

Article

The Study of Image Quality Effect on Model Performance for Bacteria Classification

Treesukon Treebupachatsakul^{1,a}, Wanwalee Chomkwah¹, Tananan Tanpatanan¹, and Suvit Poomrittigul^{2,b,*}

¹ Department of Biomedical Engineering, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 1 Chalong Krung 1 Alley, Ladkrabang, Bangkok 10520, Thailand

² School of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, 1 Chalong Krung 1 Alley, Ladkrabang, Bangkok 10520, Thailand

E-mail: ^atreesukon.tr@kmitl.ac.th, ^{b,*}suvit@it.kmitl.ac.th (Corresponding author)

Abstract. One of the key requirements for supervised learning in deep learning model construction is the dataset for training and validation. For gathering the dataset, obtaining various image qualities from different resources is unavoidable, and this has been considered to affect the supervised model performance. This research proposes to demonstrate the effect of image quality involving high and standard datasets obtained from 2 different resources on the performance of models. The various cell characteristics with gram-positive and gram-negative bacteria datasets were challenged for trial. These different datasets were matched and contributed to 5 cases; case 1: train and test with high-quality images, case 2: train with high-quality images and test with standard quality images, case 3: train and test with images of standard quality, case 4: train with standard-quality images and test with high-quality images, and case 5: train and test with combining these two image qualities. Pre-trained CNN models were implemented to prove the purpose with and without stratified K-fold cross-validation. The results of retrained models showed that the high-performance models require high-quality datasets obtained from the same resource as the testing set, which yield more than 90% of all performance evaluation metrics when tested on challenging unseen datasets. This study provides valuable insights for building high-performance models that can be applied to automate microbiology diagnostics, impacting public health and clinical practice.

Keywords: Bacteria classification, deep learning, image quality, convolutional neural network (CNN).

ENGINEERING JOURNAL Volume 29 Issue 1

Received 4 June 2024

Accepted 9 January 2025

Published 31 January 2025

Online at <https://engj.org/>

DOI:10.4186/ej.2025.29.1.11

1. Introduction

Bacteria identification is a pathological approach to identifying pathogens from patient specimens or foodborne contamination. The samples can be either a body fluid, for instance, blood and urine, or tissue from a biopsy. However, the constraints of the traditional approach include the proceeding time and the pathologist's expertise [1]. This process is time-consuming because it requires the procedure involving cell culture followed by identifying the differentiated morphology of the cell such as sizes, shapes, cell arrangements, gram stain, and colonization observed under the microscope [2]. Then biochemical tests are carried out to identify the bacteria. These steps are performed manually in the laboratory, and the process requires the workers' expertise to analyze the results precisely, which can be prone to human error due to subjective human interpretation. To address these limitations, applying computer vision is a promising technique, which has been developed and successfully applied in microscopic image classification [3-6]. Classification is one technique in deep learning in which the models autonomously extract relevant and distinguishing features to make a decision based on the input. Because deep learning models require information from input to accomplish the tasks, the input dataset is an important factor in constructing the high performance of deep learning models.

The microscopic bacterial cell images have been widely used to implement several state-of-the-art deep learning models, which are open source such as DIBas [4-9], and even private dataset images collected on their own, have different image quality with different resolutions [10]. This is typically accomplished through various techniques of database preparation, such as cell staining and the utilization of specialized equipment for image acquisition. The bacteria can be categorized into two groups of gram-positive and gram-negative according to the different peptidoglycan layers of the cell wall. The gram-positive bacteria contain a thick layer of peptidoglycan retained purple of crystal violet and gram-negative bacteria contain a thin layer of peptidoglycan retained red of safranin [11]. The bacteria cells, shape, and arrangement can be visualized under the microscope by which the image dataset was captured [4, 5]. Since the performance of deep learning models is significantly influenced by the quality and various resolutions of the images [12], it is unsurprising that the development of an effective model for various quality and resolution bacteria image classification has not yet been achieved.

Therefore, this research aims to demonstrate the effect of different quality images used to train and evaluate the performance of the deep learning model-based Convolutional Neural Network (CNN) architectures by employing two image datasets to observe the model's behaviors and their limitations. This purpose is valuable information for improving the deep learning model applied with the further application that can be used in the proposed way, e.g. multiple microbial identification under

lacking the sufficient dataset for training. Multiclass bacteria species including *Escherichia coli*, *Lactobacillus casei*, *Lactobacillus delbrueckii*, *Micrococcus spp*, and *Staphylococcus aureus* were selected for challenging examination of the models due to their diverse characteristics such as shape, size, arrangement, and gram strained. *E. coli* and *S. aureus* are used as indicators to assess hygiene conditions and sanitation in hospitals and for foodborne disease evaluation due to their simplicity and are more affordable to detect than other pathogens [13-15]. *E. coli* is a gram-negative bacillus, which has a rod shape. The same genus of *Lactobacillus* but a different group of species; *L. casei* and *L. delbrueckii* are gram-positive straight rod-shaped often with square ends occurring singly, in pairs, or chains [16]. They produce lactic acid as the major end product of fermentation [17]. *Micrococcus spp* is gram-positive cocci arranged in pairs, tetrads, or clusters but not in chains [18]. *Staphylococcus aureus* is a gram-positive spherical shape (cocci) and grows in clusters, pairs, and occasionally in short chains [19]. It is an opportunistic pathogen most associated with skin and soft-tissue infections [20, 21]. There are 2 different types of microscopic image qualities defined, including high quality and standard quality, which were prepared from different resources. The strategy of this study involves the variation of images for the training and the testing datasets, which were arranged into 5 cases; case 1: train and test with high-quality images, case 2: train with high-quality images and test with standard-quality images, case 3: train and test with the standard quality images, case 4: train with standard images and test with high-quality images, and case 5: train and test with combining images quality. The cross-validation technique was performed to ensure no overfitting and the dataset bias caused the model performance. The results were comparable with no cross-validation accomplishment. The unseen dataset was used to test the trained models simulating challenges that imitate real-world applications.

2. Literature Review

Many published studies reveal that the deep learning model can classify multi-class bacteria images under supervised learning of model recognition [3-9]. The overview of machine learning and deep learning algorithms has been compared and discussed by Y. Wu [22]. They presented the effectiveness of pre-trained Convolutional Neural Network (CNN) architectures including AlexNet, VGGNet, Inception networks, Residual Networks, and Densely Connected Convolutional Networks to classify the multi-class of bacteria. The pre-trained method leverages knowledge gained from solving different image classification tasks to accurately classify microbial images. As a result, the need for large and diverse training data is greatly reduced. The accuracy by DensNet121 yields the optimal performance, achieving an accuracy of 99.08%, precision of 99.06%, recall of 99.00%, and an F1-score of 98.99% [22]. B. Zieliński, et al. [4] reported the model involving VGG-M and VGG-VD of CNN architecture can successfully

classify the multi-class bacteria image. In addition, publications are reporting on the capability of VGG-16 in multi-class image classification [23, 24]. Therefore, we included the pre-trained models VGG-16, VGG-19, LeNet-5, InceptionV3, ResNet50, and DenseNet121 to construct comparable models for exploring how image quality affects model performance. Recently, further studies on clinical bacterial datasets aimed at detecting and classifying pathogenic have been developed by X. Wang, et al. [25]. They employed object detection and segmentation to localize and classify the multi-class of pathogenic gram-positive and gram-negative bacteria for the purpose of Microbiological Rapid On-Site Evaluation (M-ROSE). In addition, the technique of phase-contrast microscopy capturing time-lapses bacterial growth in microfluidic chip traps combined with deep learning is implemented to classify bacteria species relevant to human health as discussed by E. Hallstro, et al. [26]. Thus, the deep learning model has the potential to distinguish bacteria species in order to enhance the quality and accuracy of identification, reduce the processing time of the classification step, and minimize human errors, resulting in clinically achievable and applicable diagnosis [1, 25-28]. Furthermore, bacteria classification by deep learning can be applied more extensively, not only in the medical field but also in foodborne bacteria and other biotechnology applications [29].

The key requirement for successful deep learning model application is the training data, however, the literature reviews involving deep learning implemented for microorganism classification reveal the challenge under the limitation of quality and quantity of datasets for constructing effective models [1-3]. The technique to overcome the limitations of bacterial dataset quantification is augmentation, which is the process of increasing the number of datasets by automatically or manually generating a new dataset from the original dataset, with the underlying category unchanged [13, 30, 31].

3. Materials and Methods

3.1. Dataset Acquisition for Bacterial Classification

The bacteria images were obtained from 2 resources:

3.1.1. The open source dataset

The bacteria image dataset has been downloaded from Digital Image of Bacterial Species (DIBaS). In this study, we selected 5 different types of bacteria: *Escherichia coli*, *Lactobacillus casei*, *Lactobacillus delbrueckii*, *Micrococcus spp*, and *Staphylococcus aureus*. The cells were stained using Gram's staining method [11]. The images were taken with Olympus CX31 Upright Biological Microscope equipped with a SC30 camera (Olympus Corporation, Japan). They were evaluated using a 100 times objective under oil immersion (Nikon50, Japan) [4]. These images have a resolution of 2048x1532 pixels before splitting.

