ILKOM Jurnal Ilmiah Vol. 14, No. 1, April 2022, pp. 80-90
Accredited 2nd by RISTEKBRIN No. 200/M/KPT/2020; E-ISSN 2548-7779 | P-ISSN 2087-1716

Research Article Open Access (CC–BY-SA)

Multi Classification of Bacterial Microscopic Images using Inception V3

Ingrid Nurtanio[a,1,*]; Anugrayani Bustamin[a,2]; Christoforus Yohannes[a,3]; Alif Tri Handoyo[a,4]
[a] *Universitas Hasanuddin, Jl Perintis Kemerdekaan KM 10, 90245, Makassar, Indonesia*
[1] *ingrid@unhas.ac.id;* [2] *anugrayani@unhas.ac.id;* [3] *christoforus@unhas.ac.id;* [4] *alif.trihandoyo15@gmail.com*
** Corresponding author*

Article history: Received March 1, 2022; Revised April 4, 2022; Accepted April 4, 2022; Available online April 30, 2022

Abstract

Microorganisms such as bacteria are the main cause of various infectious diseases such as cholera, botulism, gonorrhea, Lyme disease, sore throat, tuberculosis and so on. Therefore, the identification and classification of bacteria are very important in the medical field to help doctors diagnose diseases suffered by patients. However, manual identification and classification of bacteria takes a long time and a professional individual. With the help of artificial intelligence, we can effectively and efficiently classify bacteria and save a lot of time and human labor. In this study, a system was designed to classify bacteria from microscopic image samples. This system employed deep learning with the transfer learning method. Inception V3 architecture was modified and retained using 108 image samples labeled with five types of bacteria, namely *Acinetobacter baumanii, Escherichia coli, Neisseria gonorrhoeae, Propionibacterium acnes* and *Veionella*. The data were then divided into training and validation using the k-fold cross validation method. Furthermore, the features that have been extracted by the model were trained with the configuration of *minibatchsize* 5, *maxepoch* 5, *initiallearnrate* 0.0001, and validation frequency 3. The model was tested with data validation by conducting ten experiments and obtaining an average accuracy value of 94.42%.

Keywords: Bacterial Classification; Deep Learning; Inception V3; Transfer Learning; Image Processing

Introduction

According to the Food and Agriculture Organization (FAO), the number of victims who die from bacterial infections reaches up to 700,000 people every year. Recognition of bacterial genera and species is necessary because knowledge of the biology of microorganisms is significant in medical field, veterinary medicine, biochemistry, the food industry, and agriculture. Although most microorganisms have a positive impact on various areas of life, they can cause many diseases, including infectious diseases [1] [2].

Biologists identify and classify different types of bacteria with different biochemicals and forms. They used different bacterial attributes for classification, for example, the shape of bacterial cells (spiral, cylindrical and spherical). The size and structure of the colonies formed by bacteria are examined to distinguish bacterial species. Cells of several types of bacteria have different sizes and structures depending on environmental conditions. Several species of bacteria have very similar shapes. Although each bacterial species has its own characteristics, the biochemical reactions carried out by bacteria and their metabolic activities together help classify the species. However, the classification of bacterial species is not an easy task even for an experienced specialist [3] [4]. Song, et al. have proposed analysis for Bacterial Vaginosis (BV) images from a microscope. They proposed an automatic method to diagnose BV with several stages, i.e., segmentation, splitting, and classification of overlapping bacteria image cases. Following that, the implementation of the Nugent score criterion was used in their experiment demonstration and achieved high accuracy with computing efficiency [5].

In general, microbiological image analysis using traditional laboratory methods has bacteria recognition errors and requires different experience and long processing time. Therefore, the automatic classification technique of bacterial images is more valuable than traditional visual observations for biologists because of its accurate classification, low cost, and fast diagnosis. Previous research related to bacterial classification was carried out in 2018 by Basma, et al. classifying ten types of bacteria using the Bag of words feature extraction and Support Vector Machine (SVM) method with 97% accuracy [6].

Identifying a bacterium is a laborious process. Several studies used algorithms to classify with promising alternatives [7]. Kukula, et al. used Convolutional Neural Network based approach paired with Raman spectroscopy to detect and recognize the bacteria class rapidly. The result shows 86% of accuracy with identification speed close to real-time [8]. In 2019, Treesukon, et al. conducted a study classifying two types of bacteria using a python-based deep learning method and the LeNET architecture resulting in more than 75 percent accuracy [9]. In 2018 Lei Huang used the CNN method with the AlexNET architecture to classify 18 categories of bacteria with an accuracy of 73% [10].

