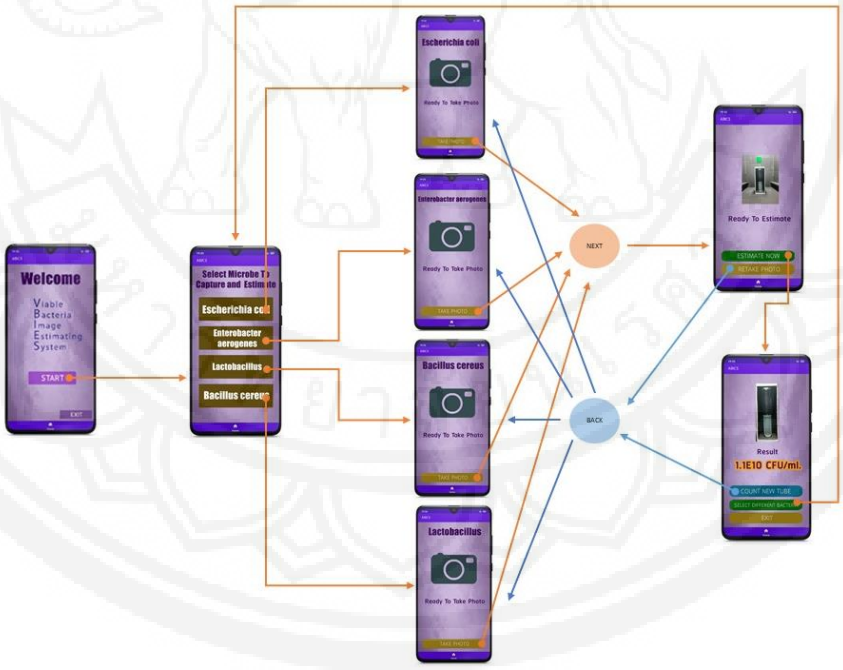


Figure 10 The user interface diagram of the VBIES application

2.2. A Block Diagram Design

A block diagram of the VBIES application shows an overview of the relationship between the processes, an actor, and the system. For this study, the actor is only the type "user" as shown in Figure 11.

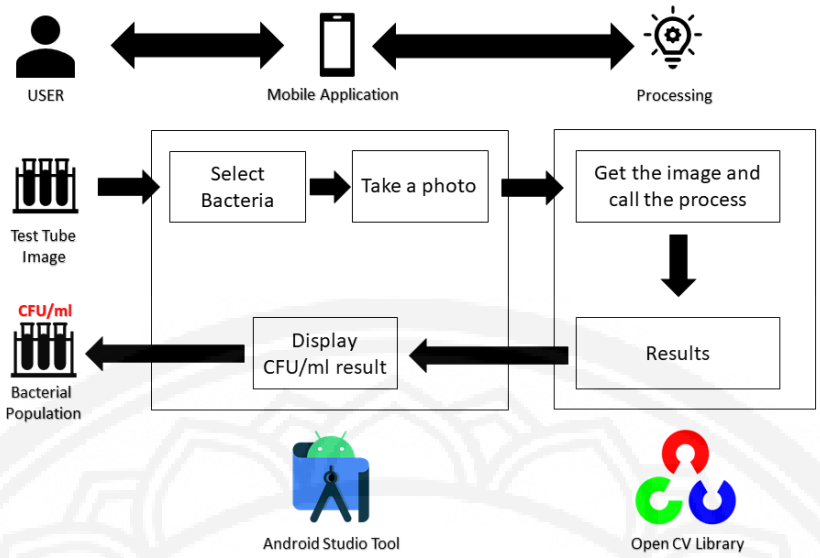


Figure 11 The block diagram of the VBIES application

The completed VBIES application estimates the concentration of viable bacteria in each test tube. The application was evaluated by five researchers in the Department of Microbiology and Parasitology at Naresuan University's Faculty of Medical Sciences. The percentage error of concentrated estimation of viable bacteria in each test tube was calculated according to the formula shown in Equation 9. The average percentage error across all tubes was calculated similarly. In this study, "accuracy" refers to the difference between 100 and the average percentage error.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(Predicted_i - Actual_i)^2}{N}} \quad \text{Eq. 9}$$

RMSE (Root Mean Square Error) is the percentage difference between predicted and actual.

Predicted is the number of bacteria that the VBIES application estimated (CFU/ml).

Actual is the number of bacteria that the traditional method determined (CFU/ml).

N is the total number of test tubes for the bacterial type.