3.1.2. The bacterial images collected by authors

The digital images of 5 bacteria strains were taken from our laboratory at School of Engineering, King Mongkut's Institute of Technology Ladkrabang: *Escherichia coli* TISTR 527, *Lactobacillus casei* TISTR 1463, *Lactobacillus delbrueckii* TISTR 1339, *Micrococcus spp* TISTR 1404, and *Staphylococcus aureus* TISTR 746. These bacteria strains were obtained from Thailand Institute of Scientific Technological Research (TISTR). The bacteria were stained following the Gram's procedure [11]. The digital image dataset was further collected under a microscope (Optika B-292) using a magnification of 100 of the object lens, equipped with a camera (Optikam B3) incorporated with a 1.3 MP CMOS sensor. Our images have a resolution of 1966x1474 pixels before image splitting.

Considering the quality of images between DIBaS dataset and the images collected from our laboratory, the image of DIBaS was certainly higher resolution and contained less image noise. Moreover, the color of the cell on images of the DIBaS dataset was more clearly apparent. This is because of the difference in the specification of tools including the microscope and equipped camera used to collect the image datasets. Therefore, the image of DIBaS open source was defined as a high-quality image dataset. Our image dataset offers standard quality. The examples of the image dataset of high-quality and standard quality of each class are shown in Fig. 1 and Fig. 2, respectively. The number of datasets was increased by splitting each original image at the defined split size of 200x200 pixels for both resources.

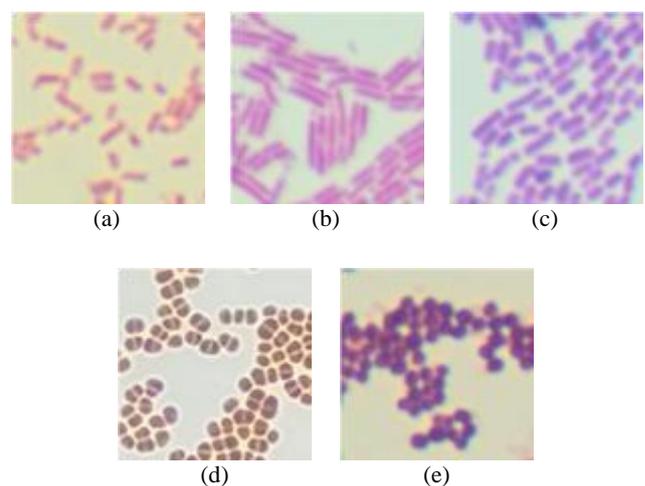


Fig. 1. The examples of high-quality bacteria images: (a) *Escherichia coli*, (b) *Lactobacillus casei*, (c) *Lactobacillus delbrueckii*, (d) *Micrococcus spp*, and (e) *Staphylococcus aureus*.

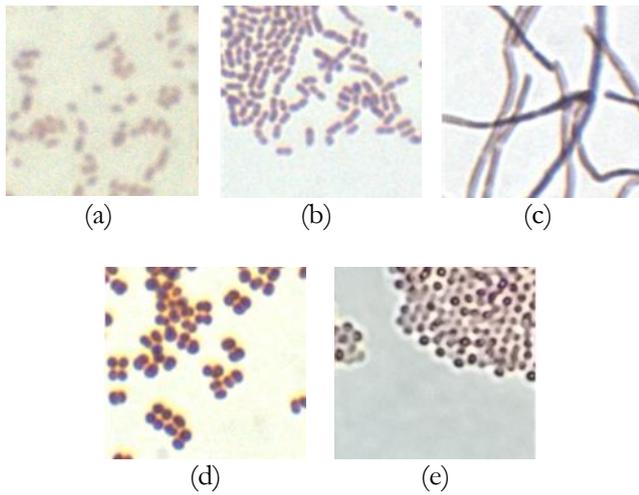


Fig. 2. The examples of standard-quality bacteria images: (a) *Escherichia coli* TISTR 527, (b) *Lactobacillus casei* TISTR 1463, (c) *Lactobacillus delbrueckii* TISTR 1339, (d) *Micrococcus spp* TISTR 1404, and (e) *Staphylococcus aureus* TISTR 746.

The unqualified split images were excluded from the dataset, whether the unsatisfied size or no cell contained on an image. The image dataset of each class was balanced in the range of 1,000 to 1,300 images to reduce the bias of the unfair dataset. The dataset was allocated into three categories: training, validation, and unseen. As shown in Fig. 3, the workflow diagram of dataset preparation. 90% of the dataset of each class was split into 80%:20% for training and validating. Another 10% of the dataset was used for unseen tests. This unseen dataset was not imported to train and validate the model. It was randomly selected and used to evaluate the model's performance. For non-cross-validated models, the unseen dataset contained 100 bacteria images manually selected from each class randomly. For cross-validated models, the allocation of unseen datasets was automated by the Python code by selecting 10% of all bacteria images from each class.

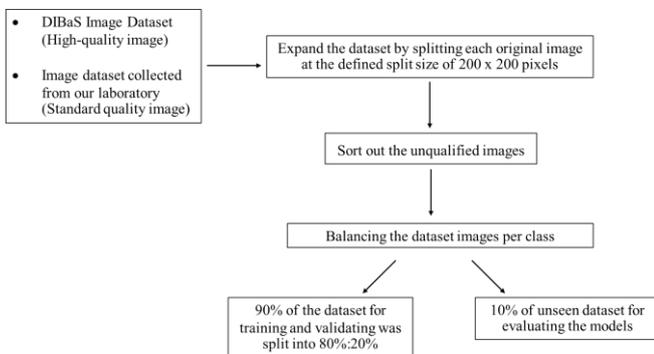


Fig. 3. Dataset acquisition workflow diagram.

3.2. Deep Learning Model Construction

The deep learning models for image classification were implemented in a Python-based workflow called

TensorFlow. To study the effect of image quality on model performance, we built the models in 6 CNN architectures: LeNet-5, VGG-16, VGG-19, InceptionV3, ResNet50, and DenseNet121. We executed them to train and evaluate their efficiency. The model was applied with the dropout parameter of 0.5, batch size of 32, and learning rate of 0.001 as the default values. The epoch of training was set at 50. The stratified K-fold cross-validation was conducted to compare with the results obtained by the model without cross-validation.

3.2.1. CNN architectures

The different complexity of CNN architectures was considered to impact the prediction performance under the difference of image quality datasets. Therefore, 6 CNN architecture of LeNet-5, VGG-16, VGG-19, InceptionV3, ResNet50, and DenseNet121 were examined.

LeNet-5 is a simple CNN architecture developed by Lecun, et al. [32]. The configuration of LeNet-5 consists of 8 layers: the input layer, 2 convolutional layers, 2 average pooling layers, 2 fully connected layers, and the output layer with softmax activation function [32]. The layers apart from the input and the output layer are defined as the hidden layers, which are involved in the feature extraction and classification [33]. The configuration of LeNet-5 is illustrated in Fig. 4.



Fig. 4. LeNet-5 architecture. Block's colour represents a layer type: input/output layer (red), convolutional layer (navy), average pooling layer (green), and fully connected layer (yellow).

VGG-16 and VGG-19 are a subset of CNN architecture called VGGNET. They stand for visual geometry groups of 16 and 19, respectively. The configuration of VGG-16 consists of an input layer, 13 convolutional layers with Rectified Linear Unit (ReLU) activation function, 5 max pooling layers, 3 fully connected layers, and an output layer with softmax activation function [34]. The configuration of VGG-19 is similar to the configuration of VGG-16 except for the number of convolutional layers with ReLU activation function changes to 16 layers [35]. In other words, the total number of layers is 23 for VGG-16 and 26 for VGG-19. The configurations of VGG-16 and VGG-19 architectures are illustrated in Fig. 5(a) and 5(b), respectively.

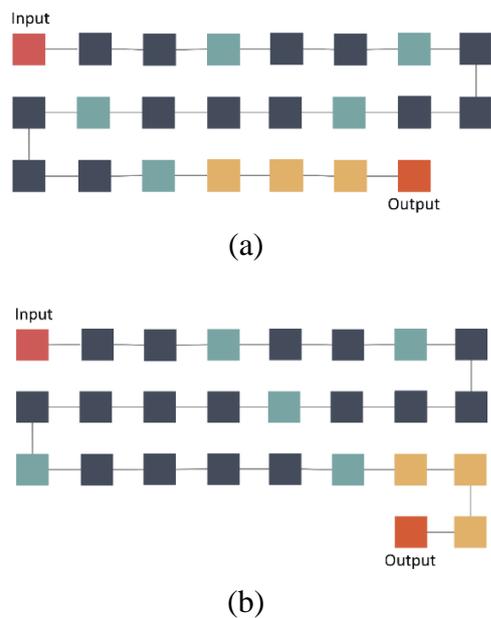


Fig. 5. Architectures of (a) VGG-16, and (b) VGG-19. The block's color represents a layer type: input/output layers (red), convolutional layers (navy), max pooling layers (green), and fully connected layers (yellow).

InceptionV3 is the third generation of Google-owned CNN architecture, launched after GoogLeNet (Inception-v1) and Inception-v2 [35]. It contains 42 layers: 5 convolutional layers, 2 max pooling layers, 1 average pooling layer, 1 dropout layer, the output layer with softmax activation function, and 11 inception modules [36, 37]. For inception modules, consist of multiple parallel convolutional layers and pooling layers, which enable the integration of features in multiscale [37]. Figure 6 illustrates the configuration of InceptionV3.