This study has classified bacteria using the CNN deep learning method with the InceptionV3 architecture. InceptionV3 architecture was used because, in previous studies, it was widely used to classify with reasonably high accuracy results, such as the classification of patients with COVID-19 disease [11] and the classification of patients with lung disease [12].

Method

A convolutional neural network (CNN) is an architecture that can recognize predictive information of an object such as images, text, sound snippets, and so on. CNN is a development of multilayer perceptron (MLP), which is designed to process data in images. CNN is included in the type of deep neural network because of its high network depth and it is widely applied to image data. Research on CNN was first conducted by Hubel and Wiesel on the visual cortex in the sense of sight of cats. CNN architecture consists of several layers, namely convolution [13].

Inception V3 uses less computing power by modifying the previous Inception architecture. This idea was proposed by an article entitled Rethinking the Inception Architecture for Computer Vision, published in 2015. Christian Szegedy co-authored the statement,Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens. Compared to VGGNet, Inception Networks (GoogLeNet / Inception v1) is proven to be more computationally efficient, both in terms of the number of parameters generated by the network and the cost of memory and other resources. If any changes are to be made to the Inception Network, they must be carefully made to ensure that the computational advantage is not lost. Thus, the adaptation of the Inception network to different cases turned out to be a problem due to the uncertainty of the efficiency of the new network. In the Inception v3 model, several techniques for optimizing the network have been suggested to reduce the constraints to make model adaptation easier. These techniques include factorized convolutions, regularization, dimension reduction, and parallelized computations [14]. The architecture of Inception V3 can be seen in Fig. 1.

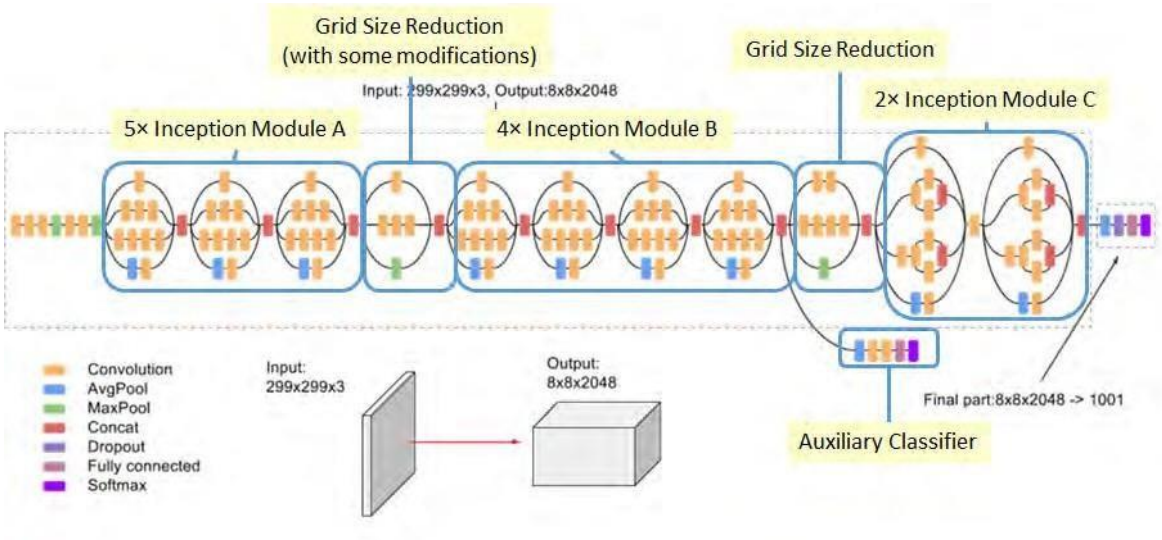


Figure 1. Inception V3 Architecture [15]

A system for classifying bacteria is divided into three essential stages. Before describing the important stages in the development of this system, this study also describes the data sources used. The data were taken in 20 images for each class of bacteria with a resolution of 2048 x 1532. There are 108 images which will later be divided into training and validation data. Bacterial image data was obtained through the DIBaS source (http://misztal.edu.pl/software/databases/dibas/). This data is an image of bacteria that has gone through the gram staining process and was taken with an Olympus CX31 microscope equipped with an SC30 camera. The data consists

of 108 images of five bacteria classes, namely *Acinetobacter baumanii, Escherichia coli, Neisseria gonorrhoeae, Propionibacterium acnes*, and *Veionella*. Examples of training data for each class of bacteria can be seen in Fig. 2.

