IJRCS
International Journal of Robotics and Control Systems
Vol. 4, No. 2, 2024, pp. 849-876
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Deep Learning-Based Automated Approach for Classifying Bacterial Images

Ahmed Jaber Abougarair [a,1,*], Abdulhamid A. Oun [a,2], Salah I. Sawan [a,3], Alfian Ma'arif [b,4]

[a] Electrical and Electronics Engineering, University of Tripoli, Tripoli, Libya
[b] Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] a.abougarair@uot.edu.ly; [2] a.oun@uot.edu.ly; [3] s.sawan@uot.edu.ly; [4] alfian.maarif@te.uad.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received April 23, 2024
Revised June 02, 2024
Accepted June 06, 2024

Keywords
AlexNet;
Bacteria;
CNN;
DIBaS;
GoogleNet;
Transfer Learning

ABSTRACT

Identifying and classifying bacterial species from microscopic images is crucial for medical applications like prevention, diagnosis, and treatment. However, because of their diversity and variability in appearance, manually classifying bacteria is difficult and time-consuming. This work suggests employing deep learning architecture to automatically categorize bacterial species in order to overcome these difficulties and raise the accuracy of bacterial species recognition. We have evaluated our suggested approach using the Digital Images of Bacteria Species (DIBaS), a publicly accessible resource of photographs of tiny bacteria. This work uses a dataset that differs in terms of bacterial morphology, staining methods, and imaging circumstances. This paper aims to enhance the accuracy and reduce the computational requirements for Convolutional Neural Networks (CNN) based classification of bacterial species using GoogLeNet and AlexNet to train the models. This paper focuses on employing transfer learning to retrain pre-trained CNN models using a dataset consisting of 2000 images encompassing 12 distinct bacteria species known to be harmful to human health. The concept of transfer learning was utilized to expedite the network's training process and enhance its categorization performance. The results are promising, with the method achieving an accuracy of 98.7% precision, recall of 99.50%, and an F1-score of 99.45% with classifier speed. Furthermore, the proposed bacteria classification approach demonstrated strong performance, irrespective of the size of the training data used. This paper contributes by automating bacterial classification to facilitate faster and more accurate identification of bacterial species, which facilitates the treatment of infections and related diseases, in addition to monitoring public health, and promoting the wise use of antimicrobial drugs. To improve outcomes in the future, researchers can also integrate deep learning techniques with other machine learning methods.

1. Introduction

Bacteria are ubiquitous, inhabiting a vast array of environments both within and outside the human body. These microbial organisms exhibit a remarkable diversity in their physical characteristics, including variations in appearance, shape, and size. While many bacterial species are beneficial to humans, some can have harmful effects and cause a variety of diseases [1].

Beneficial bacteria play crucial roles, such as aiding in food digestion and supporting drug production processes. Conversely, harmful bacterial species are a primary cause of numerous diseases that impact human health. Consequently, the accurate classification and identification of bacterial species is of paramount importance, especially in fields like medicine, biochemistry, the food industry, and agriculture. Healthcare practitioners can get important insights into medical prevention, diagnosis, and treatment techniques by being aware of the exact genera and species of bacteria. However, because of their microbial variety, small size, and imperceptibility to the human sight, identifying and classifying bacteria is a process that is intrinsically difficult [2]. Conventional laboratory approaches for identifying bacterial strains depend on expensive and intricate molecular biology or biochemical procedures. These methods usually need a great deal of sample preparation, which is handled by skilled experts or microbiologists. The shape and size of bacterial cells are the most obvious visual characteristics used to identify them. Three primary shape-based forms of bacteria may be roughly identified: spiral, spherical, and cylindrical. This categorization scheme, however, is limited since various bacteria might have comparable sizes or shapes, and even distinct species of bacteria can have different physical characteristics [3].

A selection of images of tiny microorganisms taken from an accessible dataset is shown in Fig. 1. This picture makes it very evident that size and form are not reliable markers for classifying bacteria. Furthermore, although bacteria can occasionally look solitary, they can also occasionally form clusters. Because of this, morphology is typically regarded as an additional crucial characteristic for the categorization of bacteria. However, as Fig. 1 illustrates, while many bacterial species exhibit morphological diversity, others are difficult to distinguish from one another. Even for highly qualified or experienced professionals, the categorization of bacteria is extremely challenging because of these difficulties.

Classifying bacteria is an important subject in many domains, including environmental monitoring, food safety, and clinical diagnosis. For the purposes of environmental risk assessment, illness diagnosis, and outbreak prevention, accurate and rapid bacterial identification is crucial. However, the suggested deep learning-based strategy seeks to overcome a number of drawbacks and difficulties that standard bacterial classification approaches frequently encounter. The morphological features of bacteria vary greatly, encompassing changes in size, shape, and cell organization. It can be challenging to correctly identify between various bacterial species or strains based just on their outward appearance, particularly when working with closely related or morphologically identical bacteria. Different staining procedures are frequently used to prepare bacterial samples for microscopic inspection, which might affect the bacteria's appearance [4]. Furthermore, the bacterial images may exhibit diversity due to factors including illumination, resolution, and imaging equipment, which makes it challenging to create a reliable categorization system. Efficient and automated bacterial categorization is essential for prompt decision-making in dietary, environmental, and healthcare contexts. Now that deep learning and computer vision have made such amazing strides recently, it is conceivable to automate microorganisms. Prior to the development of deep learning, there were several obstacles in the way of accurately interpreting and classifying medical reports or photos because of restrictions in publicly available datasets. However there has been a notable upsurge in published papers with cutting-edge research using deep-learning methods in the medical field in recent years. These advancements have changed the game, making it more and more possible to use artificial intelligence to automate jobs and replicate human behavior [5].

Deep artificial neural networks called CNNs are utilized for object detection based on image visualization, similarity grouping, and image categorization. In addition to ensuring dependability, the use of CNNs to extract feature properties in an interpretable manner enables the evaluation of model authenticity based on training data without incurring influence from individual intuition. Beyond scholarly investigations, deep learning is widely used in data-intensive industries. Transfer learning reduces training time and improves performance on smaller domain-specific datasets, such as bacteria photos, by utilizing pre-trained CNN models on big datasets like ImageNet [6]. By utilizing cutting-edge computational approaches, this work proposes a computer-aided recognition system particularly intended for identifying bacterial genera and species. Our objective is to create an effective and precise system that automates the recognition process. With the use of this approach, bacteria will be precisely categorized into the genera and species to which they belong.

Prior research has exhibited the efficacy of AlexNet and GoogLeNet in many biomedical picture classification assignments, including but not limited to histological examination, cellular identification, and medical diagnosis. These models are particularly well-suited for the bacterial classification issue, which frequently entails differentiating between morphologically identical bacterial species, because of their capacity to detect minute visual patterns and generalize to new data [5]. The study can make use of the distinct advantages and architectural distinctions between AlexNet and GoogLeNet by utilizing both of them simultaneously. Since the two models may capture various features of the bacterial imaging data, this enables a more thorough assessment of the resilience and effectiveness of deep learning-based bacterial categorization. Both GoogLeNet and AlexNet underwent pre-training on extensive natural picture datasets, including ImageNet, that encompass a wide range of visual patterns and textures. With only a small amount of bacterial image data, this pre-training helps the models acquire broadly applicable low- and mid-level visual characteristics that are useful when applying them to the job of bacterial classification. The bacterial dataset may be small, therefore substantial training on it may not be necessary. However, the model's performance can be greatly enhanced by the transfer learning technique [7].

This work is organized into six primary sections. An overview of pertinent literature is provided in Section 2. The issue of machine learning (ML) based approaches for classifying bacteria is covered in detail in Section 3. The findings of the simulations used for categorization are shown in Section 4, which follows. Section 5 provides a summary of the main conclusions and their consequences for the article. Finally, Section 6 presents the limits and further research.