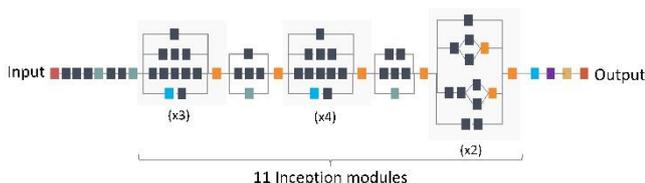


Fig. 6. InceptionV3 architecture. Block's color represents a layer type: input/output layers (red), convolutional layers (navy), max pooling layers (green), average pooling layers (sky blue), concatenate layers (orange), dropout layer (violet), and fully connected layer (yellow). Adapted from Z.-K. Chai, et al. [37].

ResNet50 is defined as a CNN architecture with deep residual connection, developed by Kaiming and his team from Microsoft Research [38]. This model is built by adding shortcut connections to each building block of convolutional layers in the architectures to obtain a compromised training error, instead of stacking the layers like VGG and LeNet-5 network [38]. Each building block consists of 3 convolutional layers [38]. The stacking of the convolutional layers in each block is in a bottleneck

fashion in which the second layer in the block has the largest filter size of 3x3 and the others have the smaller filter size of 1x1 [38]. The number of 50 represents the total number of layers in the architecture apart from the input layer and the output layer with softmax activation function: 48 convolutional layers, 1 max pooling layer, and 1 average pooling layer [39]. Fig. 7 depicts the structure of ResNet50 architecture.

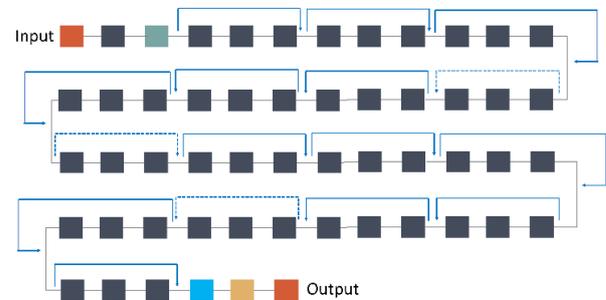


Fig. 7. ResNet50 architecture. Block's color represents a layer type: input/output layers (red), convolutional layers (navy), max pooling layer (green), average pooling layer (sky blue), and fully connected layer (yellow).

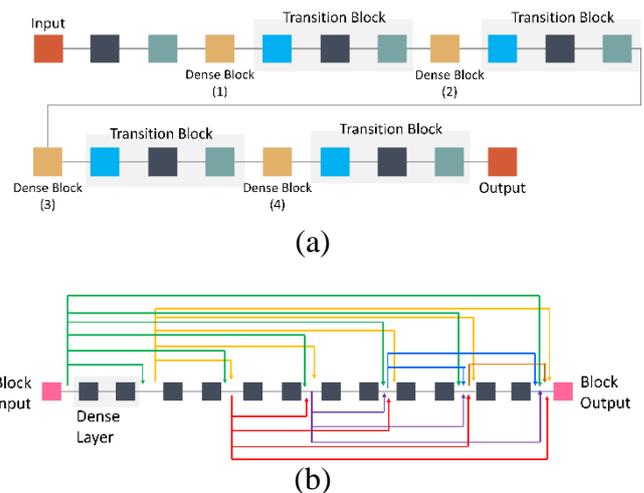


Fig. 8. DenseNet121 architecture; (a) overview configuration of the architecture, and (b) structure of dense block. Block's color represents a layer type: input/output layers (red), convolutional layers (navy), average pooling layers (green), batch normalization layers (sky blue), dense blocks (yellow), and input/output of dense blocks (pink). Adapted from H. Amin, et al. [40].

DenseNet121 is a CNN architecture with a feed-forward connection [40]. This connection style aims to augment the passage of information [41]. The configuration of DenseNet121 consists of an input layer, an average pooling layer, 4 dense blocks, 3 transition blocks, and an output layer with a softmax activation function as illustrated in Fig. 8(a) [40]. The dense block contains a different number of dense layers: 6 for the first block, 12 for the second, 24 for the third, and 16 for the last block [40]. 2 convolutional layers with different filter

sizes are included in each of dense layers as shown in Fig. 8(a) [40]. The transition block is connected to each dense block, except for the last dense block which is followed by the output layer [40]. Each transition block contains the stacking of 3 layers in the order of batch normalization layer, convolutional layer with ReLU activation function, and average pooling layer [40, 41].

3.2.2. Layers' function in CNN architectures

Apart from the input and the output layers, the layers that are mainly involved in the CNN architectures include the convolutional layer, pooling layer, and fully connected layer.

The convolutional layer is the highly prominent layer in the network responsible for feature extraction with the designated filter size [42, 43]. The nodes in this layer recognize the details of input images, and then each feature map is generated and stacked together [44]. In the generation of feature maps, the order of data collection is arranged based on the location of information, starting from the superficial details to insight details [44]. ReLU is also included in this layer as the non-linear activation function because the improved model performance is enabled without the variation of the layer's dimensions [42, 45, 46]. Moreover, training a large CNN network with this activation function is more convenient than other types of function [46]. The output of each convolutional layer, which will be the input of a subsequent layer, is based on the specification of filter including number and size [45].

The pooling layer is located after the convolutional layer for the purpose of down-sampling the feature maps to prevent output variation due to either small input or trivial dislocation of features in the image, such as translation, rotation, and scaling [42-44]. Additionally, this layer is capable of preventing the overfitting, resulting in the training of extracted features that mostly contain the relevant details of training data [44]. The output of the pooling layer relies on the type of pooling and the designated pool size. There are 2 types of pooling layers utilized in this study, which are named based on the principle of pooling: max pooling and average pooling. Each output of max pooling is the maximum value, while the individual output of average pooling comes from the average value [42].

In CNN layer configurations, the fully connected layer is typically positioned at the end [43]. Its function is to ensure that every node is connected to each other [42]. The fully connected layer containing the same number of units as the number of classes of dataset, will be defined as the output layer and the class scores will be estimated in this layer [46]. In this study, softmax was utilized as the activation function in the output layer.

3.2.3. Hardware specification

The models were compiled on Jupyter Notebook version 7.0.8. The computational hardware was ASUS

Vivobook S 14 OLED (K5404VA) with the computational specifications of 64-bit Windows 11 operating system, 16.0 GB RAM, and Intel Processor core i9-13900H (2.60 GHz).

3.3. The Study of the Effect of the Image Quality on Model Performance

To study how the quality of the dataset can affect the model's performance, we generated five case studies as follows:

Case 1: Train and test with high-quality images

Case 2: Train with high-quality images and test with standard-quality images

Case 3: Train and test with standard-quality images

Case 4: Train with standard images and test with high-quality images

Case 5: Train and test by combining high and standard-quality images

The dataset used in this research consists of high-quality images sourced from the DIBaS database, whereas standard-quality images were obtained from our laboratory. The workflow of the case study and the performance evaluation process is illustrated in Fig. 9. The evaluation metrics utilized to measure the performance of the models in this study are common formulas employed in the field of computer vision. These performance evaluation metrics include accuracy, precision, F-1 score, sensitivity, confusion matrix, Receiver Operating Characteristic (ROC) curve, and Area Under Curve (AUC) score [47, 48].

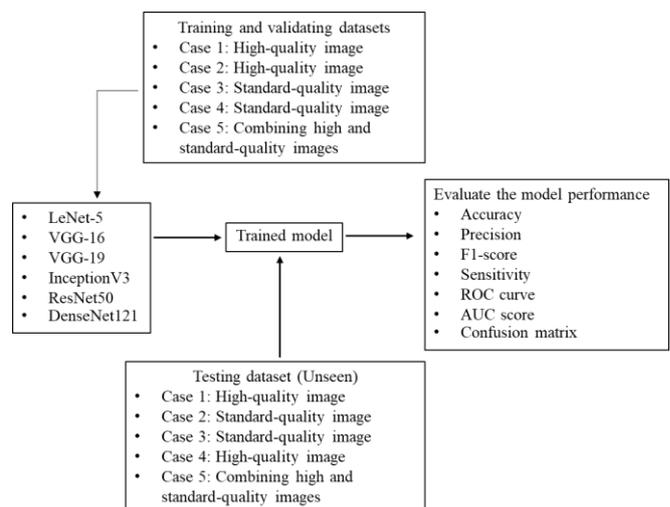


Fig. 9. Workflow diagram of study cases and evaluation.