Results

As mentioned earlier, the image processing component resulted in 2260 images, one for each test tube, and these were randomly divided into two groups: the model training dataset (1800 images, ~80% of total), and the model testing dataset (460 images, ~20% of total).

1. Accuracy Results

Once the model was complete, it was run on the testing dataset, and Equation 9 was then applied to determine the average percentage error rate and thereby the accuracy of each model. The resulting accuracy for each bacterial species of all models is shown in Table 3.

Table 3 Accuracy results of the four tested machine learning models when used to estimate the concentration of the four bacteria species

Bacteria	Average Bacterial Concentration	Machine Learning Models								Average Accuracy
		RA		ANN		DTL		SVM		
		VBIES	Accuracy	VBIES	Accuracy	VBIES	Accuracy	VBIES	Accuracy	
E. coli	5.25×10^9	5.15×10^9	98.02%	5.16×10^9	98.37%	5.20×10^9	99.03%	5.17×10^9	98.46%	**98.47%**
E. aerogenes	3.58×10^9	3.66×10^9	95.06%	3.52×10^9	99.24%	3.58×10^9	99.99%	3.37×10^9	94.24%	**97.13%**
Lactobacillus sp.	2.05×10^8	1.57×10^8	76.43%	1.93×10^8	95.59%	1.97×10^8	96.08%	1.73×10^8	84.47%	**88.14%**
B. cereus	3.04×10^8	2.73×10^8	90.72%	2.90×10^8	95.34%	2.89×10^8	95.17%	2.76×10^8	90.75%	**93.00%**
Average Accuracy	-	-	**90.06%**	-	**97.14%**	-	**97.57%**	-	**91.98%**	**94.19%**

Table 3 shows that the best accuracy of 97.57% was achieved using the DTL model. All the test tubes used in this study were by design diluted to one of the following optical densities at wavelength 600 nm: 0.06, 0.08, 0.2, 0.4, 0.6, and 0.8. The optical density of a solution will affect its estimation accuracy. The accuracy results from each of the optical densities when using the optimal model (DTL) are shown in Table 4.

Table 4 Accuracy results of concentration estimating of the four bacterial species in the six optical density ranges.

Bacteria	Optical density at wavelength 600 nm.						Average
	0.06	0.08	0.2	0.4	0.6	0.8	
E. coli	99.39%	99.06%	99.56%	99.12%	99.02%	98.02%	99.03%
E. aerogenes	100%	100%	99.99%	100%	100%	99.92%	99.99%
Lactobacillus sp.	97.64%	95.86%	96.94%	96.66%	95.12%	94.26%	96.08%
B. cereus	96.29%	96.25%	95.46%	95.75%	95.12%	92.15%	95.17%
Average	98.33%	97.79%	97.99%	97.88%	97.32%	96.09%	97.57%

2. Time results

Table 5 shows the processing time required by each of the four machine-learning models. The best single result occurred with the RA model at 3.42 seconds. However, the model with the overall shortest average processing time was the DTL model, with an average of 4.19 seconds. As mentioned earlier, conventional methods to estimate the number of bacteria in a pure source normally require more than 48 hours. In light of its best performance in both accuracy and speed, the DTL model was chosen for use in the VBIES model.

Table 5 Processing time of the four tested machine learning models

Model	Minimum (seconds)	Maximum (seconds)	Average (seconds)
Regression Analysis (RA)	3.42	5.09	4.21
Artificial Neural Network (ANN)	4.16	6.20	4.59
Decision Tree Learning (DTL)	3.51	5.15	4.19
Support Vector Machine (SVM)	4.33	6.09	5.16

3. Evaluation results

Once the VBIES application was complete, it was sent for real-life testing to five different researchers in the Department of Microbiology and Parasitology at Naresuan University's Faculty of Medical Sciences. The

five researchers tried out the VBIES application and evaluated it, and their assessment is shown in Table 6. The average rating for all criteria came to 4.33 out of 5.00 possible points, representing a grade of 86.60% from the researchers.