2. Related Work

In recent research, it has been demonstrated that deep learning methods can be effectively employed in the development of medical applications utilizing diverse algorithms. These applications have found utility in various settings, including hospitals, veterinary clinics, and the food industry, where they serve as image classification tools in both medical and biological contexts. In [8], the authors successfully performed bacterial classification by utilizing a pretrained DenseNet-201 model. The findings demonstrated that DenseNet-201 achieving an impressive accuracy rate of 99.2%. Similarly, in [9], a study focused on the identification and classification of longitudinal bacteria through transfer learning with ResNet-18. The results showed ResNet-18 to accurately classify 99% of the bacteria. Another research effort [10] employed a (CNN) model, which exhibited a 95% accuracy in identifying five species of bacteria of training data for identifying COVID-19, viral pneumonia. Reference [11] employed ResNet-18 and ResNet-50 models to classify a dataset comprising approximately 660 images of 33 bacterial species, achieving an impressive accuracy rate of 99.35%. Moreover, CNN models have been successfully utilized for automated counting of bacterial colonies without human intervention [12]. In a particular study [13], a ML approach was utilized to classify a dataset of 2,520 Klebsiella images. The results showcased an accuracy of 96.71%. In [14], the authors investigated the performance of a CNN in automatically detecting coronaviruses using transfer learning. The study achieved an accuracy of 96.7%. Transfer learning has been effectively employed in the detection, identification, and classification of bacteria. For instance, in [15], bacterial classification was performed using atrous convolutions and transfer learning, with the proposed atrous transfer learning model achieving a commendable classification accuracy of 95%.

Furthermore, [16] used a pre-trained ResNet and, without the need for any data augmentation, attained the greatest accuracy 99.2%. Another study [17] introduced a generative adversarial network (GAN) that utilized transfer learning for automated detection of COVID-19-induced pneumonia. Among the evaluated models, ResNet exhibited the highest performance, achieving an impressive accuracy of 99% on the test dataset. To overcome the limited availability in [18], researchers explored different ML techniques with the objective of constructing an ensemble-based feature extractor and selector to enhance the classification of microscopic bacterial images. [19] presented a practical approach to address the selection of deep learning models for workpiece classification and defect detection.

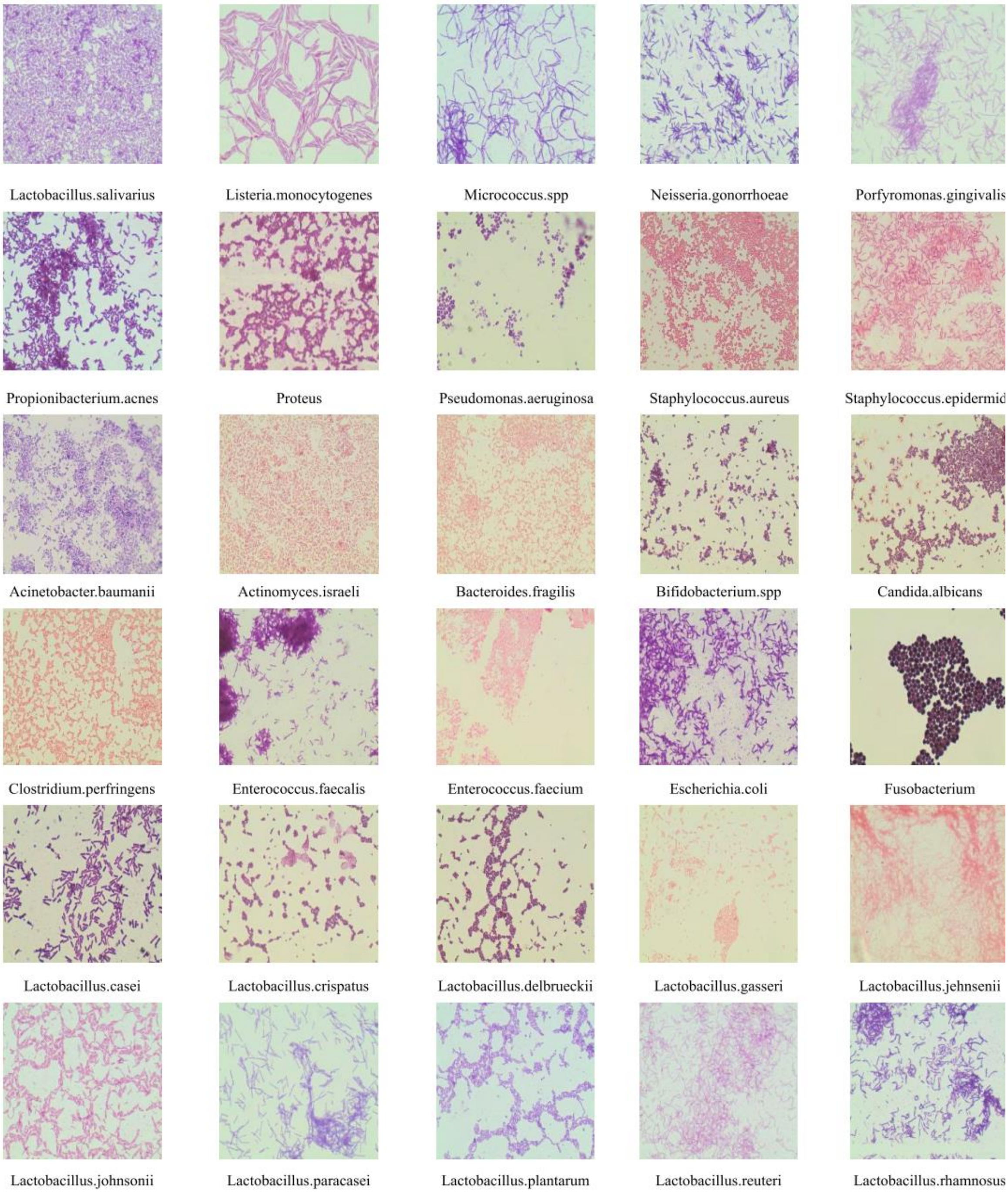

Lactobacillus.salivarius Listeria.monocytogenes Micrococcus.spp Neisseria.gonorrhoeae Porfyromonas.gingivalis
Propionibacterium.acnes Proteus Pseudomonas.aeruginosa Staphylococcus.aureus Staphylococcus.epidermid
Acinetobacter.baumanii Actinomyces.israeli Bacteroides.fragilis Bifidobacterium.spp Candida.albicans
Clostridium.perfringens Enterococcus.faecalis Enterococcus.faecium Escherichia.coli Fusobacterium
Lactobacillus.casei Lactobacillus.crispatus Lactobacillus.delbrueckii Lactobacillus.gasseri Lactobacillus.jehnsenii
Lactobacillus.johnsonii Lactobacillus.paracasei Lactobacillus.plantarum Lactobacillus.reuteri Lactobacillus.rhamnosus

Fig. 1. Distinct genera and species of bacteria from the DIBas dataset [4]

The study identified six research gaps by carefully examining the strengths and weaknesses of various algorithms in the literature. In [20], CNN was employed to recognize distinct bacterial colonies by analyzing their visual characteristics in images taken by cell phone cameras under different settings. The aim of this approach was to accomplish the task without the need for specialized expertise or the shipping of physical samples. In [21], a hybrid model called CNN-SVM was utilized, resulting in an accuracy of 98.7%. This accuracy was higher compared to other machine learning approaches used in the study. Additionally, by combining a CNN with Naïve Bayes, an accuracy of 99.9% was achieved. Furthermore, a study by researchers in [22] proposed a deep transfer learning approach that involved training and validating on X-ray images using ResNet-50, DenseNet-121, and VGG-19. The achieved accuracy was 94%, demonstrating the potential of transfer learning in identifying COVID-19 and conventional pneumonia in chest X-ray images. In another study, the objective was to detect pathogenic bacteria in food and water at an early stage using computational live bacterial detection. This was accomplished through the implementation of a deep neural network as described in [23]. Additionally, researchers in [24] introduced MotilityJ, an open-source tool developed using deep transfer learning, for detecting bacterial spread. The results demonstrated that the proposed tool achieved up to 100% accuracy in segmenting bacterial colonies. Ref. [25] introduced a new data augmentation technique based on the idea of artificial zooming as well as a classification utilizing MobileNetV2. They said that MobileNetV2 was used to obtain an accuracy of 97.38%. The DIBaS dataset was also used recently [26] to classify microorganisms automatically. They created a binary classification system with the ability to discriminate between gram-positive (GP) and gram-negative (GN) bacteria. Additionally, [27] suggested a methodology for classifying bacteria using a capsule network that works better with pictures that have a variety of forms in them. According to their experimental results with the DIBaS dataset, the maximum accuracy of 96.08% with a precision of 95.98% was attained. In contrast, a quantum-based CNN for the categorization of bacteria was proposed in the research [28]. Additionally, the authors tested their suggested approach on the DIBaS, precisely categorizing six groups of harmful bacteria, and were able to reach a 96.54% accuracy rate.