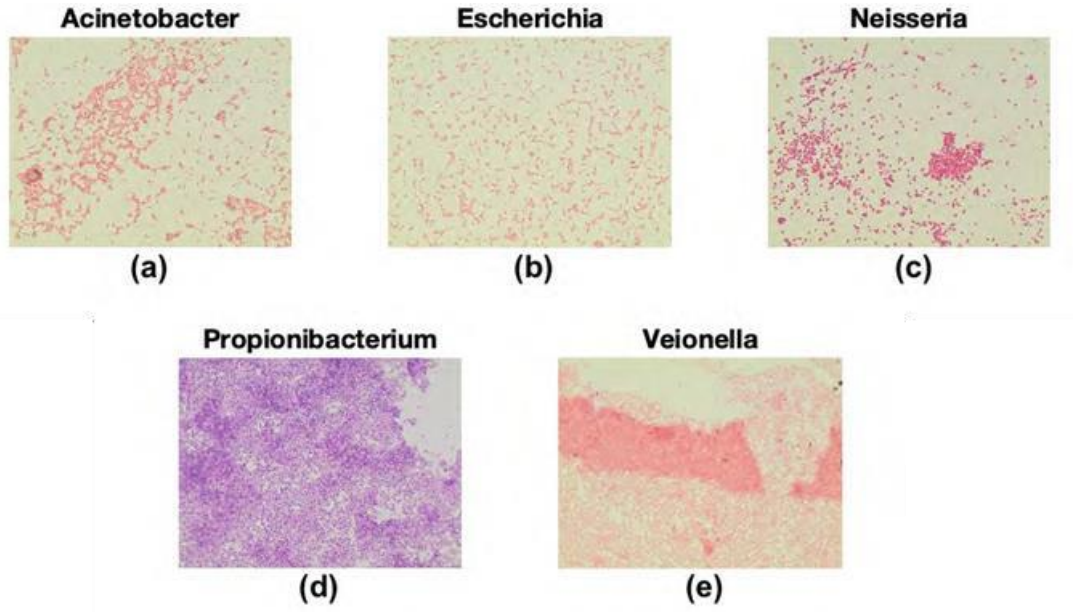


Figure 2. Sample data of (a) *Acinetobacter baumanii*,(b) *Escherichia coli*, (c) *Neisseria gonorrhoeae*, (d) *Propionibacterium acnes*, (e) *Veionella.*

The characteristics or features of each class of bacteria used in the training data are as follows:

- Acinetobacter. It has rod-shaped or is included in the bacilli category with a coccobacillus shape (ellipse shape) and has a gram-negative color (reddish-pink).
- Escherichia. This shape is included in the bacilli category with the shape of a bacillus (stem) and has a gram-negative color (reddish-pink).
- Neisseria. This shape belongs to the cocci category with a diplococci shape (a pair of circles) and has a gram-negative color (reddish-pink).
- Propionibacterium. This shape is included in the bacilli category with the shape of a bacillus (rod) and has a gram-positive color (violet).
- Veionella belongs to the cocci category with a diplococci shape (paired circle) and has a gram-negative color (reddish-pink).

After the dataset is ready for use, there were three stages that the system would pass to arrive at the expected classification results, namely the training, validation, and testing processes, as shown in Fig. 3. According to Fig. 3, training and validation process used a preprocessing stage. Before the InceptionV3 pretrained network was used, the last two layers of the architecture, namely the predictions layer and the *ClassificationLayer_predictions* layer, would be adjusted first to classify the dataset used which consisted of five types of bacterial classes. Before the layer modification process was carried out, the two layers to be changed were first searched using the *findLayersToReplace* function. After finding the layer, the predictions layer was modified to match the number of dataset classes used, the five types of bacterial classes. Furthermore, this *ClassificationLayer_predictions* layer still contained the class label of the pretrained network, so it would be replaced by the *new_classoutput* layer, which did not have a class label. The results of layer adjustments on the InceptionV3 architecture can be seen in Fig. 4.

It can be seen in Fig. 4 that the last two layers at the bottom have been adjusted to classify five types of bacteria and changed their names to *predictions_softmax* and *new_classoutput*. The image data was then go through a labelling process which was carried out to divide the data into five classes that would later be used in the training and validation process. The amount of data and its labels can be seen in Table 1. After that, the training and validation data splitting process was carried out using the k-fold cross-validation method with k=5. The results of this data division can be seen

in Fig. 5. K-fold cross-validation was carried out so that each image in the dataset has the opportunity to become training and validation data so that the best dataset to produce the highest accuracy could be obtained.