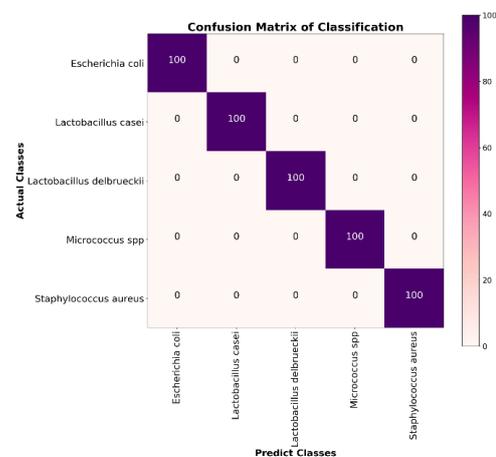
4. Results

To prove the effect of image quality on the model performance, we constructed 6 different CNN models: LeNet-5, VGG-16, VGG-19, InceptionV3, ResNet50, and DenseNet121 to classify 5 bacteria species proving 5

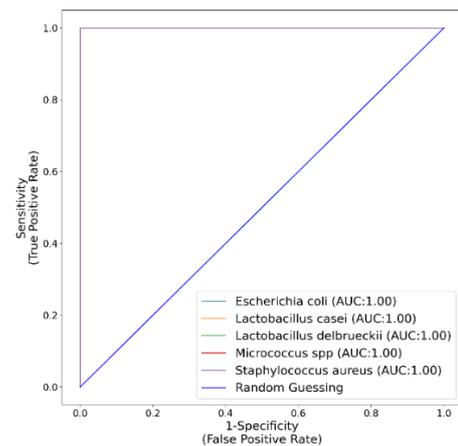
cases of different image quality of training and testing datasets. The results shown in Table 1 compare the evaluation results of the 5 cases under the different 6 CNN-trained models. Testing the models with the unseen dataset corresponds to the case study and performing with included and not included the stratified 5-folds cross-validation. Comparing the results under the same case study, the model performance relied on the model's architecture. The highest testing reaches 100% accuracy, precision, F-1 score, sensitivity, and AUC score in the case of training and testing with high-quality images (case 1) of VGG-16 and DenseNet121 under no cross-validation, also InceptionV3 under cross-validation. The evaluating parameters showed over 98% of LeNet-5, VGG-19, and ResNet50. Whereas training and testing with the different image quality datasets of cases 2 and 4, the models' performance was too small to be less than almost 50%, except for the no-cross-validated DenseNet121 model in case 4. Following case 1 result is case 5 of combining both high and standard image quality for training and testing. The results showed more than 95% of all evaluating parameters were performed under cross-validation of LeNet-5, InceptionV3, and ResNet50.

Additionally, Table 2 compares the AUC of no cross-validated models and cross-validated models, which was calculated from the prediction results of the models with the unseen dataset. The best performance obtained in case 1 represents the case study in which the models were trained and tested with high-quality images. The models achieve comparable outstanding performance of AUC score of 1 including non-cross-validated models of VGG-16 and DenseNet121, and cross-validated model of InceptionV3 for all classes. These are followed by non-cross-validated models of InceptionV3 and VGG-19 respectively. The confusion matrix and the ROC curve of non-cross-validated VGG-16 and DenseNet121 models were evaluated with the unseen dataset of high-quality images as illustrated in Fig. 10. The results of both trained models were the same because they achieved 100% predicted correctly. The outcomes of InceptionV3 model under cross-validation are depicted in Fig. 11. The evaluation results of InceptionV3 with non-cross-validated model for case 1 showed that the ROC curve demonstrated an AUC of 1 for all classes, except for *S. aureus*, which had an AUC of 0.99 (Table 2). The confusion matrix revealed that out of the unseen dataset containing a total of 500 images, the model accurately predicted 499 images. It correctly classified all images from different classes except for an image of *S. aureus* that was misclassified. The AUC represents the overall performance of a model in distinguishing between different classes. The results showed that the range of AUC with a higher value indicates better discrimination ability. An AUC of 1 suggests that the model has achieved perfect classification. The ROC curve visualizes the trade-off between the true positive rate (sensitivity) and the false positive rate (1 - specificity) at various classification thresholds. It provides a graphical representation of the model's performance across different thresholds.

Therefore, in this case, the high AUC values and the shape of the ROC curve suggest that the model exhibited excellent performance in classifying the different bacteria classes. For the evaluation results of VGG-19 in the non-cross-validated model with the unseen dataset of high-quality images, two trivial differences were found when compared to the aforementioned InceptionV3's results, even though they had a similar ROC curve and AUC score. In fact, this VGG-19 model was capable of accurately predicting 498 out of 500 high-quality images from the unseen dataset. According to Table 2, the model achieves a perfect AUC score from all classes, except for *L. casei* (AUC=0.99). Hence, the VGG-19 model misclassified 2 images of *L. casei*, while flawlessly classified bacteria images from other classes.

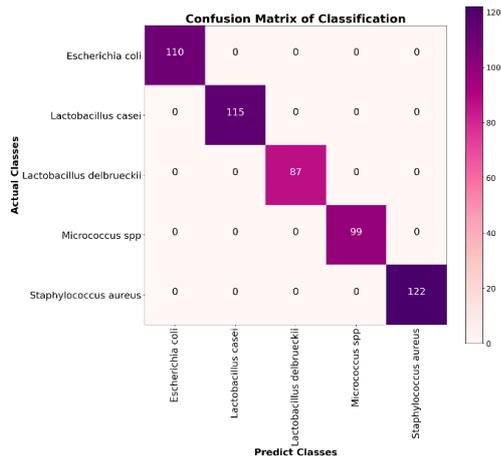


(a)

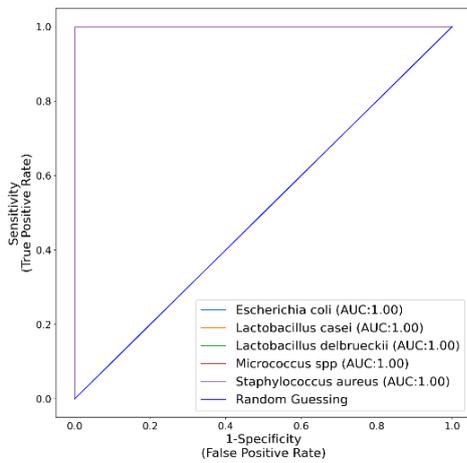


(b)

Fig. 10. The evaluation results of non-cross-validated VGG-16 and DenseNet121 models trained and tested with high-quality images (case 1): (a) confusion matrix and (b) ROC curve.



(a)



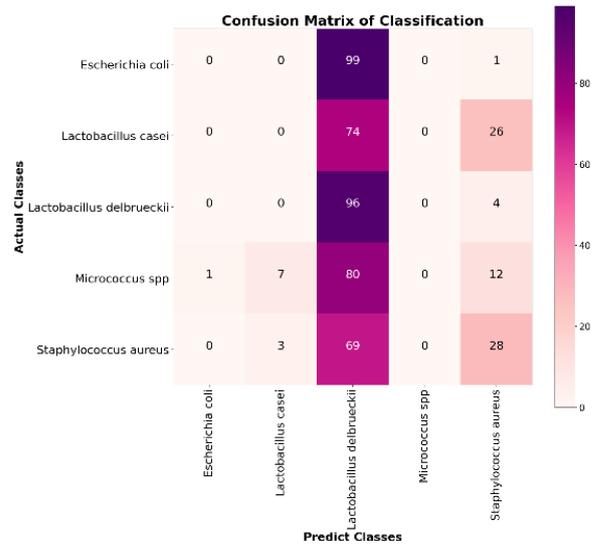
(b)

Fig. 11. The evaluation results of cross-validated InceptionV3 model trained and tested with high-quality images (case 1): (a) confusion matrix and (b) ROC curve.

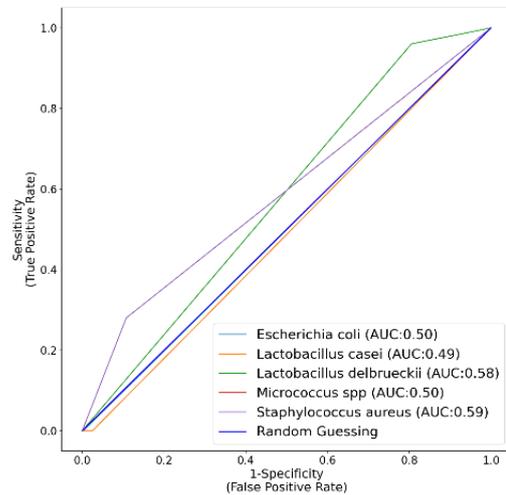
Although the evaluation results of models built in LeNet-5, VGG-16, VGG-19, and InceptionV3 with the unseen dataset of high-quality images of case 1 indicated excellent differentiating ability as the AUC of all classes approached 1 regardless of implementing cross-validation, both ResNet50 and DenseNet121 models under cross-validation obtained opposite outcomes. Even though the overall AUC of the DenseNet121 model and the other four models are alike, the AUC of *S. aureus* in this model was 0.71. For the ResNet50 model, not only the overall AUC but also the individual AUC of *L. delbrueckii*, *Micrococcus spp*, and *S. aureus* fell below 0.90. Among these classes, poor efficiency was observed on *S. aureus* with the AUC of 0.60.

Case 2 represents the case study in which the models were trained with high-quality images and tested with standard-quality images. The cross-validated ResNet50 model showed the best performance, followed by non-cross-validated models of LeNet-5 and DenseNet121 as shown in Table 1. However, the performance of the models tested by the unseen dataset was less than 0.5 for all metrics, and the AUC score of all classes was almost 0.5

as shown in Table 2, which corresponds to less performance. The confusion matrix and the ROC curve of ResNet50 model, evaluated with the unseen dataset of standard-quality images, are illustrated in Fig. 12. Despite being evaluated as the best-trained model, the ResNet50 model was mostly incorrectly predicted to be one of the two runners-up. Only a few classes achieved a score of this parameter greater than 0.5 but not more than 0.65 of *L. delbrueckii* from LeNet-5. Also, performing the cross-validation did not improve the performance of these models.



(a)



(b)

Fig. 12. The evaluation results of cross-validated ResNet50 model trained with high-quality image dataset and tested with the unseen dataset of standard-quality images (case 2): (a) confusion matrix and (b) ROC curve.

Table 1. The evaluation results of no-cross-validated models and cross-validated models built in LeNet-5, VGG-16, VGG-19, InceptionV3, ResNet50, and DenseNet121 tested by unseen dataset.