3. ML Based Bacteria Classification Methods

Conventional categorization techniques are often prone to errors. When it comes to classifying microscopic images such as those depicting diabetes, oral cancer, cervical cancer cells, microorganisms, etc. ML approaches have produced encouraging results. This provides scientists with inspiration and a place to start when classifying various bacteria in microbiology using similar effort. Machine learning techniques have shown to be more successful than conventional methods, including benefits like efficiency, precision, affordability, and ease of use. Many researchers have produced using machine learning techniques [18]. As a branch of AI, ML finds widespread use in sentiment analysis, voice recognition, medical diagnosis, financial services, signal processing, fretting fatigue analysis, emotional analysis, and network packet categorization. An ML model typically consists of two phases: testing and training [29]. In the testing phase, the learning model employs the execution engine to create predictions for the previously unknown data. In the training phase, samples are used as input, and features are learnt via learning algorithms to form the model. ML may be divided into three groups: reinforcement learning, unsupervised learning, and supervised learning.

3.1. Convolutional Neural Networks (CNNs)

CNNs are the brains of computer vision and image recognition applications. “Computer vision” is the area of AI that allows computers to interpret and process pictures and other visual data. CNNs use three-dimensional input to carry out tasks like object recognition and picture categorization. CNNs, an enlarged version of artificial neural networks, are mostly employed for feature extraction from matrix datasets that resemble grids. As seen in Fig. 2 [30], a CNN is made up of several layers, including the input layer, pooling layer, convolutional layer, and fully connected layers.

There are three primary categories of layers in them [11], [30]:

- Convolutional layer: The different characteristics from the input images are initially extracted using this layer. In this layer, we extract features from the input image using a kernel approach or a filter.
- Pooling layer: The primary objective of this layer is to reduce computing costs by reducing the convolved feature map. This may be achieved by working on each feature map independently and reducing the connections between layers. Various types of pooling procedures exist, contingent upon the method utilized. There is average and maximum pooling available.

- Fully-connected (FC) layer: The FC layer, which also contains the weights and biases, connects the neurons between two distinct layers. These layers are often placed ahead of the output layer and comprise the last few tiers of a CNN architecture.

For a bacteria classification task using machine learning, the following performance metrics can be calculated to evaluate the model's performance:

- Accuracy: This is the total percentage of the model's correct predictions. A general idea of the model's capacity to accurately categorize bacterial samples is provided by accuracy.
- Precision: The percentage of real positive predictions among all the model's positive predictions is called precision. In situations when erroneous positive forecasts are expensive or undesired, precision is especially crucial.
- Recall (Sensitivity): The percentage of accurate positive predictions among all real positive occurrences is measured by recall.
- F1-Score: This balanced statistic takes into account both measurements and is calculated as the harmonic mean of recall and accuracy. When you want a single statistic that takes into account the accuracy and recall of the model, the F1-score comes in handy.
- Confusion Matrix: By displaying the quantity of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), a confusion matrix offers a thorough analysis of the model's performance. This can assist in pinpointing the precise regions in which the model is having difficulty or excelling.
- The receiver operating characteristic's area under the curve (ROC-AUC): The model's capacity to discriminate between positive and negative classes is gauged by the ROC-AUC score. When working with unbalanced datasets or needing to modify the decision threshold, this statistic is helpful.

For the proposed bacterial classification model, the following five performance metrics were applied, as described in Table 1.

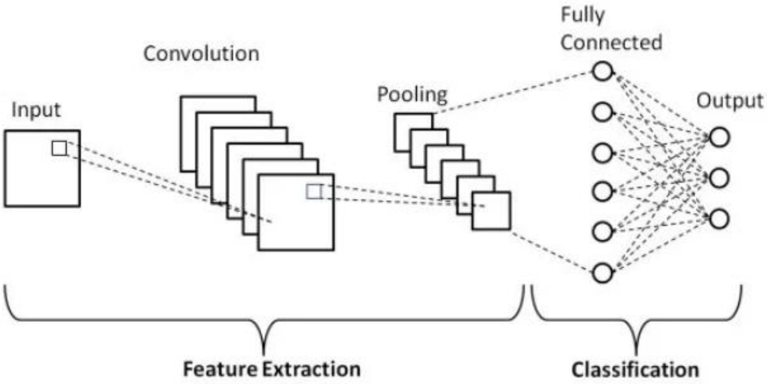


Fig. 2. CNN architecture

Table 1. Performance measurements

Measurement	Formula/Description
Accuracy	$\frac{TN+TP}{TN+FP+TP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1 Score	$2*\frac{Precision*Recall}{Precision*Recall}$ (with weighted average)
AUC_ROC	Area under the receiver operating characteristic curve

The steps depicted in Fig. 3 are followed when employing image processing to detect microorganisms and starting with setting up the bacterial database, which is required for the data collection procedure [32].

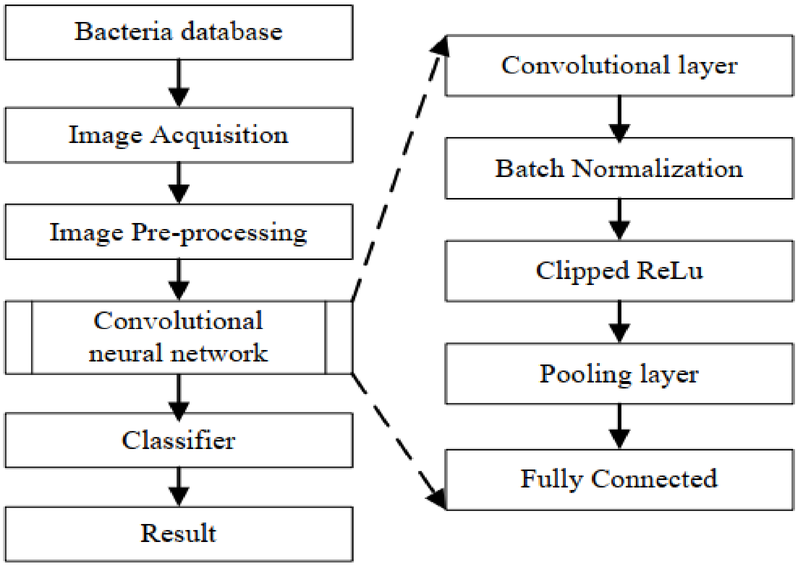


Fig. 3. Bacteria classification

3.1.1. AlexNet Network

AlexNet is a CNN that has made a significant impact in the field of machine learning, particularly in the application of deep learning for machine vision tasks. The structure of AlexNet comprises several layers, including convolutional, max-pooling, and fully connected layers [31]. The initial layer is a convolutional layer that generates 96 images, each with a size of 55 × 55 pixels. This is followed by a max-pooling layer, which produces an output of size 27 × 27 × 96 by using a window size of 3 × 3 and a stride of 2. The subsequent layer is another convolutional layer with 256 filters of size 5 × 5. It applies a stride of 1 and a padding of 2. Layer 4 is a down-sampling layer, or max-pooling layer, that results in an output size of 13 × 13 × 256. It uses a window size of 3 × 3 and a stride of 2 [32].