Table 6 Evaluation results of the VIBES application

App Evaluation Criteria	Very good	Good	Fair	Poor	Very poor	$\bar{x}$	S.D.
Ease of use	2 votes (40%)	2 votes (40%)	1 vote (20%)	0 (0%)	0 (0%)	4.20	0.836
Appropriate placement of menus and buttons	3 votes (60%)	2 votes (40%)	0 (0%)	0 (0%)	0 (0%)	4.60	0.547
Modernity of design	4 votes (80%)	0 (0%)	1 vote (20%)	0 (0%)	0 (0%)	4.60	0.894
Easy to download and install	2 votes (40%)	1 vote (20%)	1 vote (20%)	1 vote (20%)	0 (0%)	3.80	1.303
The application reduces work time.	5 votes (100%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	5.00	0.00
The application works well.	4 votes (80%)	1 vote (20%)	0 (0%)	0 (0%)	0 (0%)	4.40	0.894
Averages						**4.33**	**0.746**

Discussion

The results of the VBIES model estimating the concentration of bacteria in a pure liquid source were well aligned with the lab-determined concentration, which compared VBIES application predictions results and laboratory experiment results by the expert, demonstrating a good level of accuracy. Errors are more common when the bacterial concentration is high. For example, 71% of the errors that occurred were in bacterial suspensions with turbidity higher than 0.8. Another important factor affecting accuracy is that each test tube needs to be shaken just before the photo is taken, since the bacteria are suspended in the solution and have a tendency to settle, particularly in the case of *Lactobacillus* sp. which settles relatively quickly.

Both the experimental time results of the VBIES application and the hands-on assessment by outside researchers demonstrate that VBIES greatly reduces the work time needed for estimating bacterial concentration. The application uses the DTL model because DTL had both the lowest average processing time and the highest average accuracy rate. RA was not too far behind DTL in speed, because the two have a similar level of computational complexity. ANN and SVM both have higher computational complexity, so they both take longer (Jena & Dehuri, 2020; Mohammadiun et al., 2021). In computer science, the computational complexity of an algorithm is the number of resources that the computer requires to process the algorithm, mostly time and memory. The higher the complexity, the more time and memory are used (Jena & Dehuri, 2020).

The four ML algorithms yield different estimation results, with the highest accuracy from DTL closely followed by ANN. ANN is generally good for image processing and video processing (Jena & Dehuri, 2020), but it is prone to overfitting data and it requires significant computing power (Shetty, Acharya, Narendra, & Prajual, 2020). The accuracy rates of RA and SVM cannot compete with the other two algorithms (DTL and ANN) when processing this data. DTL is generally good for use with data with relatively low complexity because

highly complex data produces highly complex trees that are difficult to code into an application (Shetty et al., 2020).

Users showed good overall satisfaction with VBIES in the evaluation questionnaire, giving the application an average of 4.42 points out of five across all evaluation criteria. In particular, the users report high satisfaction with VBIES for its ability to reduce their work time. The evaluation question on downloading and installation was the only one that received relatively weaker responses. This was because one user said that he wished to download the app from Google Play rather than using the standard APK installation file that was delivered, and another user said that he wished to run the app on his iOS device. This comment reflects the fact that the app does not run on iOS, but developing an iOS version in the future is a possibility. Also, the app is newly developed and the option of downloading from Google Play is not yet available.

A limitation of this study was the limited number of bacterial species. *E. coli*, *E. aerogenes*, *Lactobacillus* sp., and *B. cereus* were selected because these are the most commonly used indicators for the quality of raw milk. Additional species of bacteria could be added to the application by culturing them and then handling them with the same data collection procedures used for the current four bacterial species.

Conclusion and Suggestions

The Viable Bacteria Image Estimating System (VBIES) successfully achieves its objective of reducing the time required to accurately estimate the number of bacteria in a pure liquid source. The conventional method typically requires 24-48 hr with the necessary culturing. In contrast, VBIES can dramatically reduce this time to only 3-6 seconds per test tube. VBIES is best suited for analyzing suspensions with uniformly fine turbidity, whereas suspensions with large particle aggregates will yield lower accuracy. Another attraction of the VBIES application is its ease of use, making the process accessible and virtually foolproof for all users. Since almost all of the process is automated, the user only needs to take a photo of the test tube and confirm. No parameters need to be adjusted manually as was necessary for some previous studies, overcoming this potential source of user error. The image processing techniques and the machine learning model integrated here result in the highest accuracy of 97.57% using the DLT algorithm model. The RA, ANN, DLT, and SVM models work with the same data, but the results vary in speed and accuracy depending on the process of each model. The VBIES application is therefore a reliable and robust tool that saves a substantial amount of time when estimating the number of bacteria in a liquid pure source.

Additional experiments in the future might be able to improve further on the current results. For example, the key statistic values (minimum, maximum, mean, median, and mode) are important variables in the Analysis and Optimization component. Testing additional statistics besides the five already trialed might reveal an even better option, as might testing additional models besides the four used in this study, for example, K-Nearest Neighbors or deep learning. Thus, there are opportunities for expanding the current research.