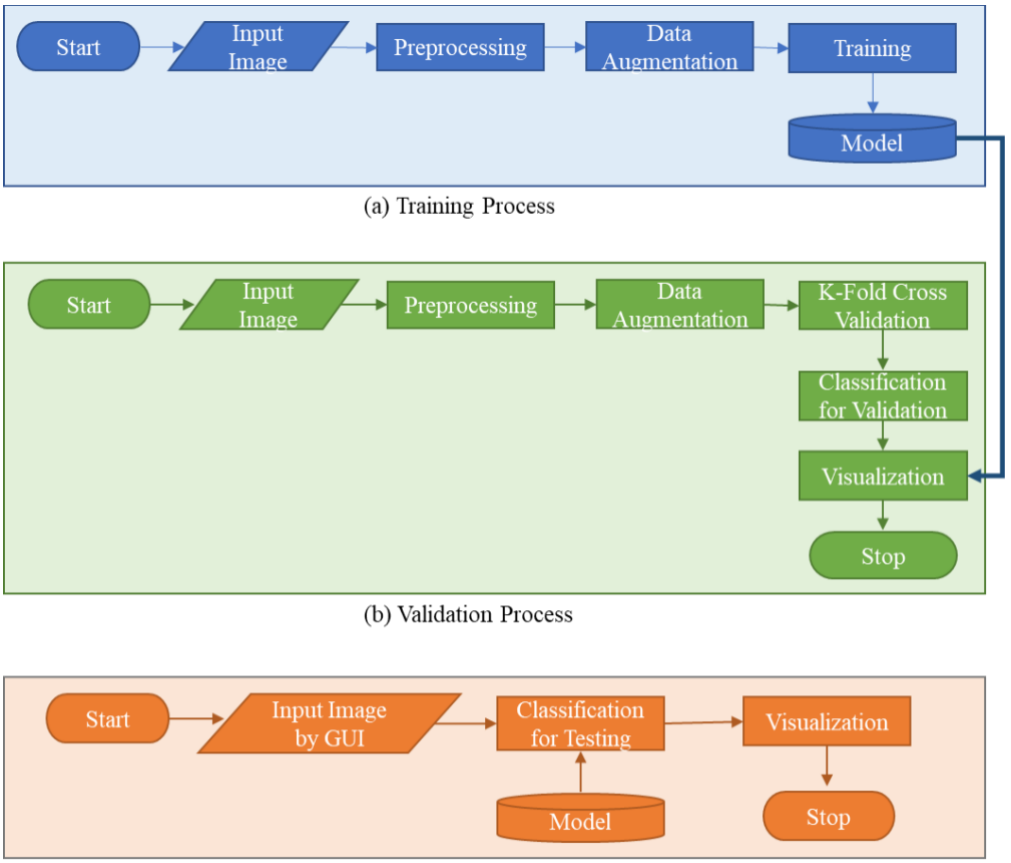


Figure 3. Flowchart of System Design using Inception V3 (a) Training Process, (b) Validation Process and (c) Testing Process

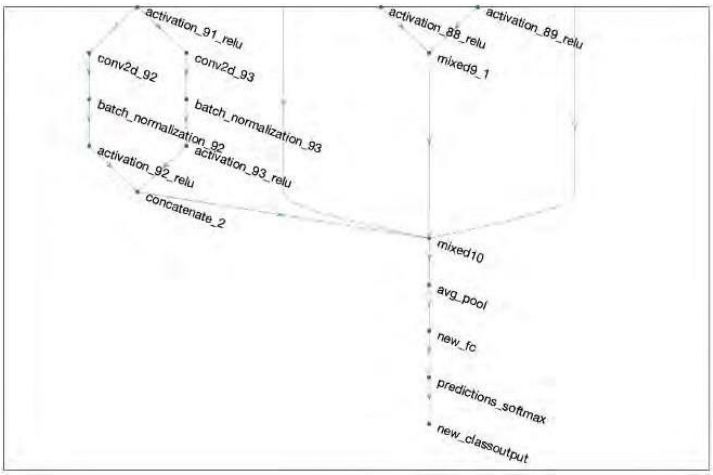


Figure 4. The illustration of Inception V3 Layer Adjustment

Table 1. The Quantity of Datasets based on Five Classes

Class	Label	Count
1	*Acinetobacter baumanii,*	20
2	*Escherichia coli,*	20
3	*Neisseria gonorrhoeae,*	20
4	*Propionibacterium acnes*	20
5	*Veionella*	20

```
i = 1
Numberoftraindata = 86
Numberofvalidationdata = 22
i = 2
Numberoftraindata = 87
Numberofvalidationdata = 21
i = 3
Numberoftraindata = 86
Numberofvalidationdata = 22
i = 4
Numberoftraindata = 86
Numberofvalidationdata = 22
i = 5
Numberoftraindata = 87
Numberofvalidationdata = 21
```

