

# **Classifying Various Bacteria Genera by Transfer Learning Model**

Mohammed Elgaed, Khaled Elgdamsi and Ali Ganoun\*

m.elgead@uot.edu.ly, k.elgdamsi@uot.edu.ly, a.ganoun@uot.edu.ly

Electrical and Electronic Engineering Department, Faculty of Engineering, University of Tripoli,  
Libya

\*Corresponding author email: a.ganoun@uot.edu.ly

## **ABSTRACT**

Bacteria Classification using computer-aided methods makes the identification and recognition processes more automatic and thus greatly reduces the time needed for classification. In this paper, we explored an approach to automating the process of classifying bacteria with the use of deep Convolutional Neural Network (CNN). CNNs is one of deep machine learning methods that mimics the connectivity pattern between visual cortex neurons. It can extract hierarchical image feature representations based on multi-layer processing. The 'transfer learning' approach was used to retrain a famous convolutional neural network model with a dataset of 152 images of 7 different bacteria species. The retrained model has been able to recognize and classify all 7 different species of bacteria with very high accuracy.

**Keywords:** bacteria classification, deep learning, convolution neural network, transfer learning, computer Vision

## **1 Introduction**

The identification of bacterial genera and species is vital, as the biological knowledge of microorganisms is extremely significant in medicine, veterinary science, biochemistry or food industry. Although most of the microorganisms have positive impact on various areas of life, they can be a reason of many diseases (including the infectious ones). Therefore, automating the process of recognition can find application in different sectors such as in medical prevention, diagnosis and treatment. The recognition of microbiological samples is preceded by the culturing process. This process includes dedicated equipment and chemical agents to be used for staining the samples. It also follows stringent culturing procedures and safety protocols. As a result, we obtain samples that are studied for the purpose of discovering characteristic features and classifying the particular genera and species of bacteria.

Classical laboratory methods for bacteria identification and recognition require an expert knowledge and experience [1]. The use of recent artificial intelligence methods such as Convolutional Neural Network (CNN) to automatically identify and classify bacteria is vital for improved recognition accuracy.

Classification is an active research area in pattern recognition which has been used in many applications that involves a decision mechanism that assigns a new observation to a set of classes, based on a training dataset. Traditional methods largely rely on manually crafting descriptors and characteristics. On the contrary, this paper proposes a Deep Learning (DL) approach that can automatically extract features from two-dimensional bacteria images.

The CNN models are an extension of deep learning of artificial networks. They consist of Multi-hidden Layer Perceptron's (MLPs) which involve multiple convolution, pooling, Rectified Linear Unit (ReLU), and fully-connected layers. Generally, the features maps of the previous layers are convolved with learnable weights on a convolutional layer and fed through the activation functions to form the output features maps. Each output map combines convolutions with multiple input features maps. For a specific output map, the input maps will be convolved with distinct kernels and the convolutional layer shares its weights in a same output map. Convolutional layers are usually combined with pooling layers to reduce dimensions and increase speed. Although the pooling layer is desirable as it reduces the computational time, it may cause loss of some information. Generally, the pooling layer produces sub-sampled versions of the input maps [2].

## **2 System Model**

Image-processing and pattern recognition techniques combined with various types of classifiers are often used as a useful tool for recognition of laboratory samples, occurring in the form of images. There are many methods for automatic recognition of bacteria species and strains such as statistical methods [3], artificial neural networks [4], or other machine learning classifiers [5]. Most of the mentioned methods are used to recognize very few species or strains (sometimes only one, e.g., tuberculosis). Moreover, in many cases, the algorithms based on the morphological features combined with some classification methods. Therefore, they are constrained.

This paper considers the AlexNet – a pre-trained Convolutional Neural Network (CNN) model – which trained on a subset of ImageNet database of around 1.2 million images of 1000 classes and tested on 150,000 images belonging to 1000 classes [6, 7]. The last three layers of the pre-trained AlexNet network are configured for 1000 classes and must be fine-tuned for the new classification problem. The fine-tuning done by extracting all layers except the last three from the pre-trained network. Then Transferring the layers to the new classification task by replacing the last three layers with a Fully Connected layer, a SoftMax layer, and a classification output layer. Figure 1 shows the AlexNet network architecture.