Case	CNN Architecture	No-Cross-Validation					Cross-Validation				
		Accuracy	Precision	F1-score	Sensitivity	AUC	Accuracy	Precision	F1-score	Sensitivity	AUC
Case 1	LeNet-5	0.9920	0.9921	0.9920	0.9920	0.9950	0.9944	0.9944	0.9944	0.9944	0.9965
	VGG-16	1.0000	1.0000	1.0000	1.0000	1.0000	0.9418	0.9530	0.9399	0.9418	0.9613
	VGG-19	0.9960	0.9961	0.9960	0.9960	0.9975	0.9867	0.9872	0.9868	0.9869	0.9914
	InceptionV3	0.9980	0.9980	0.9980	0.9980	0.9988	1.0000	1.0000	1.0000	1.0000	1.0000
	ResNet50	0.9840	0.9849	0.9840	0.9840	0.9900	0.6698	0.8356	0.6649	0.6698	0.8049
Case 2	DenseNet121	1.0000	1.0000	1.0000	1.0000	1.0000	0.8518	0.8979	0.8389	0.8518	0.9139
	LeNet-5	0.2400	0.1303	0.1595	0.2400	0.5250	0.2180	0.0989	0.1273	0.2180	0.5113
	VGG-16	0.2080	0.1367	0.1644	0.2080	0.5050	0.1980	0.0732	0.0767	0.1980	0.4987
	VGG-19	0.2140	0.0825	0.1166	0.2140	0.5088	0.2020	0.0822	0.0883	0.2020	0.5013
	InceptionV3	0.1560	0.0988	0.1193	0.1560	0.4725	0.1960	0.0647	0.0726	0.1960	0.4975
Case 3	ResNet50	0.2400	0.1011	0.1401	0.2400	0.5250	0.2480	0.1248	0.1396	0.2480	0.5300
	DenseNet121	0.2400	0.1198	0.1360	0.2400	0.5250	0.1960	0.3264	0.1316	0.1960	0.4975
	LeNet-5	0.7440	0.7514	0.7442	0.7440	0.8400	0.9218	0.9227	0.9219	0.9218	0.9511
	VGG-16	0.8040	0.7992	0.7795	0.8040	0.8775	0.9855	0.9859	0.9855	0.9855	0.9909
	VGG-19	0.7800	0.7890	0.7794	0.7800	0.8625	0.5545	0.7724	0.5395	0.5545	0.7216
Case 4	InceptionV3	0.8300	0.8447	0.8236	0.8300	0.8938	0.9345	0.9430	0.9336	0.9345	0.9591
	ResNet50	0.8300	0.8356	0.8287	0.8300	0.8938	0.2127	0.2691	0.1174	0.2127	0.5080
	DenseNet121	0.8320	0.8369	0.8292	0.8320	0.8950	0.3727	0.5484	0.3303	0.3727	0.6080
	LeNet-5	0.2540	0.3840	0.1721	0.2540	0.5338	0.2080	0.1085	0.0902	0.2080	0.5050
	VGG-16	0.3820	0.4171	0.2587	0.3820	0.6138	0.3320	0.4301	0.2439	0.3320	0.5825
Case 5	VGG-19	0.4000	0.1783	0.2415	0.4000	0.6250	0.3240	0.2122	0.2216	0.3240	0.5775
	InceptionV3	0.2260	0.2445	0.1189	0.2260	0.5163	0.2080	0.2411	0.0835	0.2080	0.5050
	ResNet50	0.4000	0.1823	0.2461	0.4000	0.6250	0.3260	0.3132	0.2111	0.3260	0.5788
	DenseNet121	0.7780	0.6850	0.7144	0.7780	0.8613	0.4200	0.4956	0.3307	0.4200	0.6375
	LeNet-5	0.8260	0.8605	0.8243	0.8260	0.8913	0.9777	0.9778	0.9778	0.9777	0.9863
Case 5	VGG-16	0.9240	0.9363	0.9248	0.9240	0.9525	0.8998	0.9302	0.8949	0.8998	0.9384
	VGG-19	0.9380	0.9488	0.9385	0.9380	0.9613	0.9475	0.9548	0.9480	0.9475	0.9675
	InceptionV3	0.9680	0.9689	0.9679	0.9680	0.9800	0.9889	0.9891	0.9889	0.9889	0.9932
	ResNet50	0.8600	0.8857	0.8428	0.8600	0.9125	0.9523	0.9565	0.9516	0.9523	0.9709
	DenseNet121	0.9360	0.9418	0.9355	0.9360	0.9600	0.6518	0.7920	0.6316	0.6518	0.7835

Case 1: Train and test with high-quality image

Case 3: Train and test with standard-quality images

Case 5: Train and test by combining high and standard-quality images

Case 2: Train with high-quality images and test with standard-quality images

Case 4: Train with standard images and test with high-quality images

Table 2. The area under curve (AUC) of no-cross-validated models and cross-validated models built in LeNet-5, VGG-16, VGG-19, InceptionV3, ResNet50, and DenseNet121 tested by unseen dataset.

Case	CNN Architecture	No Cross-Validation					Cross-Validation				
		<i>E. coli</i>	<i>I. casvi</i>	<i>I. delbrueckii</i>	<i>M. spb</i>	<i>S. aureus</i>	<i>E. coli</i>	<i>I. casvi</i>	<i>I. delbrueckii</i>	<i>M. spb</i>	<i>S. aureus</i>
Case 1	LeNet-5	1.00	1.00	1.00	0.99	0.98	1.00	1.00	1.00	0.99	0.99
	VGG-16	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.85	0.96	
	VGG-19	1.00	0.99	1.00	1.00	1.00	1.00	0.99	0.97	1.00	
	InceptionV3	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	
	ResNet50	1.00	0.97	0.99	0.99	0.99	0.93	0.83	0.74	0.60	
Case 2	DenseNet121	1.00	1.00	1.00	1.00	1.00	0.98	0.95	0.93	0.71	
	LeNet-5	0.50	0.49	0.65	0.50	0.48	0.50	0.62	0.50	0.45	
	VGG-16	0.27	0.50	0.62	0.50	0.63	0.46	0.50	0.50	0.53	
	VGG-19	0.50	0.50	0.56	0.50	0.48	0.49	0.50	0.50	0.51	
	InceptionV3	0.50	0.50	0.52	0.41	0.44	0.50	0.49	0.50	0.49	
Case 3	ResNet50	0.50	0.50	0.52	0.50	0.61	0.50	0.58	0.50	0.59	
	DenseNet121	0.50	0.49	0.55	0.50	0.59	0.50	0.49	0.52	0.57	
	LeNet-5	0.80	0.79	0.85	0.98	0.77	0.94	0.94	0.99	0.95	
	VGG-16	0.88	0.63	0.95	1.00	0.93	0.98	1.00	1.00	0.99	
	VGG-19	0.85	0.70	0.89	1.00	0.87	0.78	0.61	0.91	0.64	
Case 4	InceptionV3	0.84	0.70	0.98	1.00	0.94	0.88	0.98	1.00	0.99	
	ResNet50	0.85	0.76	0.93	1.00	0.92	0.53	0.50	0.54	0.50	
	DenseNet121	0.86	0.73	0.96	1.00	0.93	0.54	0.68	0.74	0.49	
	LeNet-5	0.56	0.57	0.47	0.63	0.44	0.50	0.50	0.59	0.42	
	VGG-16	0.93	0.51	0.50	0.64	0.49	0.81	0.51	0.69	0.41	
Case 5	VGG-19	0.50	0.91	0.50	0.72	0.50	0.64	0.50	0.79	0.47	
	InceptionV3	0.50	0.46	0.56	0.56	0.49	0.50	0.52	0.52	0.50	
	ResNet50	0.50	0.91	0.50	0.74	0.47	0.69	0.50	0.54	0.50	
	DenseNet121	0.99	1.00	0.95	0.88	0.49	0.51	0.48	0.74	0.63	
	LeNet-5	0.91	0.77	0.96	1.00	0.81	0.98	0.99	1.00	0.99	
Case 5	VGG-16	0.89	0.96	0.98	1.00	0.94	0.77	1.00	0.99	0.99	
	VGG-19	0.89	0.96	1.00	1.00	0.96	0.96	0.93	1.00	0.99	
	InceptionV3	0.98	0.98	0.99	1.00	0.95	0.98	1.00	1.00	0.99	
	ResNet50	0.95	0.94	0.98	0.99	0.70	0.97	0.99	1.00	0.91	
	DenseNet121	0.97	0.99	1.00	0.96	0.88	0.84	0.74	0.91	0.83	

Case 1: Train and test with high-quality image

Case 3: Train and test with standard-quality images

Case 5: Train and test by combining high and standard-quality images

Case 2: Train with high-quality images and test with standard-quality images

Case 4: Train with standard images and test with high-quality images

Case 3 represents the case study in which the models were trained and tested with standard-quality images. Under cross-validation, VGG16 model performed the best performance, followed by InceptionV3 and LeNet-5 respectively (Table 1). The confusion matrix and the ROC curve, evaluated with the unseen dataset of standard-quality images, were illustrated in Fig. 13 for VGG-16 cross-validation. In these top 3 models, the AUC score of all classes was approached to 1, except for *Escherichia coli* in the InceptionV3 model (AUC=0.88) as shown in Table 2. Also, based on Table 2, it can be inferred that the potential to enhance the overall outcomes by applying cross-validation to the models built in these architectures is plausible.