Figure 5. Distribution of training and validation data using k-fold cross validation

Data Augmentation provided more variety of data in the network by performing various random transformation processes such as *RandXReflection*, *RandXTranslation*, and *RandYTranslation* on each training data and transferring it to the network, so that later the network contains multiple dataset variances. As for the amount of data, data augmentation produced the same amount of data as in the process without data augmentation. Hence, this augmentation data provided more data variance. In the training process, each epoch iteration provided various data variances transformed randomly with *RandXReflection*, *RandXTranslation*, and *RandYTranslation*. In addition, a resizing process was also carried out to change the image size to 299x299 pixels. This was done because the InceptionV3 architecture *InputLayer* requires an image size of 299x299 pixels. The results of the data augmentation preview (eight images) can be seen in Fig. 6.

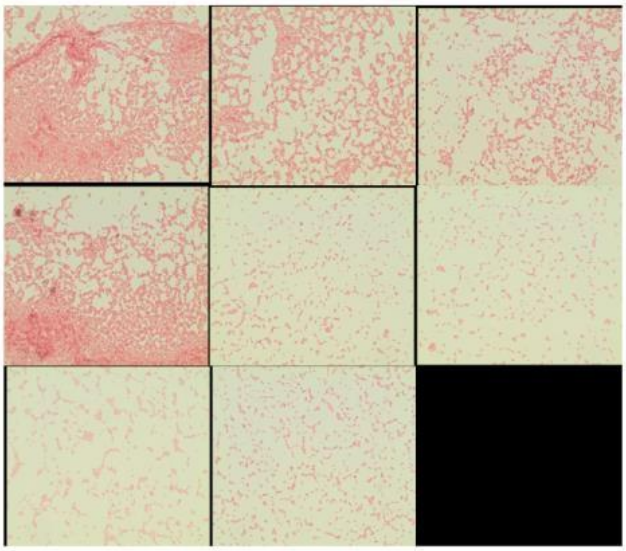


Figure 6. Augmentation results with different parameters

Inception modules are the core block of this transfer learning model. There were 11 inception modules, ranging from *mixed0* to *mixed10*. Two hundred fifty-six channel features with a resolution of 35x35 were generated by the inception module in the mixed0 layer. Furthermore, the convolution process further developed along with the increase in layers to extract more abstract features. This convolution layer produced 768 and 2048 channel-feature maps of 17x17 and 8x8. Finally, all feature extraction results were combined into a one-dimensional vector with an output of five neurons according to the number of data classes, namely five bacterial classes. The neuron with the highest probability would be the class label of the bacterial image used as test data.

A. Training Process

The training process was carried out using the transfer learning method with a *minibatchsize* configuration of 5, epoch 5, and a learning rate of 0.0001. A lot of training and validation data was used following the distribution of the

dataset through a k-fold cross-validation process with k=5 meaning that the process was repeated five times. The explanation of the parameters used in the configuration of the training process are as follows:

- *MiniBatchSize*

  This parameter is a measure of the mini-batch that is used in each iteration of the training process. The mini-batch is the part of the training process used to evaluate the gradient of the loss function and update the weights.

- *MaxEpoch*

  This parameter is the maximum number of epochs used in the training process. Iteration is the step taken by the gradient descent algorithm to minimize the loss function with batches. The epoch is the full path of the training algorithm across the training set.

- *InitialLearnRate*

  This parameter indicates the *initiallearnrate* used in the training process. Three kinds of solvers can be used, namely *sgdm*, *rmsprop* and *adam*. The solver used in this training process is the *sgdm* solver. The training process will take longer when the learning rate is too low. On the other hand, when the learning rate is too high, the training process will be faster, but the training results may give less than optimal or distorted results.

- *ValidationFrequency*

  This parameter indicates the number of iterations performed for the network validation process in each epoch.

- Model

  The output of the training process is a model that would later be used for the validation process using data validation that has been divided through the k-fold cross-validation process.