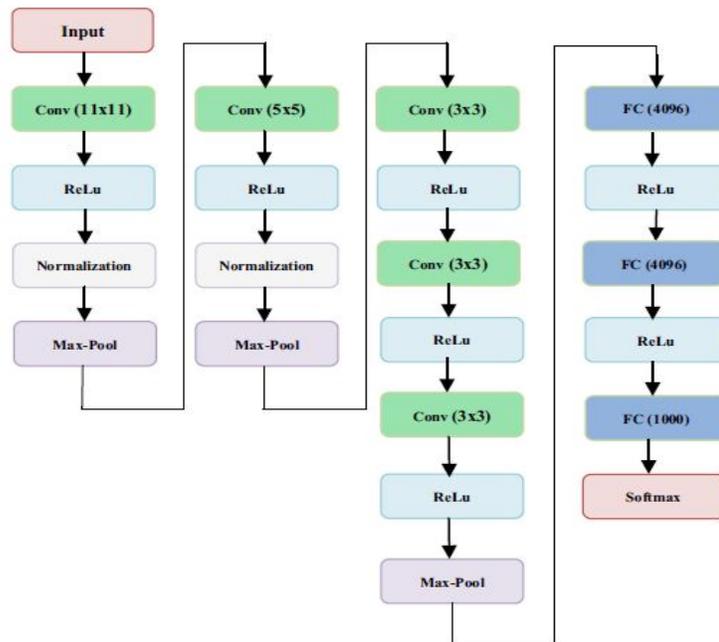


Figure 1: Structure of the AlexNet network architecture

## 2.1 Convolution & Pooling Layers

Convolution layer is the core building block of a convolutional neural network which uses convolution operation. The first layer is a convolution layer which is a feature extraction layer, it takes its input from the input layer or sampling layer. A pooling layer takes each feature map output from the convolutional layer and down-samples it, i.e., pooling layer summarizes a region of neurons in the convolution layer. Figure 2 shows the max-pooling technique

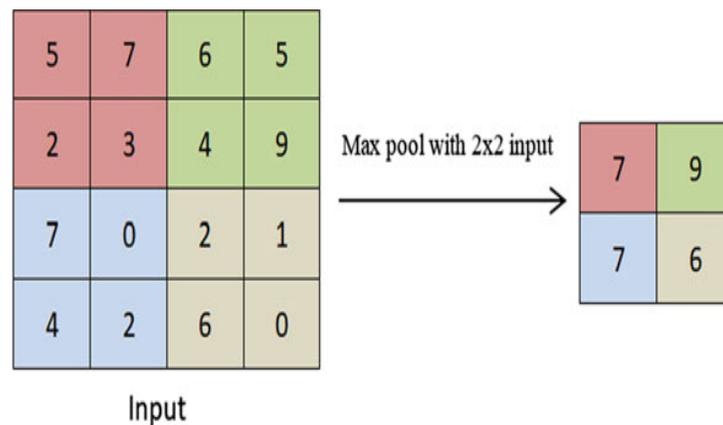


Figure 2: Max-Pooling Operation.

## **2.2 ReLU Layer**

Choosing a proper activation function is crucial since it significantly increases the performance of a CNN for certain applications. ReLU is one of the most commonly used function, this activation function produces a graph which is zero when  $x < 0$  and linear with slope 1 when  $x > 0$  [6, 8]. The remaining layers of convolution and subsampling perform similar to those in previous layers. Only difference is that the extracted features become more abstract as the depth increases.

## **2.3 FullyConnected Layer**

The process of convolution and pooling continues to detect the needed features. If there is a classification problem, then the detected features used in the spatial domain to obtain class score, i.e. the probabilities that these features represent each class [6].