Layer 5 of AlexNet is a convolutional layer that produces feature maps of size 13 × 13 × 384 using 384 filters of size 3 × 3. It applies a stride of 1 and a padding of 1. The sixth layer is another convolutional layer with 384 filters of size 3 × 3, a stride of 1, and a padding of 1.

Layer 7 follows as yet another convolutional layer, comprising 256 filters of size 3 × 3, a stride of 1, and a padding of 1. The eighth layer is a down-sampling, or max-pooling, layer with a window size of 3 × 3 and a stride of 2.

Layer 9 represents the first fully connected layer, consisting of 4096 neurons. It is followed by a second fully connected layer, also with 4096 neurons, in Layer 10. The eleventh layer serves as the final fully connected layer, containing 1000 neurons that correspond to the 1000 classes in the ImageNet dataset. The final layer is a softmax layer, utilized for classification purposes. The architecture diagram of AlexNet can be seen in Fig. 4 [33].

This paper chose AlexNet CNN architectures for the bacterial species classification task for the following reasons:

- AlexNet is one of the pioneering and influential CNN models, known for its groundbreaking performance in the ImageNet competition.
- The researchers may have chosen AlexNet as a baseline or comparison model to evaluate the improvements achieved by their proposed optimizations and techniques.
- The relatively simpler architecture of AlexNet, compared to GoogLeNet, may have made it a suitable choice for exploring the impact of architectural changes and optimization strategies.

The essential layers of AlexNet topologies and their roles in enhancing the models' overall usefulness are as follows [35]:

1- Convolutional Layers:

- The convolutional layers in AlexNet are responsible for extracting low-level to high-level visual features from the input images.
- The early convolutional layers (e.g., Conv1, Conv2) capture simple, low-level features like edges, shapes, and textures.
- The later convolutional layers (e.g., Conv3, Conv4, Conv5) aggregate these low-level features into more complex, higher-level representations, such as object parts and semantic components.

2- Activation Functions:

- Non-linear activation functions, such as ReLU (Rectified Linear Unit), which add non-linearity and allow the model to learn more intricate patterns, come after the convolutional layers.

3- Pooling Layers:

- The pooling layers, such as Max Pooling, perform downsampling and dimensionality reduction by extracting the most salient features from the feature maps.
- This helps to reduce the number of parameters in the model, making it more computationally efficient and less prone to overfitting.

4- Fully Connected Layers:

- The final layers of AlexNet are fully connected layers, which take the high-level features extracted by the convolutional and pooling layers and use them for classification.

The fully connected layers map the flattened feature representations to the final output classes, effectively performing the classification task.

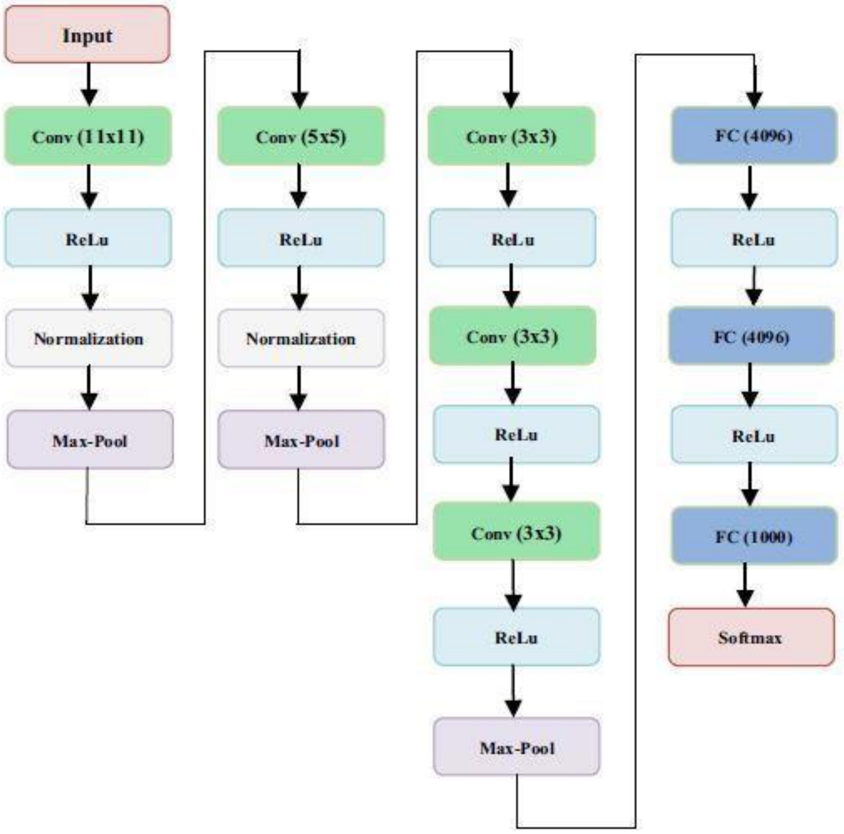


Fig. 4. Architecture diagram of AlexNet [34]

3.1.2. GoogleNet Network

GoogleNet distinguishes itself from previous state-of-the-art architectures like AlexNet and ZF-Net through the incorporation of various techniques. These techniques include 1×1 convolutions and global average pooling, which enable the creation of a deeper architecture with improved capabilities. Fig. 5 provides a visual representation of the architectural details of GoogLeNet. The utilization of 1×1 convolutions is a notable feature of the Inception architecture employed by GoogleNet. These convolutions are employed to reduce the number of parameters in the architecture, resulting in increased depth. By decreasing the parameters, the architecture can capture more expressive representations, leading to enhanced performance [36].

In contrast to previous architectures like AlexNet, GoogleNet replaces fully connected layers at the end of the network with global average pooling. This layer takes a 7×7 feature map and computes its average, resulting in a 1×1 output. This approach reduces the number of trainable parameters to zero and improves top-1 accuracy by 0.6%. The Inception module within GoogleNet differs from those in previous architectures. It incorporates parallel 1×1, 3×3, and 5×5 convolutions, as well as 3×3 max pooling, at the input. The outputs of these operations are concatenated to generate the final output. This design allows the network to effectively handle objects at multiple scales, as filters of different sizes capture different levels of detail. GoogleNet incorporates auxiliary classifier branches as an integral part of its architecture, strategically devised for training purposes. These branches are comprised of a 5×5 average pooling layer with a stride of 3, a 1×1 convolution employing 128 filters, two fully connected layers with 1024 and 1000 outputs respectively, and a softmax classification layer. The losses produced by these branches are combined with the overall loss, with a weight of 0.3 assigned to them. These auxiliary classifiers effectively tackle the issue of gradient vanishing while also contributing to regularization during the training process [37].

The GoogleNet architecture consists of a total of 22 layers, designed to be computationally efficient. The aim was to ensure that the architecture could run on individual devices, even those with limited computational resources. The auxiliary classifiers within the architecture have the following architectural details [38]:

- Average Pooling Layer: This layer performs average pooling on the input feature maps using a filter size of 5×5 and a stride of 3.
- 1×1 Convolution: Following the average pooling layer, a 1×1 convolution is applied to reduce the dimensionality of the feature maps. This convolutional layer employs 128 filters and uses the Rectified Linear Unit (ReLU) activation function.
- Fully Connected Layer: After the 1×1 convolution, a fully connected layer is employed with 1025 outputs and the ReLU activation function. This layer combines and refines the features extracted from the previous layers.
- Dropout Regularization: Dropout regularization is applied with a dropout ratio of 0.7 to prevent overfitting. During training, this technique randomly drops out units from the fully connected layer to improve generalization.
- Softmax Classifier: The final component of the auxiliary classifier is a softmax classifier with 1000 classes output, similar to the main softmax classifier. It assigns probabilities to each class, enabling classification based on the learned features.