B. Validation Process

This validation process was carried out to see the loss value and model performance resulting from the classification in the training process. The classify function generated matrix scores of 21x5, which contained the accuracy of the five dataset classes. The highest accuracy class was used as a data label from validation data that has been classified and then stored in the *ypred* matrix with a size of 21x1.

When executed, the classify function classified the validation data in the *uugimdsvalidation* cell using a network net. This cell contained 21 data validation images. A label is a matrix of size N-of-1 where N is the number of validation data, and scores is a matrix of size N-of-C, where N is the number of validation data and C is the number of classes of the dataset.

The process of calculating the probability score for multiclass using the classify function in Equation 1 below.

$$P(c_r|x,\theta) = \frac{P(x,\theta|c_r)P(c_r)}{\sum_{j=1}^{k} P(x,\theta|c_j)\,P(c_j)} = \frac{\exp\,(a_r\,(x,\theta))}{\sum_{j=1}^{k} \exp\,(a_j\,(x,\theta))} \quad (1)$$

Where $0 \leq P(c_r|x,\theta) \leq 1$ dan $\sum_{j=1}^{k} P(c_j \mid x,\theta) = 1$

In addition, a_r is the conditional probability of the sample by class r and $P(c_r)$ is the probability of the previous class. The probability score calculation process is in the range of 0 – 1, and the class with the highest probability was used as the label of data validation.

C. Testing Process

- Input Image

  The data used for the testing process differs from the training and validation data dataset. For each session of the testing process, the maximum number that can be classified is one image of bacteria.

- Classification

The classification stages are similar with data validation process.

- Visualization

  The classification results will display the original image, the Grad-CAM visualization, and the prediction and accuracy results of the testing data input.

- GUI (Graphical User Interface)

  The GUI in this system plays a role in facilitating the user in classifying bacteria. The user inputted an image of the bacteria classified by clicking the "browse" button, then clicking the "prediction" button. The system then classified bacteria based on the network that has been trained, and provided output in the form of prediction results, percentage accuracy, and Grad-CAM visualization.

D. Metric Evaluation

In measuring the performance of the bacterial classification system, calculation of the prediction performance made by design on the input image data in the training process using the help of a confusion matrix should be conducted. The Confusion Matrix consists of four parts, namely:

- True Positive (TP). Represents the number of correct predictions from positive data.
- False Positive (FP). Represents the number of false predictions from positive data.
- True Negative (TN). Represents the number of correct predictions from negative data.
- False Negatives (FN). Represents the number of false predictions from negative data.

To calculate the system accuracy value, the following Equation 2 can be applied:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

Results and Discussion

This deep learning bacterial classification system employed a transfer learning-based model with the Inceptionv3 architecture. The data used were 108 images consisting of five classes of bacteria, namely Acinetobacter baumanii, Escherichia coli, Neisseria gonorrhoeae, *Propionibacterium acnes* and *Veionella*. This data was then divided into training and validation data using the k-fold cross-validation method with the number of folds k=5, producing 87 images for training data and 21 for data validation. Before training the data, the data augmentation process was carried out with the transformation of *RandXReflection*, *RandXTranslation*, and *RandYTranslation*. The training process would provide more variety of datasets into the network.

In this study, three testing scenarios were carried out, starting with the first experiment using the Inceptionv3 architecture with k-fold cross-validation. The second experiment applied the Inceptionv3 architecture without k-fold cross validation then divided the training validation data with a percentage of 70:30. The third experiment employed the inceptionv3 architecture with k-fold cross validation without augmentation. Finally, a comparison of the performance of the inceptionv3 architecture with seven other architectures, namely *googlenet*, *vgg16*, *vgg19*, *resnet101*, *inceptionresnetv2*, *squeezenet* and *xception* was made.

The training configuration in each experiment used the same parameters, which are *minibatchsize*, *maxepochs*, *initiallearnrate*, and *validationfrequency*. The values used for each of these parameters are *minibatchsize* 5, *maxepoch* 5, *initiallearnrate* 0.0001, and *validationfrequency* 3. Each experiment was trained five times to see the average performance and tested with data testing.

A. First Scenario

In the first experiment, the system was built using the Inceptionv3 architecture with k-fold cross-validation. The results of the five-time trial training can be seen in Table 2.

Table 2. *InceptionV3* with *K-fold cross validation*

Training	Time	Accuracy
1	7 minutes 8 seconds	95.45%
2	7 minutes 23 seconds	90.48%
3	7 minutes 39 seconds	95.45%
4	7 minutes 39 seconds	95.45%
5	7 minutes 29 seconds	95.29%
Average of Time Training		7 minutes 27 seconds
Average of Accuracy		94.42%

B. Second Scenario

Secondly, the system was built using the Inceptionv3 architecture without k-fold cross-validation. The training and testing data were divided by the percentage of 70:30. The results of the five-time trial training can be seen in Table 3.