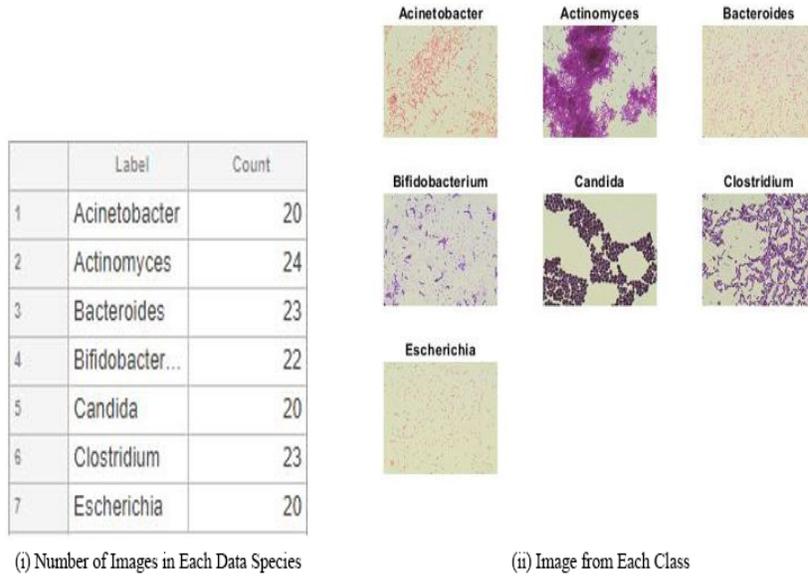
## **2.4 Dropout Layer**

Deep neural networks consist of multiple hidden layers used to learn more complicated features followed by fully connected layers for decision-making. A fully connected layer is connected to all features, and it is prone to overfitting. Overfitting refers to the problem when a model is trained and it works so well on training data that it negatively impacts the performance of the model on new data. In order to overcome the problem of overfitting, a dropout layer can be introduced in the model in which some neurons along with their connections are randomly dropped from the network during training. A reduced network is left; incoming and outgoing edges to a dropped-out node are also removed. Only the reduced network is trained on the data in that stage. The removed nodes are then reinserted into the network with their original weights. Dropout notably reduces overfitting and improves the generalization of the model [6, 9, 10].

# **3 Simulation Results**

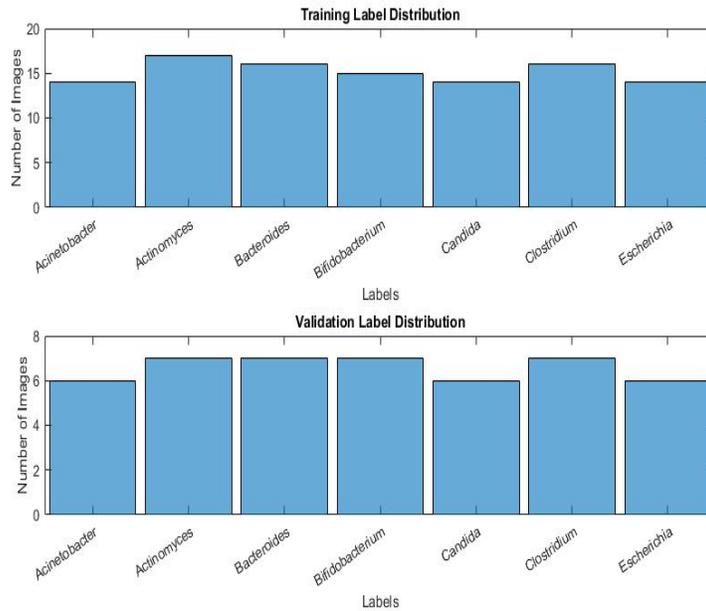
## **3.1 Data Set**

The data images consist of 152 images with 7 different genera and species of bacteria. The data images are collected from Digital Image of Bacterial Species (DIBaS) dataset which contains 660 images with 33 different genera and species of bacteria [11]. Figure 3 shows the selected number of data images in each class and an image from each class.