For the purpose of classifying bacterial species, this article selected GoogLeNet CNN architectures for the following reasons:

- GoogLeNet is a well-established and highly effective CNN model that has demonstrated impressive performance in various image classification tasks.
- The researchers likely selected GoogLeNet due to its depth, efficient use of parameters, and proven ability to handle complex image datasets.

- GoogLeNet's “Inception” modules, which allow it to extract features at multiple scales, were likely seen as particularly suitable for capturing the diverse characteristics of bacterial colonies.

The essential layers of GoogLeNet topologies and their roles in enhancing the models' overall usefulness are as follows [39]:

- Convolutional Layers: From the input images, GoogLeNet's convolutional layers extract both high-level and low-level visual characteristics. These layers enable the model to learn representations and patterns that are helpful for picture classification tasks by applying a series of learnable filters to the input.
- The Inception module: a customized convolutional block that executes parallel convolutions with filters of various sizes (1x1, 3x3, 5x5), is the fundamental invention of GoogLeNet. This enhances the model's capacity to handle a wide range of intricate visual patterns by enabling it to simultaneously record elements at many sizes.
- Pooling Layers: GoogLeNet uses pooling layers, such max-pooling and average-pooling, to downscale the input by reducing its spatial dimensions. This strengthens the model's resistance to slight distortions and translations in the input data.
- Fully Connected Layers: The last layers of GoogleNet are fully connected layers that provide the final classification outputs using the previously obtained flattened feature maps. These layers support the model's final prediction-making process by integrating the learnt visual information.

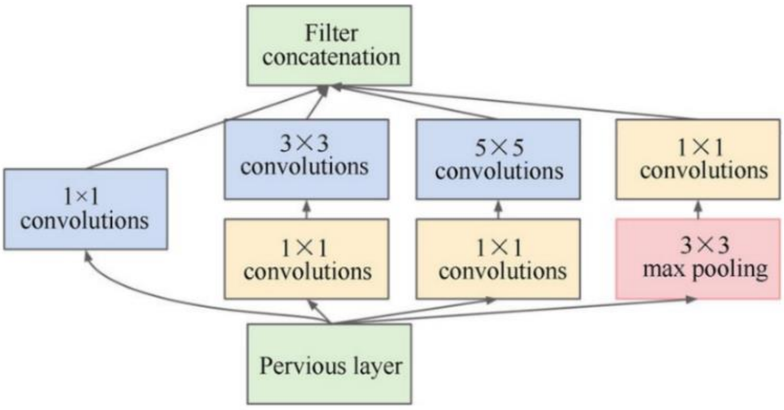


Fig. 5. Architectural details of GoogLeNet [39]

3.1.3. Transfer Learning

A machine learning technique called transfer learning seeks to improve performance on a new task or domain by applying information from a related but unrelated task or domain. Transfer learning allows us to utilize pre-existing models that have been trained on large datasets and have picked up general patterns and characteristics, saving us the trouble of starting the model training process from zero. The fundamental concept underlying transfer learning is that a model trained on a particular task or domain can capture valuable representations and knowledge that can be advantageous for a different task or domain [40].

By leveraging the knowledge already present in the pre-trained model, we can achieve improved performance even when the labeled data available for the new task is limited. During the training of a neural network, the weights or parameters of the network must be initialized with initial values. The selection of an appropriate technique for weight initialization is crucial as it can greatly influence both the rate at which the network converges and its final accuracy. If the weights are initialized randomly or arbitrarily, it can result in slow convergence or hinder the learning process altogether. Hence, it is vital to choose a reliable and effective method for weight initialization [41], [42]. Transfer learning is an approach to weight initialization that involves transferring the weights obtained from a pre-trained model to a target model. Instead of initializing the weights randomly, the target model inherits the weights that were previously learned. This process, known as transfer learning, entails extracting the learned weights from a trained base network (referred to as a pretrained model) and transferring them to an untrained target network, rather than starting the training of the target network from scratch [43]. Through this approach, the learned features from one network are transferred and utilized in another network that is specifically designed for a similar task.

The difference between Transfer learning versus traditional learning is seen in the Fig. 6. Both of the models in the above figure, which represent the classic method without transfer learning, are trained from scratch even if they are intended for distinct tasks. Conversely, in transfer learning, we train a pre-trained model that does a different job using our data set [44].

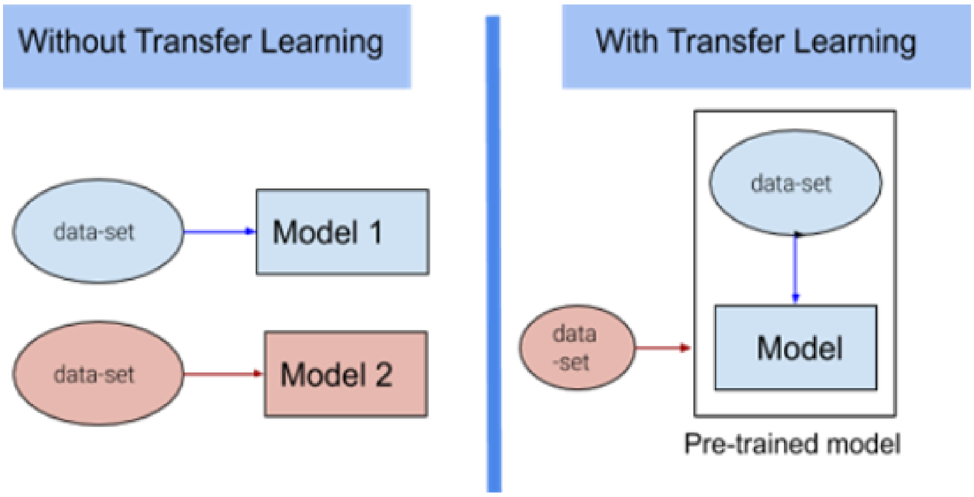


Fig. 6. Traditional learning vs transfer learning

We now present a few case studies and examples that demonstrate the use of transfer learning in fields related to the categorization of bacteria [45], [46]:

- Analysis of medical images: In the field of medical imaging, where large datasets are frequently required and substantial training data is needed for deep learning models, transfer learning has been applied widely.
- Plant disease detection: Another field that has profited from transfer learning is the categorization of plant diseases using photos of the leaves.
- Food safety and quality: Transfer learning has also been used for tasks involving food, such identifying food pollutants or classifying photos of food.
- Microbial identification: There are instances of transfer learning being used to identify other microorganisms, such as fungus, even though it is not directly connected to the categorization of bacteria.

This paper selected transfer learning for the following reasons [47]:

- The researchers stated that they “leverage transfer learning to build models using pre-trained data” before applying their optimization techniques.
- The researchers then fine-tune this pre-trained model on the specific task of bacterial species classification.
- This transfer learning approach allows the model to benefit from the general image recognition capabilities it has learned from the pre-training dataset and adapt them to the target bacterial classification problem.
- Transfer learning can significantly improve the performance and reduce the training time compared to training a model from scratch, especially when the target dataset (in this case, the bacterial images) is relatively small.

The following actions may be taken in order to apply transfer learning for the categorization of bacteria [46], [48]:

1. Select a Pre-Trained Model

- Select a pre-trained model that has been trained on a sizable dataset and is well-performing.
- The generic image attributes that these pre-trained models have acquired can be used to the task of classifying microorganisms.

2. Fine-Tuning the Pre-Trained Model

- Remove the final classification layer(s) of the pre-trained model, as the original output classes do not match the bacteria classification task.
- Add new fully connected layers at the end of the pre-trained model, with the final layer having a number of units equal to the number of bacteria classes in the target dataset.
- Initialize the weights of the new fully connected layers randomly, while keeping the weights of the pre-trained convolutional layers frozen.
- Train the model on the bacteria classification dataset, updating only the weights of the newly added layers.
- This allows the model to fine-tune the pre-trained features to the specific bacteria classification task, leveraging the knowledge learned from the large-scale pre-training dataset.