Table 3. *InceptionV3* without *K-fold cross validation*

Training	Time	Accuracy
1	8 minutes 3 seconds	90.91%
2	7 minutes 54 seconds	93.94%
3	7 minutes 1 seconds	93.94%
4	7 minutes 42 seconds	84.85%
5	8 minutes 11 seconds	81.82%
Average of Time Training		7 minutes 49 seconds
Average of Accuracy		89.09%

C. Third Scenario

In the last experiment, the system was built using the Inceptionv3 architecture with k-fold cross validation without the augmentation process. The results of the five-time trial training can be seen in Table 4.

Table 4. InceptionV3 without augmentation

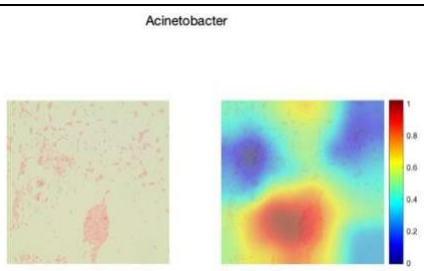
Training	Time	Accuracy
1	7 minutes 9 seconds	72.73%
2	7 minutes 4 seconds	85.71%
3	7 minutes 10 seconds	95.45%
4	7 minutes 9 seconds	90.91%
5	7 minutes 16 seconds	95.24%
Average of Time Training		7 minutes 9 seconds
Average of Accuracy		88.01%

Based on the results of the three above scenarios, the average accuracy in the first experiment was higher than in the second experiment. Due to the influence of k-fold cross-validation, it increased the system's accuracy. By applying k-fold cross-validation, every data in the dataset has the opportunity to become training and validation data so that the best dataset is obtained that produces the highest accuracy.

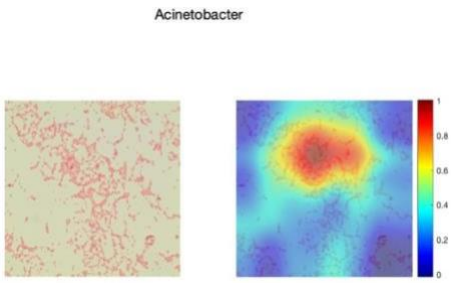
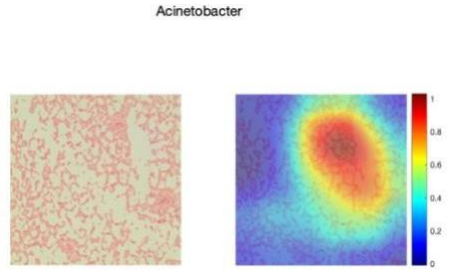
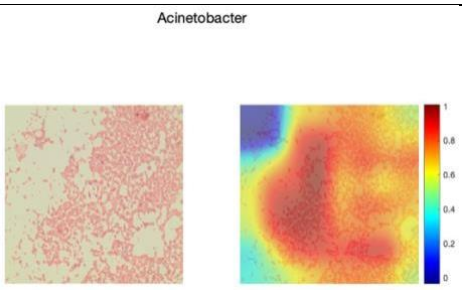
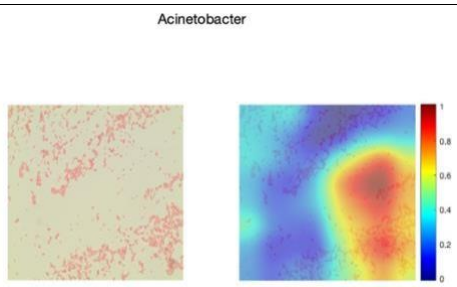
Finally, in the third experiment, in the absence of augmentation on the system, the accuracy of the system will decrease due to the lack of data variation in the training process so that the resulting network is not optimal. The system in these three experiments was made to classify four types of bacteria comprising of *Acinetobacter baumanii, Escherichia coli, Neisseria gonorrhoeae, Propionibacterium acnes*, and *Veionella*.