**Figure 3: Data Set Images.**

The data set divided randomly into 70% of the data for training and 30% for validation. The training data will be used to train the network and the validation data will be used to test the network, so that training data contains 106 images and the validation data contains 46 images. Figure 4 shows a histogram distribution of the different class labels in the training and validation sets.



**Figure 4: Distribution of Different Class Labels in the Training and Validation Data Sets.**

### 3.2 Data Augmenter

An augmented image datastore has been used to automatically resize the training images. Additional augmentation operations have been applied on training images: randomly flip the training images along the vertical axis, and randomly translate them up to 30 pixels horizontally and vertically. Data augmentation helps prevent the network from overfitting and memorizing the exact details of the training images [7].

### 3.3 Network Training

The experiments were implemented with the following parameters: The maximum epoch is set to be 10 and the minibatch size is 15, wight optimization done with stochastic gradient descent algorithm with momentum.

From the training results shown in Figure 5 it can be noted that since the neural network is not trained enough (at the first training epochs), both accuracy and losses are poor. At some points, the training accuracy is significantly higher than the validation accuracy, which means at this point, the network is overfitting. In neural networks, the more training, the more improvement, both accuracy, and losses are improved after re-training, epoch after epoch, which leads to a good result after nine epochs. The training repeated serval times to calculate average accuracy and time; Table 1 shows the re-training results.

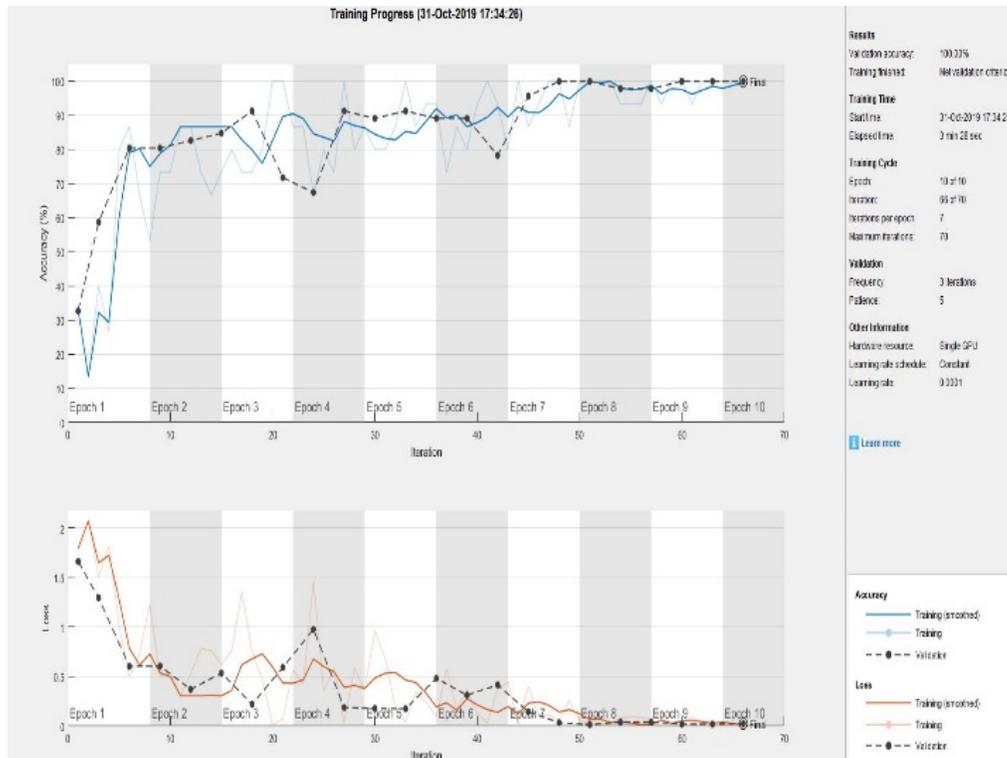


Figure 5: Training Progress.