3. Layer Selection for Transfer

- Choose the right pre-trained model depth to apply to transfer learning.
- The pre-trained model's higher-level layers often capture more specialized, domain-dependent information, whereas the lower-level layers typically catch more general, domain-independent data.
- As the traits unique to each bacteria are likely to be recorded in these levels, it is advised to refine the pre-trained model's higher-level layers for the purpose of classifying bacteria.
- If the performance improves, you can gradually unfreeze more layers after fine-tuning only the last few.

4. Handling Domain-Specific Features

- Examine the dataset for the classification of bacteria to find any distinct, domain-specific features that the previously trained model could have missed.
- Some bacteria, for instance, could have distinct colors, textures, or forms that aren't included in the pre-training dataset. In these situations, you may use data augmentation methods to improve the bacterium dataset and assist the model in learning these domain-specific characteristics, such as random rotations, flips, or custom transformations.
- If the domain-specific characteristics change considerably from the pre-training dataset, you may also explore building a model from scratch or experimenting with other pre-trained models.

5. Hyperparameter Tuning and Evaluation

- To maximize the model's performance on the bacteria classification task, carefully adjust the hyperparameters, such as batch size learning rate, and number of epochs.
- To evaluate the model's performance, use suitable assessment measures like accuracy, precision, recall, and F1-score.

To guarantee the stability and generalizability of the model's output, take into account methods such as holdout validation or cross-validation.

3.2. Proposed Method

The section that follows gives more information about the dataset that was used to train and evaluate the CNN models for classifying different types of bacteria. In this study, we will apply the Digital Images of Bacteria Species (DIBaS) Dataset to classify digital bacteria images. It was released by Zielinski et al. in their 2017 article [49]. The dataset is publicly available and composed diffrent bacterial species.

Image Characteristics:

- The DIBaS dataset consists of high-resolution microscopic images of various bacterial species.
- The images were captured using standard staining techniques, such as Gram staining, to enhance the visual contrast and distinguish different bacterial morphologies.
- The image resolution is reported to be 2048×1532 pixels, providing a level of detail suitable for CNN-based classification.

Data Collection and Curation:

- The DIBaS dataset was curated by researchers from multiple institutions, collecting samples from clinical settings, research laboratories, and existing bacterial image repositories.
- The dataset was carefully reviewed and annotated to ensure accurate labeling of the bacterial species represented in each image.
- Efforts were made to include a diverse range of clinically relevant and commonly studied bacterial species in the dataset.

Dataset Size and Diversity:

- The DIBaS dataset contains a total of 19,764 images across 50 different bacterial species.
- This relatively large and diverse dataset provides a comprehensive resource for training and evaluating the performance of CNN models in bacterial species. Having a sizable dataset of images is crucial for effectively training and evaluating neural network models. In this section, the details of the dataset used in the study are discussed.

The dataset employed in this paper consists of 2000 images depicting 12 distinct bacterial genera and species. To enable proper training and validation of the neural network models, the dataset was randomly divided into two subsets:

- Training set: 70% of the 2000 images, totaling 1400 samples.
- Validation set: 30% of the 2000 images, totaling 600 samples.

This division allows the models to be trained on the training set and their performance to be evaluated on the validation set, which helps prevent overfitting.

A basic random split of the dataset into training and validation sets raises questions regarding possible bias or variability. It is true that stratified sampling is a superior method for guaranteeing that classes are fairly represented in both the training and validation sets. The following are some ways to deal with this problem: stratified sampling, repeated random sampling, cross-validation and holdout test set. We may increase the model evaluation's robustness and dependability by putting one or more of these tactics into practice. This will guarantee that the performance metrics provided appropriately represent the model's actual capabilities and potential for generalization.

Now we discuss the key data preprocessing techniques that can be important for a classification task involving bacteria images using Convolutional Neural Networks (CNNs) [50].

- The training images have been automatically resized using an enhanced image datastore. On training images, additional augmentation procedures have been implemented, such as random vertical axis flipping and random horizontal and vertical translation of up to 30 pixels. Data augmentation keeps the network from being overfit and from learning every nuance of the training pictures.

- Rescale the supplied photos' pixel values to a standard range, such [0, 1] or [-1, 1]. This keeps big pixel values from controlling the network, which can aid in the model's performance and speedier convergence.
- Utilize image processing methods to lessen the effect of noise or artifacts in the bacteria photos, such as median filtering or Gaussian smoothing. This can aid in the model's concentration on the pertinent elements for categorization.

The training process involves updating the model's weights using a method called backpropagation. This was done with a learning rate of 0.0001 and a batch size of 64. The models were trained for 350 epochs, which represents the number of complete passes through the training data. The learning rate is a crucial hyperparameter that needs to be tuned carefully [51]. A high learning rate can lead to convergence issues, while a very low learning rate may cause the algorithm to get stuck in a suboptimal local minimum, leading to poor generalization [52].

In summary, this section outlines the details of the bacterial image dataset used in the study and the procedures for preparing the training and validation sets, as well as the training process and the importance of the learning rate hyperparameter.

4. Simulation Results

In this section, we will explore and analyze the process of applying transfer learning techniques to the AlexNet and GoogLeNet convolutional neural networks. The objective is to utilize these pre-trained networks to classify different species of bacteria. Additionally, we will employ MATLAB R2023a to visualize the learned features and activations within the trained networks [53]-[56]. The results obtained from this analysis will be compared and evaluated to assess the effectiveness of the transfer learning approach in the context of bacterial species classification. Fig. 7 presents a flowchart outlining the key steps followed in this paper to classify various species of bacteria [57].

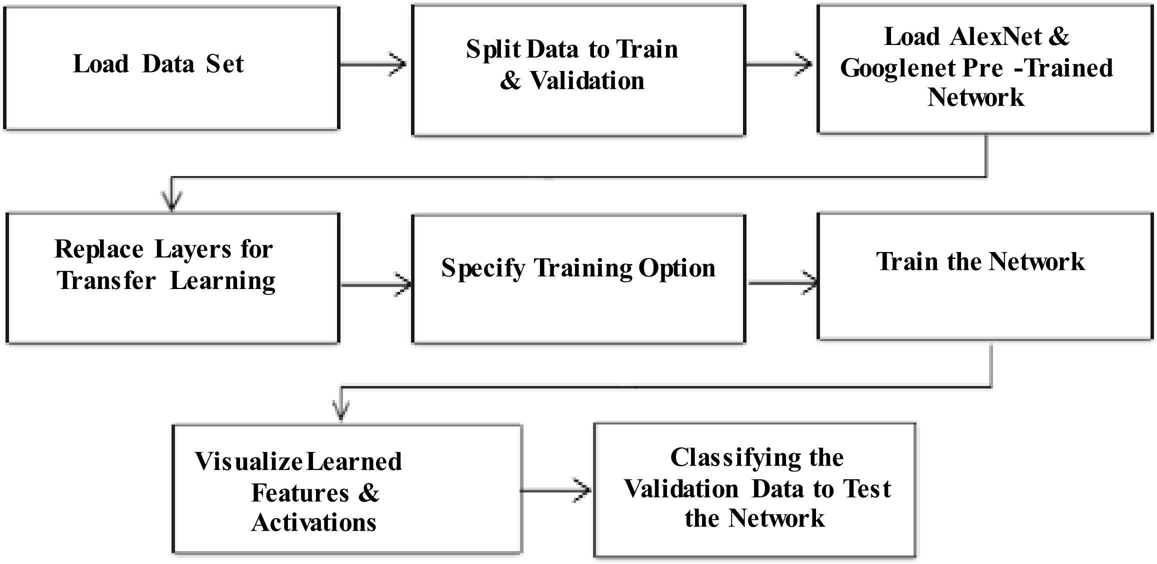


Fig. 7. Flowchart outlining the key steps followed to classify various species of bacteria

- Load Data Set: The data set is loaded into memory. It is essential to divide the data into training set and the validation set. The training set is utilized to train the model, while the validation set is employed to assess the model's performance during the training process [49], [58].
- Split Data to Train & Validation: This step splits the training data into two sets: training and validation sets. The typical split is 70% training and 30% validation.