Acknowledgments

The authors are grateful to Ms. Nootsara Yinyom of the Faculty of Medical Science, Naresuan University for her invaluable assistance in the laboratory. Thank you to the Computer Vision and Human Interaction

Technologies Laboratory and the Faculty of Engineering, Naresuan University for kindly providing support of their facilities and scholarship for the current study. Thank you also to Mr. Paul Freund of Naresuan University Writing Clinic (DIALD) for his help editing this manuscript and to Mr. Roy I. Morien of the Naresuan University Graduate School for his final editing of the grammar, syntax and general English expression in this manuscript.

References

Agricultural Product Quality Standards Act (2017). *(C.9)*. Thailand: Ministry of Agriculture and Cooperatives.

Atmaja, R. D., Murti, M. A., Halomoan, J., & Suratman, F. Y. (2016). An image processing method to convert RGB image into binary. *Indonesian Journal of Electrical Engineering and Computer Science, 3*(2), 377-382-382. http://dx.doi.org/10.11591/ijeecs.v3.i2.pp377-382

Beal, J., Farny, N. G., Haddock-Angelli, T., Selvarajah, V., Baldwin, G. S., Buckley-Taylor, R., . . . Emory. (2020). Robust estimation of bacterial cell count from optical density. *Communications Biology, 3*(1), 512. http://dx.doi.org/10.1038/s42003-020-01127-5

Beck, V. L. (2017). *Linear Regression: Models, Analysis, and Applications*. Hauppauge, New York: Nova Science.

Bovik, A. C. (2009). *The essential guide to image processing* (2nd ed.). USA: Academic Press.

Boyle, B. H. (2011). *Support Vector Machines: Data Analysis, Machine Learning, and Applications*. New York: Nova Science.

Cangliang, S., & Yifan, Z. (2022). *Introductory Microbiology Lab Skills and Techniques in Food Science*. London: Academic Press.

Cheewaprakobkit, P. (2019). Predicting Student Academic Achievement by Using the Decision Tree and Neural Network Techniques. *Human Behavior, Development and Society, 12*(2), 34-43.

Dong, M., Yang, S., Yang, X., Xu, M., Hu, W., Wang, B., . . . Sun, G. (2022). Water quality drives the distribution of freshwater cable bacteria. *The Science of the Total Environment, 841*, 156468. http://dx.doi.org/10.1016/j.scitotenv.2022.156468

Godoy, A. C., Santos, O. O., Nakano, A. Y., Siepmann, D. A. B., Schneider, R., & Pfrimer, F. W. D. (2018). Snapshots Analyses for Turbidity Measurements in Water. *Water, Air, and Soil Pollution, 229*(12), 1-11. http://dx.doi.org/10.1007/s11270-018-4034-4

Jayapal, M., Jagadeesan, H., Krishnasamy, V., Shanmugam, G., Muniyappan, V., Chidambaram, D., & Krishnamurthy, S. (2022). Demonstration of a plant-microbe integrated system for treatment of real-time textile industry wastewater. *Environmental Pollution, 302*, 119009. http://dx.doi.org/10.1016/j.envpol.2022.119009

Jena, M., & Dehuri, S. (2020). Decision Tree for Classification and Regression: A State-of-the Art Review. *Informatica, 44*(4), 291. http://dx.doi.org/10.31449/inf.v44i4.3023

Kanan, C., & Cottrell, G. W. (2012). Color-to-Grayscale: Does the Method Matter in Image Recognition?. *PLoS ONE, 7*(1), 1-7. http://dx.doi.org/10.1371/journal.pone.0029740

Kumar, L., Afzal, M. S., & Ahmad, A. (2022). Prediction of water turbidity in a marine environment using machine learning: A case study of Hong Kong. *Regional Studies in Marine Science, 52*, 102260. http://dx.doi.org/10.1016/j.rsma.2022.102260

Lior, R., & Oded, Z. M. (2008). *Data Mining With Decision Trees: Theory And Applications*. Singapore: World Scientific.

Maturin, L., & Peeler, J. T. (2001). *Laboratory Methods (Food). In BAM Chapter 3: Aerobic Plate Count.* Berlin: Silver Spring.

Mohammadiun, S., Hu, G., Alavi Gharahbagh, A., Mirshahi, R., Li, J., Hewage, K., & Sadiq, R. (2021). Optimization of integrated fuzzy decision tree and regression models for selection of oil spill response method in the Arctic. *Knowledge-Based Systems, 213*, 106676. http://dx.doi.org/10.1016/j.knosys.2020.106676

Moore-Colyer, M. J. S., & Fillery, B. G. (2012). The effect of three different treatments on the respirable particle content, total viable count and mould concentrations in hay for horses. *EAAP Scientific Series, 132*(1), 101-106.