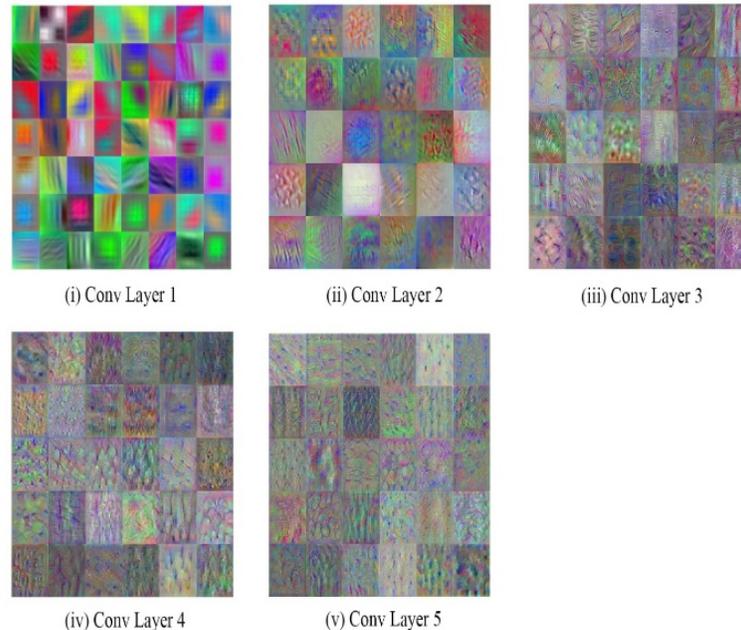
In total, the training consists of 10 epochs and 70 iterations; each epoch consists of 7 iterations to complete the full data set. The entire network learning takes average time equal to 4 min & 27 sec and the average accuracy was 98.45%, which considered excellent and fast training compared to training a neural network from scratch.

**Table 1:** Training progress results.

Training Progress	Completed Epochs	Mini Batch Size	Learning Rate	Accuracy	Time
1	10	15	1e-4	100%	3min&15sec
2	10	15	1e-4	100%	6min&33sec
3	7	15	1e-4	97.83%	2min&52sec
4	9	15	1e-4	95.65%	3min&32sec
5	10	15	1e-4	100%	6min&19sec
6	8	15	1e-4	95.65%	3min&28sec
7	10	15	1e-4	100%	7min&16sec
<b>Average</b>				98.45%	4min&27sec

### 3.4 Learned Features

Convolution neural networks use features to classify images. What the network learns during training is sometimes unclear, the layers towards the end of the network have larger receptive field sizes and learn larger features. The channels at the convolution layer 1 mostly contain edges and colors, which indicates that the filters at the convolution layer 1 are edge detectors and color filters, while the layers which are deeper into the network yield more detailed filters. Figure 6 shows some of the learned feature maps in the convolution layers.



**Figure 6:** Convolution Layers Learned Features.

### 3.5 Classification Results

Figure 7 shows the confusion matrix of the network validation results. On the confusion matrix, the rows correspond to the predicted class (Output Class) and the columns correspond to the true class (Target Class). The diagonal cells correspond to observations that are correctly classified. The off-diagonal cells correspond to incorrectly classified observations. Both the number of observations and the percentage of the total number of observations are shown in each cell. Since the testing done with 100% accuracy model, it can be noticed that all the validation data images are classified correctly, and there's no overfitting during network classifying.

Sensitivity, Specificity, and Accuracy are the terms which are most commonly associated with a classification test and they statistically measure the performance of the test. In general, Sensitivity indicates, how well the test predicts one category and Specificity measures how well the test predicts the other category. The training has been repeated several times to measure the average sensitivity and specificity; Table 2 shows the sensitivity and the specificity of this model.