- Load AlexNet/GoogLeNet: This step loads the AlexNet or GoogLeNet model architecture. AlexNet and GoogLeNet are CNN that were trained on the ImageNet dataset.
- Replace Layers for Transfer Learning: This step replaces some of the layers in the AlexNet or GoogLeNet model with new layers that are specific to the task that the model is being trained on.
- Specify Training Option: This step specifies the training options, such as the learning rate, optimizer, and number of epochs. The learning rate controls how quickly the model learns, and the optimizer is the algorithm used to update the model parameters [59], [60].
- Train The Network: This step trains the model on the training data.
- Visualize Learned Features & Activations: This step visualizes the features and activations that the model has learned. This can be helpful for understanding how the model is working.
- Classifying the Validation Data to Test the Network: This step classifies the validation data using the trained model. This helps to evaluate the model's performance on unseen data [61].

AlexNet ConvNet model consist of the following: Input layer, five Convolution layers, three Max-Pooling layers, seven ReLU Activation function layers, two Normalization layers, three fully connected layers, two dropout layers, and SoftMax and Output layer [62].

GoogLeNet ConvNet model consists of the following layers: Input layer, multiple inception modules, composed of various convolutional, pooling, and activation layers, multiple auxiliary classifiers for intermediate supervision, global average pooling layer, fully connected layers, dropout layers and Softmax and output layer [63].

Table 2 provides a concise comparison of AlexNet and GoogLeNet, highlighting their key features and distinguishing characteristics:

Table 2. Comparison of AlexNet and GoogLeNet networks structure

Feature	AlexNet	GoogLeNet
Architecture	Deeper, more traditional CNN	Inception modules, auxiliary classifiers
Computation complexity	Higher	Lower
Feature extraction	Extracts features at different scales	Extracts features from multiple dimensions
Overfitting resistance	More prone to overfitting	Less prone to overfitting
Performance	Good on image classification tasks	Better on image Net and other large-scale tasks

4.1. Replace Layers for Transfer Learning

The In the case of the pre-trained networks AlexNet and GoogLeNet, the last three layers are initially designed to handle classification tasks. These layers need to be fine-tuned to suit the new classification problem. To achieve this, the process involves extracting all layers except the last three from the pre-trained network. By doing so, the network retains its foundational layers that capture general features from the initial training. These extracted layers are then transferred and integrated into the new classification task. To align the network with the specific problem of 12 classes, the last three layers are replaced with new layers that are customized to accommodate the desired number of classes. However, the last three layers in GoogLeNet and AlexNet serve different purposes and have distinct functionalities [64], [65]:

1. AlexNet

Fully Connected Layers: AlexNet also includes fully connected layers towards the end of the network. These layers learn high-level representations and perform the final classification.

Dropout Layers: Dropout layers are utilized in AlexNet as a regularization technique to prevent overfitting. During training, dropout randomly sets a fraction of the input neurons to zero, forcing the network to learn redundant representations and increasing its robustness.

Softmax and Output Layer: Similar to GoogLeNet, AlexNet applies the softmax activation function to the output of the fully connected layers to generate class probabilities. The output layer represents the final predictions.

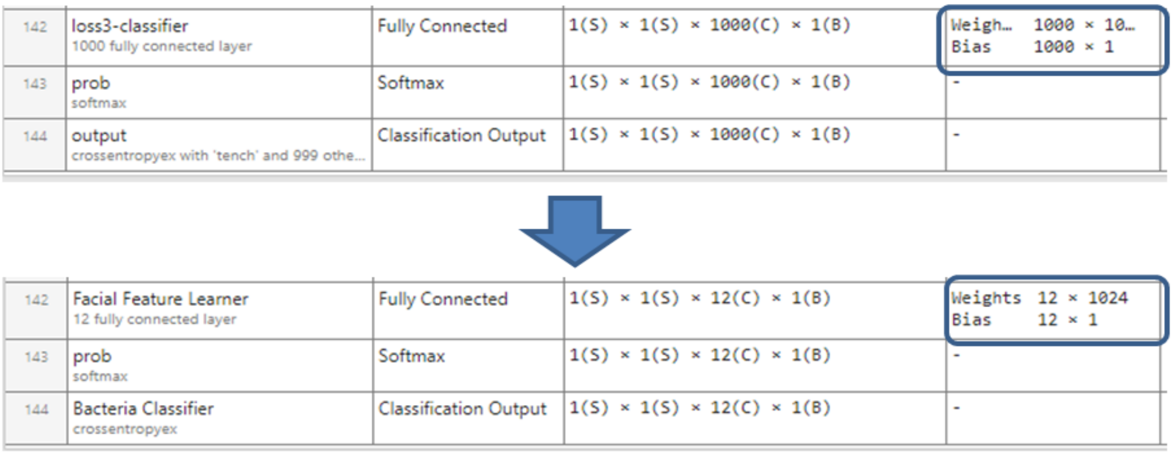
To summarize, while both GoogLeNet and AlexNet include fully connected layers and a softmax output layer, GoogLeNet incorporates a global average pooling layer to reduce spatial dimensions before classification. Additionally, AlexNet incorporates dropout layers to regularize the network and prevent overfitting.

2. GoogLeNet

Global Average Pooling Layer: This layer performs spatial pooling by taking the average of each feature map's values, reducing the spatial dimensions to a single value per feature map. It aids in capturing global contextual information and reducing the spatial dimensions before the final classification.

Fully Connected Layers: These layers are responsible for learning high-level representations and making predictions. They take the output of the previous layer and connect each neuron to every neuron in the subsequent layer. In GoogLeNet, these fully connected layers are typically followed by a softmax activation function to generate class probabilities.

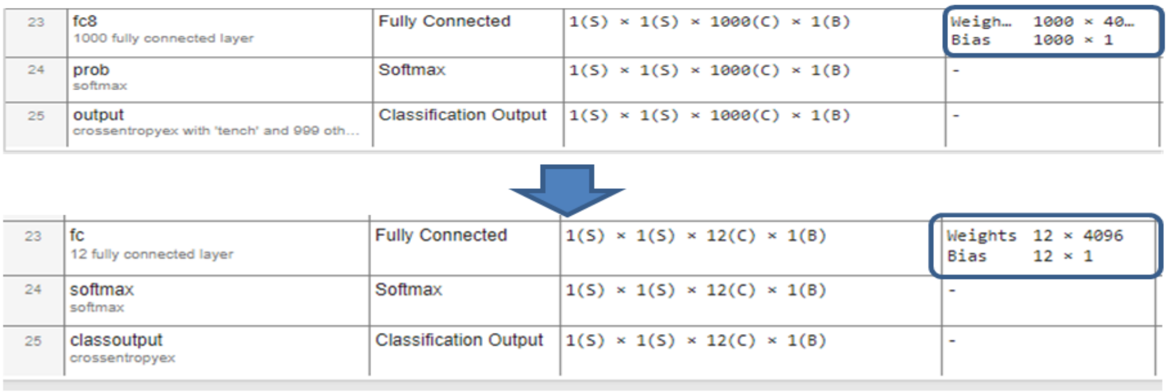
Softmax and Output Layer: The softmax activation function is applied to the output of the fully connected layers to produce a probability distribution over the classes. The output layer represents the final predictions, where each neuron corresponds to a specific class, and the highest activation indicates the predicted class. Fig. 8 shows the Alexnet's last 3 layers before and after fine-tuning. Fig. 9 illustrates the Googenet's last 3 layers before and after fine-tuning.