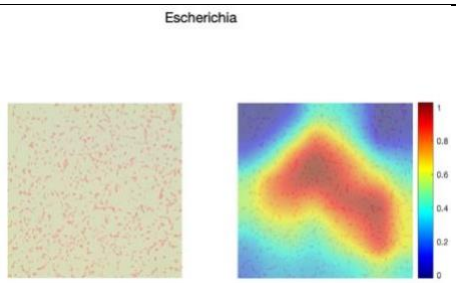
In the confusion matrix of the first experiment, there was one false positive or misclassification in the data validation classification of *Escherichia coli* bacteria, which then was classified as as *Acinetobacter baumanii* bacteria by the system. These two bacteria have similar characteristics of gram-negative colour as well as similar shape as the shape in the bacilli category. This wrong classification process can be seen with the Grad-Cam visualization in Table 5. As for the visualization of the results of the correct bacterial classification, it can be seen in Table 6.

Tabel 5. Classification Error Results

Data Validation	Visualization
8	

Tabel 6. The Correct Classification

Data Validation	Visualization
1	
2	
3	
4	

Data Validation	Visualization
5	

Conclusion

This study has conducted analysis on bacterial classification using deep learning. It can be concluded that of the three experiments that have been carried out, the bacterial classification system using the InceptionV3 architecture with k-fold cross-validation and augmentation provides the highest average accuracy. This system can classify five types of bacteria, namely *Acinetobacter baumanii, Escherichia coli, Neisseria gonorrhoeae, Propionibacterium acnes* and *Veionella*. The experimental results of the bacterial classification system applying the InceptionV3 architecture with k-fold cross-validation and augmentation using the *minibatchsize* 5, *maxepoch* 5, *initiallearnrate* 0.0001, and *validationfrequency* 3 training configurations result in an average validation accuracy value of 94.42% with an average training time of 7 minutes 27 seconds.

In the future, the system can be developed by adding the types of classified bacteria to classify more types of bacteria. In addition, the system that has been made can also be modified to classify other objects to overcome different problems.

References

[1] B. Zieliński, A. Plichta, K. Misztal, P. Spurek, M. Brzychczy-Włoch and D. Ochońska, "Deep learning approach to bacterial colony classification," PLoS One, vol. 12, no. 9, 2017.

[2] Z. V. , D. C. and M. P., "Algorithmic escherichia coli bacteria incidence evaluation," in Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges, Porto, Portugal, 2018.

[3] M. Talo, "An Automated Deep Learning Approach for Bacterial Image Classification," in International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES2019), Alanya, Turkey, Apr 2019.

[4] M. F. Wahid, M. J. Hasan and S. M. Alom, "Deep Convolutional Neural Network for Microscopic Bacteria Image Classification," in International Conference on Advance in Electrical Engineering, Dhaka, Bangladesh, September 2019.

[5] Y. Song, L. He, F. Zhou, S. Chen, D. Ni, B. Lei and T. Wang, "Segmentation, Splitting, and Classification of Overlapping Bacteria in Microscope Images for Automatic Bacterial Vaginosis Diagnosis," IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 4, pp. 1095-1104, 2017.

[6] M. Basma A and A. Heba M, "Automated classification of Bacterial Images extracted from Digital Microscope via Bag of Words Model," in 9th Cairo International Biomedical Engineering Conference (CIBEC), Cairo, 2018.

[7] e. a. Leite, "Exploration of multiclass and one-class learning methods for prediction of phage-bacteria interaction at strain level," in IEEE International Conference on Bioinformatics and Biomedicine, Madrid, Spain, 2018.

[8] K. Kukula et al., "Rapid Detection of Bacteria Using Raman Spectroscopy and Deep Learning," in IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), NV, USA, 2021.

[9] T. T and P. S. , "Bacteria Classification using Image Processing and Deep learning," in 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), United States, 2019.

[10] L. Huang and T. Wu, "Novel neural network application for bacterial colony classification," Theoretical biology & medical modelling, vol. 15, no. 1, pp. 1-16, 2018.

[11] M. G. Nakrani and U. B. Shinde, " Classification of COVID-19 from Chest Radiography Images Using Deep Convolutional Neural Network," Journal of Xidian University, vol. 14, no. 8, pp. 565-569, 2020.

[12] C. W. e. al, "Pulmonary image classification based on inception-v3 transfer learning model," IEEE Access, vol. 7, pp. 146533-146541, 2019.

[13] D. Hubel and T. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," Journal of Physiology, vol. 195, pp. 215-243, 1968.

[14] V. Kurama, "A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet," 2020. [Online]. Available: https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/. [Accessed 20 May 2021].

[15] S. Tsang, "Review: Inception-v3 — 1st Runner Up (Image Classification) in ILSVRC 2015," 2018. [Online]. Available: https://sh-tsang.medium.com/review-inception-v3- 1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c. [Accessed 13 November 2020].