Novik, G., Savich, V., & Meerovskaya, O. (2018). *Geobacillus Bacteria: Potential Commercial Applications in Industry, Bioremediation, and Bioenergy Production.* London: IntechOpen.

Osano, E., Bessho, M., Asai, Y., & Takei, M. (1979). Evaluation of diluents for the recovery of total viable microorganisms in human oral material (author's transl). *Aichi Gakuin Daigaku Shigakkai shi, 17*(1), 7-18.

Pare, S., Kumar, A., Singh, G. K., & Bajaj, V. (2020). Image Segmentation Using Multilevel Thresholding: A Research Review. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering, 44*(1), 1-29. http://dx.doi.org/10.1007/s40998-019-00251-1

Prockop, D. J. E., Bunnell, B. A. E., Phinney, D. G. E., Pochampally, R., & Walker, J. P. E. (2008). *Colony Forming Unit Assays for MSCs.* Totowa, N J: Humana Press.

Raveendran, S., Edavoor, P. J., Kumar, Y. B. N., & Vasantha, M. H. (2018). *Design and Implementation of Reversible Logic based RGB to Gray scale Color Space Converter.* United States: IEEE.

Renganathan, S., & Olubukola Oluranti, B. (2021). Utilization of Microbial Consortia as Biofertilizers and Biopesticides for the Production of Feasible Agricultural Product. Biology, 10(1111), 1111-1111. http://dx.doi.org/10.3390/biology10111111

Sarrafzadeh, H., Aghajari, G., & Shanbehzadeh, J. (2010, 2010/01/01/). *A Text Localization Algorithm in Color Image via New Projection Profile.* New Zealand: Australia/Oceania.

Shetty, D. K., Acharya, U. D., Narendra, V. G., & Prajual, P. J. (2020). Intelligent System to Evaluate the Quality of DRC using Image Processing and then Categorize using Artificial Neural Network (ANN). *Indian Journal of Agricultural Research, 54*(6), 716-723. http://dx.doi.org/10.18805/IJARe.A-5374

Tao, S., Luze, Z., & Long-Fei, W. (2014). A Method for Quantitative Determination of the Number of Magnetosomes in Magnetotactic Bacteria by a Spectrophotometer. *IEEE Transactions on Magnetics, 50*(11), 1-4. http://dx.doi.org/10.1109/TMAG.2014.2323953

Taylor, R. H., Allen, M. J., & Geldreich, E. E. (1983). Standard plate count: A comparison of pour plate and spread plate methods. *Journal (American Water Works Association), 75*(1), 35-37. http://dx.doi.org/10.1002/j.1551-8833.1983.tb05055.x

Tomasiewicz, D. M., Hotchkiss, D. K., Reinbold, G. W., Read, R. B., & Hartman, P. A. (1980). The Most Suitable Number of Colonies on Plates for Counting 1. *Journal of food protection, 43*(4), 282-286. http://dx.doi.org/10.4315/0362-028X-43.4.282

Velier, M., Chateau, A. N. N. E. L. I. N. E., Malenfant, C., Ouffai, S., Calmels, B., Chabannon, C., & LemariÉ, C. (2019). Validation of a semi automatic device to standardize quantification of Colony-Forming Unit (CFU) on hematopoietic stem cell products. *Cytotherapy, 21*(8), 820-823. http://dx.doi.org/10.1016/j.jcyt.2019.06.005

Yunxia, W., Lijuan, J., Cuizhi, L. I., Zhiyong, L. U., & Lijun, L. I. U. (2021). Evaluation on Uncertainty of Detection Results of Aerobic Plate Count. *Asian Agricultural Research, 13*(6), 59-62. http://dx.doi.org/10.19601/j.cnki.issn1943-9903.2021.06.014

Zgurovsky, M., Sineglazov, V., & Chumachenko, E. (2021). *Artificial Intelligence Systems Based on Hybrid Neural Networks*. Switzerland: Springer Cham.

Zhu, G., Yan, B., Xing, M., & Tian, C. (2018). Automated counting of bacterial colonies on agar plates based on images captured at near- infrared light. *Journal of Microbiological Methods, 153*, 66-73. http://dx.doi.org/10.1016/j.mimet.2018.09.004