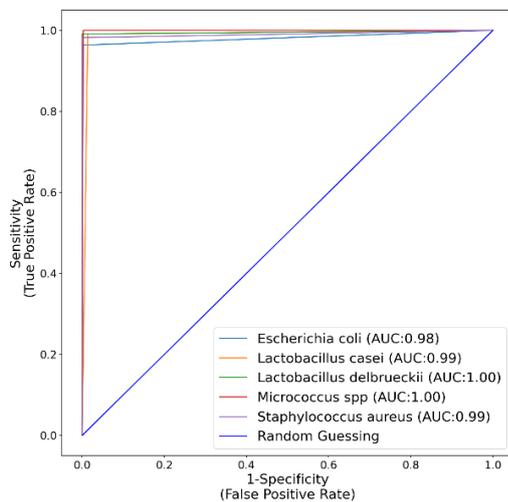
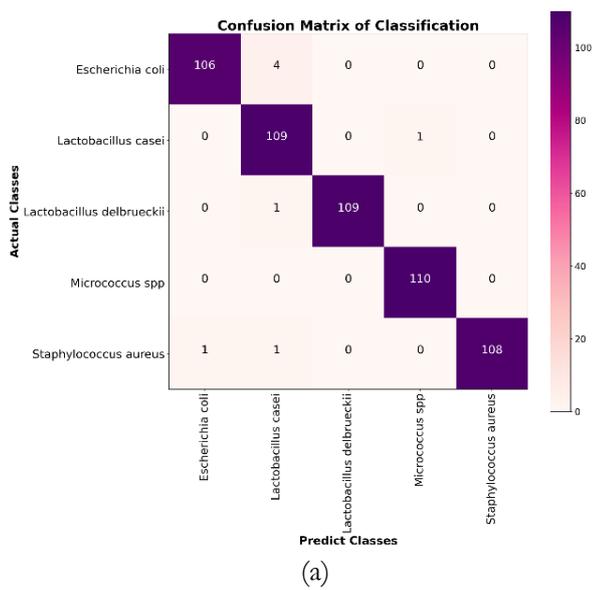


Fig. 13. The evaluation results of VGG-16 cross-validated model trained and tested with the standard-quality images (case 3): (a) confusion matrix and (b) ROC curve.

Case 4 represents the case study in which the models were trained with standard-quality images and tested with high-quality images. Among all models in this case,

DenseNet121 model without cross-validation outperformed with an accuracy of 77.8% and AUC score of 86.13% (Table 1). Compared to other models built in the same datasets in other cases, the accuracy was higher than the one in the same architecture in case 2 but lower than those that trained and tested with the dataset generated from the same resource of case 1 and case 3 under no cross-validation. The confusion matrix and the ROC curve are illustrated in Fig. 14 for DenseNet121. Although the accuracy of DenseNet121 model was the greatest, the AUC score of *S. aureus* was less than 0.5 while the AUC score of other classes was greater than 0.85 (Table 2). Moreover, the cross-validation did not significantly improve the performance of the model.

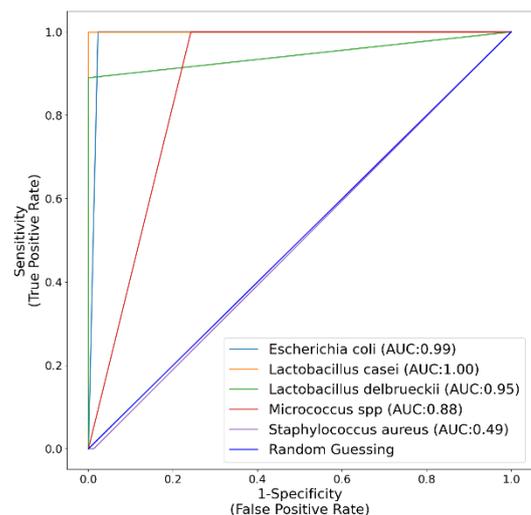
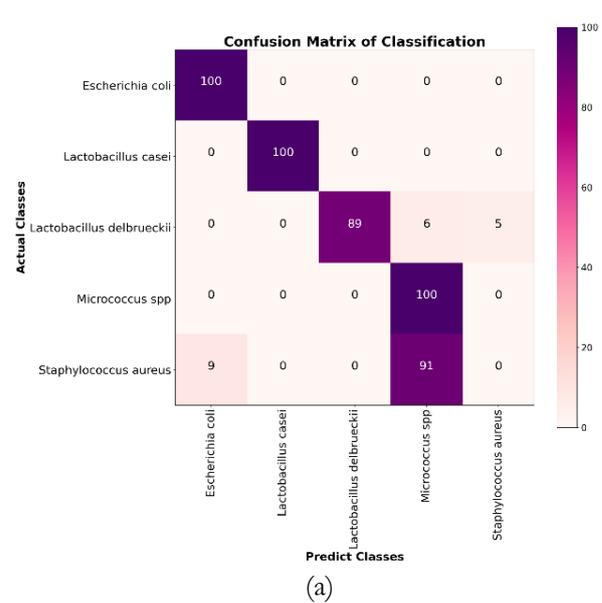


Fig. 14. The evaluation results of non-cross-validated DenseNet121 model trained with standard-quality images and tested with high-quality unseen images (case 4): (a) confusion matrix and (b) ROC curve.

Case 5 represents the case study in which the models were trained and tested with combined high-quality and

standard-quality images. Apart from DenseNet121 model, the parameters for evaluated performance of models built in other architectures with and without cross-validation were higher than 80%. Furthermore, apart from VGG-16 and DenseNet121 models, performing cross-validation could improve the performance turned to more than 90% as shown in Table 1. The AUC score approached 1 for almost all classes as shown in Table 2. Only the AUC score of *L. casei* for the cross-validated DenseNet121 model and *S. aureus* for non-cross-validated ResNet50 model was not more than 0.70, indicating the poor classification efficiency. The confusion matrix, AUC score, and ROC curve of the best cross-validated InceptionV3 model are illustrated in Fig. 15 indicating that *L. delbrueckii* and *Micrococcus spp* images were classified into correct categories, which aligns with the ideal AUC score for both classes.

5. Discussion

The results of each study case demonstrated that the quality of the training image dataset and the unseen validating image dataset significantly impact the performance of the model. The quality of microscopic images is influenced by characteristics of microorganisms, such as their size and gram-stained color. The larger size of the cell was visible in its shape. Even belonging to the same gram-staining group such as *L. casei*, *L. delbrueckii*, *Micrococcus spp*, and *S. aureus*, the purple color-stained different shade, and intensity. The stained colors were distinguishable in high-quality images, whereas the stained colors were less distinct in standard-quality images. Moreover, the higher specifications of the microscope and the equipped camera allowed for the collection of higher-resolution images with less noise, offering clearer detail in physical characteristics such as gram-stained color, and cell morphology. Therefore, it is unsurprising that the high-quality microscopic image dataset led to better model performance, as shown in the case 1 study results. The results observed in cases 2 and 4, where the training and validation datasets had different qualities, indicate lower classification performance. In contrast, cases 1, 3, and 5, where the training and validation datasets came from the same source with consistent image quality, showed higher classification performance. The results of all cases suggested that using the same training and unseen testing datasets generally yields better performance compared to using different quality datasets for training and testing. There are several reasons why matching the training and validating datasets in terms of quality leads to better performance:

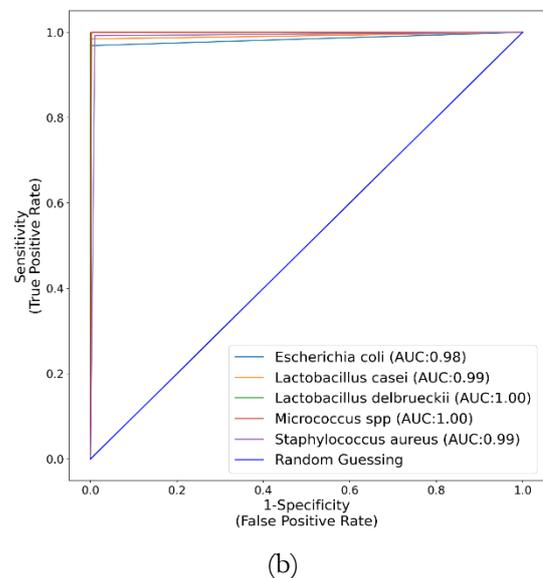
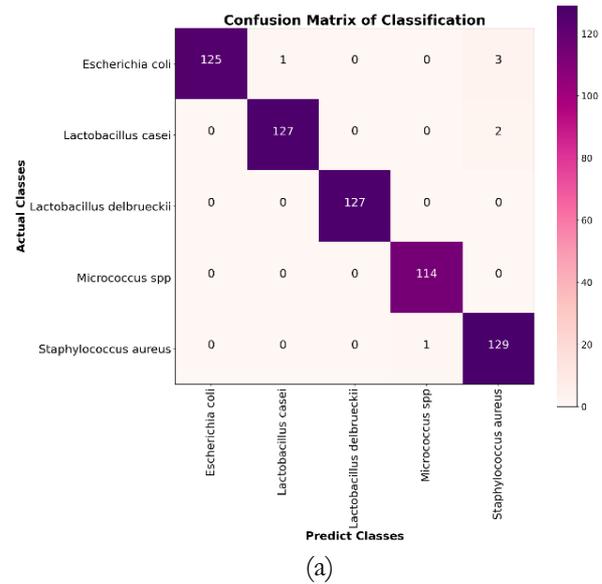


Fig. 15. The evaluation results of cross-validated InceptionV3 model trained and tested with combined high-quality and standard-quality images (case 5): (a) confusion matrix and (b) ROC curve.