**Validation Data Confusion Matrix**

Output Class	Acinetobacter	Actinomyces	Bacteroides	Bifidobacterium	Candida	Clostridium	Escherichia	
Acinetobacter	6 13.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Actinomyces	0 0.0%	7 15.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Bacteroides	0 0.0%	0 0.0%	7 15.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Bifidobacterium	0 0.0%	0 0.0%	0 0.0%	7 15.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Candida	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 13.0%	0 0.0%	0 0.0%	100% 0.0%
Clostridium	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 15.2%	0 0.0%	100% 0.0%
Escherichia	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 13.0%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	Acinetobacter	Actinomyces	Bacteroides	Bifidobacterium	Candida	Clostridium	Escherichia	
	Target Class							

**Figure 7: Confusion Matrix.**

From the table and on average, the sensitivity tells us that 100% of the acinetobacter, bacteroides, candida, clostridium or escherichia images were correctly identified, and 93.87% of the actinomyces images were correctly identified, and 95.91% of the bifidobacterium images were correctly identified.

**Table 2: Sensitivity and specificity of the model.**

	Acinetobacter	Actinomyces	Bacteroides	Bifidobacterium	Candida	Clostridium	Escherichia
<b>Avg. Sen.</b>	100%	93.87%	100%	100%	100%	100%	100%
<b>Avg. Spec.</b>	100%	100%	100%	98.9%	100%	99.2%	100%

From the table and on average, the specificity tells us that 100% of the non-acinetobacter, non-actinomyces, non-bacteroides, non-candida or non-escherichia images were correctly identified, and 98.90% of the non-bifidobacterium images were correctly identified, and 99.62% of the non- clostridium images were correctly identified

#### 4 CONCLUSION

Convolutional neural networks are biologically inspired networks that are used in computer vision for image classification and object detection. The most fundamental advantage of CNN is the automated feature extraction for a given mission. In general, when using such a network, a machine learning designer will need to think about the specific features for the task and assign each of them to an element of the input vector. On the other hand, a convolutional neural network appears to drive these features automatically. This becomes incredibly useful for tasks such as object recognition in which it is difficult for human-machine learning designers to come up with relevant features for the task.

A convolution neural network based approach has been applied for the classification of a variety of bacteria species was implemented In this paper. The proposed ConvNet architecture will automatically identify images of seven different bacteria genera and species with very high accuracy.

Also, instead of training a convolution neural network from scratch, a pre-trained model - AlexNet - was trained on a large dataset and tuned to the appropriate task. Transfer Learning is an effective important method of object recognition as it decreases training time and produces an improved overall performance.

#### References

- [1] B. Zieliński *et al*, “Deep Learning Approach to Bacterial Colony Classification”, *PLoS ONE* 12(9), 2017.
- [2] H. Yalcin and S. Razavi, “Plant Classification using Convolutional Neural Networks”, *IEEE Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)* 2016.
- [3] S. Trattner *et al*, “Automatic identification of bacterial types using statistical imaging methods”, *IEEE transactions on medical imaging*. 2004; 23(7):807±820. <https://doi.org/10.1109/TMI.2004.827481> PMID: 15250633.
- [4] N. Blackburn *et al*, “ Rapid determination of bacterial abundance, biovolume, morphology, and growth by neural network-based image analysis. *Applied and Environmental Microbiology*”. 1998
- [5] K. De Bruyne, *et al*, “Bacterial species identification from MALDI-TOF mass spectra through data analysis and machine learning. *Systematic and applied microbiology, Systematic and Applied Microbiology* 2011; 34(1):20±29. <https://doi.org/10.1016/j.syapm.2010.11.003> PMID: 21295428

- [6] M. Wani et al, "Advances in Deep Learning", *Studies in Big Data* 57, Springer Nature Singapore Pte Ltd. 2019.
- [7] A. Krizhevsky, *et al*," ImageNet Classification with Deep Convolutional Neural Networks". *Communications of the ACM*, 2017.
- [8] C. Aggarwal, "Neural Networks and Deep Learning", *Springer International Publishing*, 2018.
- [9] N. Srivastava, *et al*,"Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, 2014.
- [10] L. Huang, W. Tu, "Novel neural network application for bacterial colony classification", *Theoretical Biology and Medical Modelling*, 2018.
- [11] Digital Image of Bacterial Species (DIBaS) dataset, <http://misztal.edu.pl/software/databases/dibas/> (accessed on the 27th of September 2020).