Consistency in image characteristics: when the training and validating datasets have the same quality, the model becomes familiar with and learns to recognize the specific characteristics and patterns of detail within those images. This consistency of quality images allows the model to generalize better and accurately classify unseen images during validation.

Distribution shift: using datasets with different qualities introduces a distribution shift between the training and validating datasets. This shift can cause the model to struggle in adapting to the differences and fail to generalize well to new images. The model can avoid this distribution shift and maintain reliable performance by maintaining consistency in image quality.

Robust feature extraction: models trained on datasets with consistent image quality can develop robust feature

extraction capabilities. When the different sources of the validating images match the training images, the model can effectively extract relevant features from the unseen images, leading to better classification performance.

Constructing the model with and without cross-validation did not significantly affect the model's performance, especially case 1 and case 5. These results confirm that the models did not contain the overfitting of the training and validation of the dataset. It is noteworthy that whether the model included or excluded cross-validation, all trained models exposed similar results of evaluation performance for each case as the high-quality image dataset used in training and testing sets produced the best performance models. However, the cross-validated model was more effective when the models were trained and tested with high-quality combined images from the DIBaS open-source dataset and private standard-quality images. This result indicated that the randomly separated training, validating, and unseen testing dataset of cross-validation 5 folds improves the training efficiency of the model by balancing the proportion of both images' quality. According to the model performance which was trained and tested by the unseen dataset from different sources, the performance of the model trained by the high-quality image dataset was better than those using the standard-quality image dataset. Therefore, the applicable model for microscopic bacteria image classification is suggested to be trained by high image resolution and included with the images prepared by the clinical user to achieve effective performance.

Considering the pre-trained CNN models used to study the image quality affects the model performances, several states of the art for CNN architectures have been investigated for bacterial classification as discussed by Y. Wu, et al. [22]. The assessment of performance involved the consideration of accuracy, precision, recall, and F1-score of 13 trained models including AlexNet, GoogleNet, Inception V3, VGG-16, and 19. ResNet18, 34, 50, and 152, and DensNet121, 161, 169, and 201 have been reported and showed over 95% of performance metrics from all CNN algorithms. Among these CNN algorithms, the pre-trained DensNets121 and 161 displayed superior performances with 99.08% accuracy, 99.06% precision, 99% recall, and 98.99% F1-score obtained from the DenseNet121 model emerging as the most favorable architecture concerning the DIBaS dataset [22]. These results were similar to those of our re-trained DenseNet121 model, which was implemented using the high-quality image dataset from DIBaS, where all performance metrics reached 100%. The Inception V3 and DensNet121 exhibited outstanding performances, especially in most cases. These results reveal that the complexity of the architecture impacts the model's ability to learn and recognize key distinguishable features. Especially case 4 of DensNet121 was trained with standard images and tested with high-quality images showing a surprising performance compared to other models that the accuracy reached 77.8% and AUC score of 86.13% (Table 1) and for AUC of *E. coli*, *L. casei*, and

L. delbrueckii were 0.99, 1 and 0.95, respectively under no cross-validation (Table 2). However, these evaluated performances were unacceptable in case 4. Considering case 5 of DenseNet121 under cross-validation, the performance decreased compared to when no cross-validation was used. This observation could be explained by the reasons for the transfer learning. Transfer learning algorithms, such as DenseNets, are typically pretrained on large-scale datasets with distinct statistical properties, which can create a domain gap between the source and target datasets [22]. Consequently, the re-trained DenseNet121 model with the randomly separated training, validating, and unseen testing dataset may struggle to effectively adapt to the unique characteristics of the DIBaS and our datasets. The importance of image resolution when selecting deep learning models for complex tasks, such as bacterial classification. Since deeper networks require more detailed input data to effectively capture hierarchical features, high-resolution images are essential for achieving optimal results. We focused on analyzing feature maps in the convolutional layers of different deep learning architectures. While our preliminary research suggested that dropout affects models with more classes, we did not specifically observe its impact here, as the primary focus was on identifying the optimal architecture. We observed that high-resolution images enhanced performance in deeper models like InceptionV3, ResNet50, and DenseNet121. These models, with their deeper architectures, benefit from high-resolution data, enabling better feature extraction and improved classification accuracy. In contrast, lower-resolution images hindered feature extraction, particularly in the deeper layers, leading to reduced performance.

This research addresses the challenges of building high-performance CNN models using datasets from two different sources, both with and without cross-validation. It highlights the crucial role that input image quality plays in training CNN models for optimal performance. By demonstrating how an unseen test set can mirror real-world bacterial classification applications, the study underscores the importance of training models with consistent image sources. This approach is essential for developing robust, reliable applications in the field. Nevertheless, this study is limited to five categories of bacteria, including one gram-negative bacillus and four gram-positive bacilli and cocci. Expanding the study to include more diverse datasets would improve the generalizability of the findings. Additionally, the state-of-the-art in deep learning does not limit only CNN architecture, several techniques of machine learning and deep learning can challenge this purpose. Furthermore, the findings from this study reveal a strategy for developing various applications in the automatic detection of microorganisms, such as the rapid detection and identification of pathogens. Using clinical bacterial datasets with high image resolution is a more effective approach to achieving highly accurate prediction performance for clinical samples. Additionally, this study focused solely on feature maps without exploring

additional interpretability techniques. Future research could integrate advanced techniques such as saliency maps or activation maximization to provide deeper insights into how models make classification decisions, further enhancing the understanding of model behaviour and improving interpretability.

6. Conclusion

Our findings demonstrate that image quality significantly influences the performance of deep learning models in bacterial classification tasks. This study introduces a theoretical perspective on the critical role of dataset quality, proposing that high-quality datasets not only improve model accuracy but also enhance the feature extraction process, enabling better differentiation between bacterial species. This contribution provides a foundation for developing systematic frameworks to evaluate and optimize dataset quality in microbiological imaging, with potential applications in advancing automated diagnostic tools for bacterial classification and other medical imaging tasks.

References

- [1] S. Kotwal, P. Rani, T. Arif, J. Manhas, S. Sharma, “Automated bacterial classifications using machine learning based computational techniques: Architectures challenges and open research issues,” *Arch. Comput. Methods Eng.*, vol. 29, pp. 2469–2490, 2022.
- [2] N. Mohamad, N. A. Jusoh, Z. Zaw, and S. L. Win, “Bacteria identification from microscopic morphology: A survey,” in *International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI)*, vol. 3, no. 2, 2014.
- [3] L. Chen, W. Kai, and X. Ning, “A survey for the applications of content-based microscopic image analysis in microorganism classification domains,” *Artif. Intell. Rev.*, vol. 51, pp. 577–646, 2019.
- [4] B. Zieliński, A. Plichta, K. Misztal, P. Spurek, M. Brzywczy-Włoch, D. Ochońska, “Deep learning approach to bacterial colony classification,” *Plos One*, vol. 12, no. 9, p. e0184554, 2017.
- [5] S. Visitsattapong, M. Bunkum, C. Pintavirooj, and M. P. Paing, “A deep learning model for bacterial classification using big transfer (BiT),” *IEEE Access*, vol. 12, pp. 15609–15621, 2024.
- [6] M. F. Wahid, T. Ahmed, and M. A. Habib, “Classification of microscopic images of bacteria using deep convolutional neural network,” in *10th International Conference on Electrical and Computer Engineering (ICECE)*, Dhaka, Bangladesh, 2018, pp. 217–220.
- [7] D. T. Mai and K. Ishibashi, “Small-scale depthwise separable convolutional neural networks for bacteria classification,” *Electronics*, vol. 10, no. 23, p. 3005, 2021.
- [8] P. Rani, S. Kotwal, J. Manhas, and S. Sharma, “Machine learning and deep learning based computational approaches in automatic microorganisms image recognition: Methodologies, challenges, and developments,” *Arch. Comput. Methods Eng.*, vol. 29, pp. 1801–1837, 2022.
- [9] C. Spahn, E. Gómez-de-Mariscal, R. F. Laine P. M. Pereira, L. V. Chamier, M. Conduit, M. G. Pinho, G. Jacquemet, S. Holden, M. Heilemann, and R. Henriques, “DeepBacs for multi-task bacterial image analysis using open-source deep learning approaches,” *Commun. Biol.*, vol. 5, 2022.
- [10] M. G. F. Costa, C. F. F. Filho, A. Kimura, P. C. Levy, C. M. Xavier, and L. B. Fujimoto, “A sputum smear microscopy image database for automatic bacilli detection in conventional microscopy,” in *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Chicago, IL, USA, 2014, pp. 2841–2844.
- [11] N. Tripathi, and A. Sapra, *Gram Staining*. StatPearls Publishing, 2023.
- [12] M. Koziarski, and B. Cyganek, “Impact of low resolution on image recognition with deep neural networks: An experimental study,” *Int. J. Appl. Math. Comput. Sci.*, vol. 24, pp. 735–744, 2018.
- [13] S. G. Traoré, G. Fokou, A. S. Wognin, et al., “Assessment of handwashing impact on detection of SARS-CoV-2, *Staphylococcus aureus*, *Escherichia coli* on hands in rural and urban settings of Côte d’Ivoire during COVID-19 pandemic,” *BMC Public Health*, vol. 24, 2024.
- [14] X. Wen, F. Chen, Y. Lin, H. Zhu, F. Yuan, D. Kuang, Z. Jia, Z. Yuan, “Microbial indicators and their use for monitoring drinking water quality—A review,” *Sustainability*, vol. 12, no. 6, pp. 2249, 2020.
- [15] D. O. Matuka, B. Binta, H. A. Carman, T. Singh, “*Staphylococcus aureus* and *Escherichia coli* levels on the hands of theatre staff in three hospitals in Johannesburg, South Africa, before and after handwashing,” *SAMJ: South African Medical Journal*, vol. 108, 2018.
- [16] M. Gobbetti and F. Minervini, “*Lactobacillus* | *Lactobacillus casei*,” in *Encyclopedia of Food Microbiology*, 2nd ed. Academic Press, 2014, pp. 432–438.
- [17] S. A. Ibrahim, “Lactic acid bacteria: *Lactobacillus* spp.: other species,” in *Reference Module in Food Science*. Elsevier, 2016.
- [18] G. Betts, “Other spoilage bacteria” in *Food Spoilage Microorganisms*. Woodhead Publishing, 2006, ch. 23, pp. 668–693.
- [19] S. Baron, “*Staphylococcus*,” in *Medical Microbiology*, 4th ed. Galveston, TX: University of Texas Medical Branch at Galveston, 1996, ch. 12.
- [20] K. C. Keim and A. R. Horswill, “*Staphylococcus aureus*,” *Trends in Microbiology*, vol. 31, pp. 1300–1301, 2023.
- [21] A. Gnanamani, P. Hariharan, and M. Paul-Satyaseela, “*Staphylococcus aureus*: Overview of Bacteriology, Clinical Diseases, Epidemiology, Antibiotic Resistance

- and Therapeutic Approach,” in *Frontiers in Staphylococcus aureus*, *InTech*, 2017.
- [22] Y. Wu and S. A. Gadsden, “Machine learning algorithms in microbial classification: A comparative analysis,” *Front. Artif. Intell.*, vol. 6, 2023.
- [23] N. M. Khalifa, M. H. Taha, A. E. Hassanien, and A. A. Hemedan, “Deep bacteria: Robust deep learning data augmentation design for limited bacterial colony dataset,” *Int. J. Reason. based Intell. Syst.*, vol. 11, pp. 256-264, 2019.
- [24] C. M. Liang, C. C. Lai, S. H. Wang, and Y. H. Lin, “Environmental microorganism classification using optimized deep learning model,” *Environ. Sci. Pollut. Res. Int.*, vol. 28, pp. 31920-31932, 2021.
- [25] X. Wang, Y. Shi, S. Guo, X. Qu, F. Xie, Z. Duan, Y. Hu, H. Fu, X. Shi, T. Quan, K. Wang, and L. Xie, “Clinical bacterial dataset for deep learning in microbiological rapid on-site evaluation,” *Sci Data*, vol. 11, 2024.
- [26] E. Hallström, V. Kandavalli, P. Ranefall, J. Elf, and C. Wählby, “Label-free deep learning-based species classification of bacteria imaged by phase-contrast microscopy,” *PLoS Comput Biol*, vol. 19, no. 11, 2023.
- [27] R. O. Panicker, K. S. Kalmady, J. Rajan, and M. K. Sabu. “Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods,” *Biocybern. Biomed. Eng.*, vol. 38, pp. 691-699. 2018.
- [28] S. Trattner, H. Greenspan, G. Tepper, S. Abboud, “Automatic identification of bacterial types using statistical imaging methods,” *IEEE Trans Med Imaging*, vol. 23, pp. 807-820, 2004.
- [29] B. Park, T. Shin, B. Wang, B. McDonogh and A. Fong, “Classification between live and dead foodborne bacteria with hyperspectral microscope imagery and machine learning,” *Journal of Microbiological Methods*, vol. 209, pp. 106739, 2023.
- [30] M. A. Morid, A. B. and G. D. Fiol, “A scoping review of transfer learning research on medical image analysis using ImageNet,” *Computers in Biology and Medicine*, vol. 128, pp. 104115, 2021.
- [31] Y. Xie, X. Chen, H. Zhan, P. Shivakumara, B. Yin, C. Liu and Y. Lu, “Weakly supervised scene text generation for low-resource languages,” *Expert Systems with Applications*, vol. 237, pp. 121622, 2024.
- [32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in *Proceedings of the IEEE*, vol. 86, pp. 2278-2324, 1998.
- [33] J. Zhang, X. Yu, X. Lei, and C. Wu, “A novel deep LeNet-5 convolutional neural network model for image recognition,” *Computer Science and Information System*, vol. 19, no. 3, pp. 1463-1480, 2022.
- [34] Md Zahangir Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, “The history began from AlexNet: A comprehensive survey on deep learning approaches,” arXiv:1803.01164 [cs.CV], 2018.
- [35] A. H. Md. Linkon, Md. M. Labib, T. Hasan, M. Hossain, and M.-E-Jannat, “Deep learning in prostate cancer diagnosis and Gleason grading in histopathology images: An extensive study,” *Informatics in Medicine Unlocked*, vol. 24, 2021.
- [36] O. Iparraguirre-Villanueva, V. Guevara-Ponce, O. R. P. F. Sierra-Liñan, J. Zapata-Paulini, and M. Cabanillas-Carbonell, “Convolutional neural networks with transfer learning for pneumonia detection,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 9, pp. 544-551, 2022.
- [37] Z.-K. Chai, L. Mao, H. Chen, T.-G. Sun, X.-M. Shen, J. Liu, and Z.-J. Sun, “Improved diagnostic accuracy of ameloblastoma and odontogenic keratocyst on cone-beam CT by artificial intelligence,” *Frontiers in Oncology*, vol. 11, 2021.
- [38] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770-778, 2016.
- [39] S. Mascarenhas and M. Agarwal, “A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification,” in *2021 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON)*, Bengaluru, India, pp. 96-99, 2021.
- [40] H. Amin, A. Darwish, A. E. Hassanien, and M. Soliman, “End-to-end deep learning model for corn leaf disease classification,” *IEEE Access*, vol. 10, pp. 31103-31115, 2022.
- [41] J. Potsangbam and S. S. Devi, “Classification of breast cancer histopathological images using transfer learning with DenseNet121,” *Procedia Computer Science*, vol. 235, pp. 1990-1997, 2024.
- [42] S. Tammina, “Transfer learning using VGG-16 with deep convolutional neural network for classifying images,” *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, no. 10, pp. 143-150, 2019.
- [43] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaria, M. A. Fadhel, M. Al-Amidle, and L. Farhan, “Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, 2021.
- [44] Y. H. Liu, “Feature extraction and image recognition with convolutional neural networks,” *Journal of Physics: Conference Series*, vol. 1087, no. 6, pp. 062032, 2018.
- [45] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, “Feature extraction using convolutional neural networks (CNN) and deep learning,” in *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, pp.2319-2323, 2018.
- [46] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, “Convolutional neural network (CNN) for image detection and recognition,” in *2018 First International*

Conference on Secure Cyber Computing and Communication (ICSCCC), Jalandhar, India, pp. 278-282, 2018.

- [47] P. Zhao, C. Li, M. M. Rahaman, H. Xu, P. Ma, H. Yang, H. Sun, T. Jiang, N. Xu, and M. Grzegorzec, "EMDS-6: Environmental microorganism image dataset sixth version for image denoising, segmentation, feature extraction, classification, and

detection method evaluation," *Front. Microbiol.*, vol. 13, 2022.

- [48] J. I. E. Hoffman, *Probability, Bayes Theorem, Medical Diagnostic Evaluation, and Screening. Basic Biostatistics for Medical and Biomedical Practitioners*, 2nd ed. Academic Press, pp. 311-338, 2019.



Treesukon Treebupachatsakul received D.Eng. degree in integrated bioscience and technology and M.Eng. degree in Bioengineering from the Nagaoka University of Technology, Japan, in 2011 and 2015, respectively. She is currently an Assistant Professor at the Biomedical Engineering Department, School of Engineering, King Mongkut's Institute of Technology Ladkrabang, Thailand. Her research interests are the application of biochemistry, bioengineering, and biotechnology, especially cell culture, proteins, and enzymatic reactions applied with biomedical engineering, computer software, and communication.



Wanwalee Chomkwah was born in Bangkok, Thailand in 1999. She received B.Eng. degree in biomedical engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2022.

In 2024, she received M.Sc. degree in biomedical engineering with cell and tissue engineering at University of Strathclyde, Glasgow, UK.



Tananan Tanpatanan was born in Chonburi, Thailand in 2000. She received B.Eng. degree in Biomedical engineering from King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2022.

Since 2022, her research interest includes deep learning for the classification of multiclass bacteria and different types of oral white lesions.



Suvit Poomrittigul received a D.Eng. degree in Information Science and Control Engineering from Nagaoka University of Technology, Japan in 2014. He received his M.Eng. in Computer Engineering from Chulalongkorn University and B.Eng. in Telecommunication Engineering from King Mongkut's Institute Technology Ladkrabang (KMITL), Thailand in 2009 and 2005, respectively. From, 2009 to 2022.

He worked as a lecturer and an Assistant Professor at the Department of Software Engineering and Information System, Pathumwan Institute of Technology, Thailand. He is currently an Assistant Professor at the School of Information Technology, KMITL, Thailand. His current research interests include Artificial Intelligence, Image Processing, Intelligent Transportation System, Computer Simulation, and Information Systems.