142	loss3-classifier 1000 fully connected layer	Fully Connected	1(S) × 1(S) × 1000(C) × 1(B)	Weigh… 1000 × 10… Bias 1000 × 1
143	prob softmax	Softmax	1(S) × 1(S) × 1000(C) × 1(B)	-
144	output crossentropyex with 'tench' and 999 othe…	Classification Output	1(S) × 1(S) × 1000(C) × 1(B)	-

142	Facial Feature Learner 12 fully connected layer	Fully Connected	1(S) × 1(S) × 12(C) × 1(B)	Weights 12 × 1024 Bias 12 × 1
143	prob softmax	Softmax	1(S) × 1(S) × 12(C) × 1(B)	-
144	Bacteria Classifier crossentropyex	Classification Output	1(S) × 1(S) × 12(C) × 1(B)	-

Fig. 8. Alexnet transferred layers



23	fc8 1000 fully connected layer	Fully Connected	1(S) × 1(S) × 1000(C) × 1(B)	Weigh… 1000 × 40… Bias 1000 × 1
24	prob softmax	Softmax	1(S) × 1(S) × 1000(C) × 1(B)	-
25	output crossentropyex with 'tench' and 999 oth…	Classification Output	1(S) × 1(S) × 1000(C) × 1(B)	-

23	fc 12 fully connected layer	Fully Connected	1(S) × 1(S) × 12(C) × 1(B)	Weights 12 × 4096 Bias 12 × 1
24	softmax softmax	Softmax	1(S) × 1(S) × 12(C) × 1(B)	-
25	classoutput crossentropyex	Classification Output	1(S) × 1(S) × 12(C) × 1(B)	-

Fig. 9. Googlenet transferred layers

4.2. Network Training

The training progress for the AlexNet network is visualized in Fig. 10, while the training progress for the GoogLeNet network is depicted in Fig. 11. These figures consist of two distinct curves, each representing different aspects of the training process. The first curve in both figures represents the accuracy per iterations or epoch. This curve quantifies the performance of the network by measuring its accuracy in correctly classifying the data samples during each iteration or epoch. It provides insights into how well the network is learning and improving over time. The second curve in both figures represents the losses per iterations or epoch. This curve indicates the amount of error or loss incurred by the network during each iteration or epoch. It reflects the discrepancy between the predicted outputs of the network and the actual ground truth labels. Monitoring this curve is essential for assessing the convergence and effectiveness of the training process [66], [67].

AlexNet

- Epoch 1-2: AlexNet rapidly learns the basic features of the bacteria images, leading to a significant decrease in training loss and a sharp increase in validation accuracy.
- Epoch 3-5: As AlexNet continues to learn more complex features, the training loss plateaus, and the validation accuracy increases gradually.
- Epoch 6-8: Overfitting starts to occur, causing the training loss and validation accuracy to plateau, indicating that the model is memorizing the training data rather than generalizing to unseen images.

Despite its simplicity, AlexNet, a convolutional neural network, quickly grasps the fundamental features of bacteria images during the initial training epochs. However, AlexNet's susceptibility to overfitting causes the validation accuracy to plateau early on, hindering its generalization performance. Key observations from the training process:

1. AlexNet's rapid decrease in training loss during the initial epochs indicates its ability to quickly grasp the fundamental features of bacteria images.
2. AlexNet's initial surge in validation accuracy followed by a plateau suggests its capacity to learn accurate bacteria classification but also its vulnerability to overfitting.
3. The convergence of both training loss and validation accuracy after 8 epochs signals that the model has reached its optimal learning state.

To mitigate AlexNet's overfitting tendency, early stopping is crucial to prevent the model from over-adapting to the training data, thereby enhancing its generalization capability.

AlexNet demonstrates proficiency in classifying bacteria images with minimal training time, but careful monitoring and early stopping are essential to combat its overfitting tendency and achieve improved generalization performance.

GoogLeNet

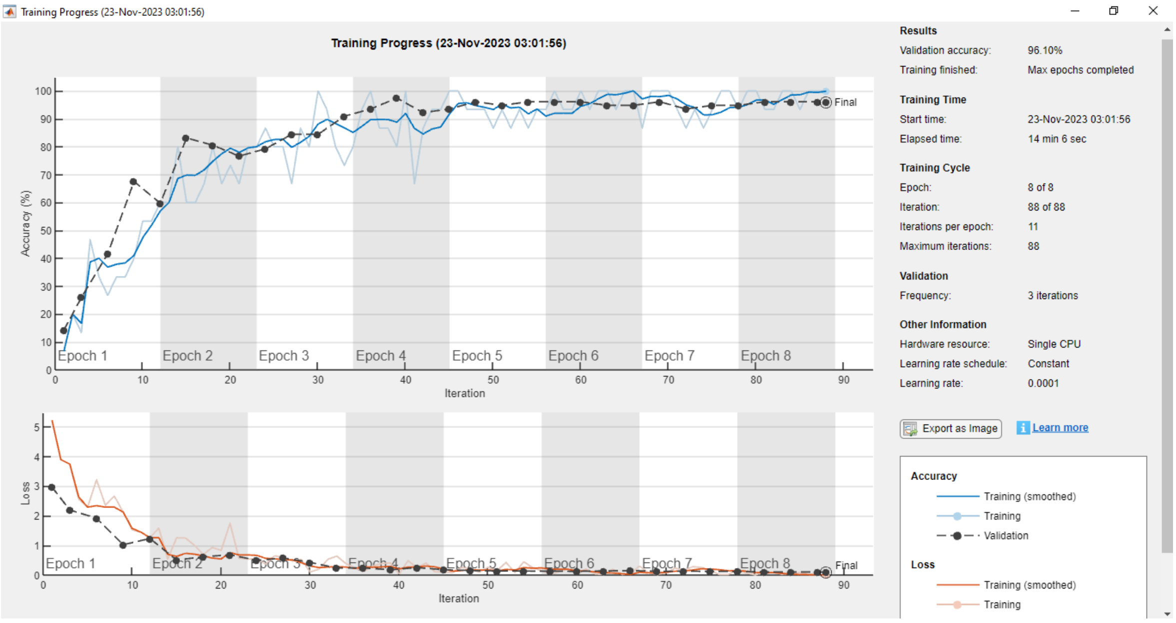
- Epoch 1-3: GoogLeNet's inception modules and auxiliary classifiers enable efficient feature extraction and reduce overfitting risk, leading to a rapid decrease in training loss and a corresponding increase in validation accuracy.
- Epoch 4-10: GoogLeNet refines its feature representations, causing the training loss to gradually decrease and the validation accuracy to steadily increase.
- Epoch 11-20: While the training loss plateaus, the validation accuracy continues to increase slowly, demonstrating GoogLeNet's ability to generalize effectively without overfitting.
- Epoch 21-30: The training loss and validation accuracy both plateau, indicating that the model has converged.

Despite its computational complexity, GoogLeNet, a more sophisticated network than AlexNet, exhibits reduced overfitting susceptibility, allowing for extended training up to 30 epochs without incurring overfitting limitations. Key observations from the training process:

1. GoogLeNet's gradual decrease in training loss throughout the training process indicates its ability to continuously refine its understanding of the bacteria images, extracting increasingly complex and invariant features.
2. GoogLeNet's steady increase in validation accuracy throughout the training process suggests its effectiveness in generalizing to unseen bacteria data.
3. The convergence of both training loss and validation accuracy after 30 epochs signals that the model has reached its optimal learning state.

The overall training process of GoogLeNet for bacteria classification demonstrates its capability to accurately classify bacteria images with ample training time of 30 epochs. Additionally, its reduced susceptibility to overfitting makes it a favorable choice for bacteria classification tasks.

GoogLeNet proves to be a robust and accurate classifier for bacteria images due to its ability to continuously learn complex features and its reduced overfitting tendency compared to AlexNet.



Results
Validation accuracy: 96.10%
Training finished: Max epochs completed

Training Time
Start time: 23-Nov-2023 03:01:56
Elapsed time: 14 min 6 sec

Training Cycle
Epoch: 8 of 8
Iteration: 88 of 88
Iterations per epoch: 11
Maximum iterations: 88

Validation
Frequency: 3 iterations

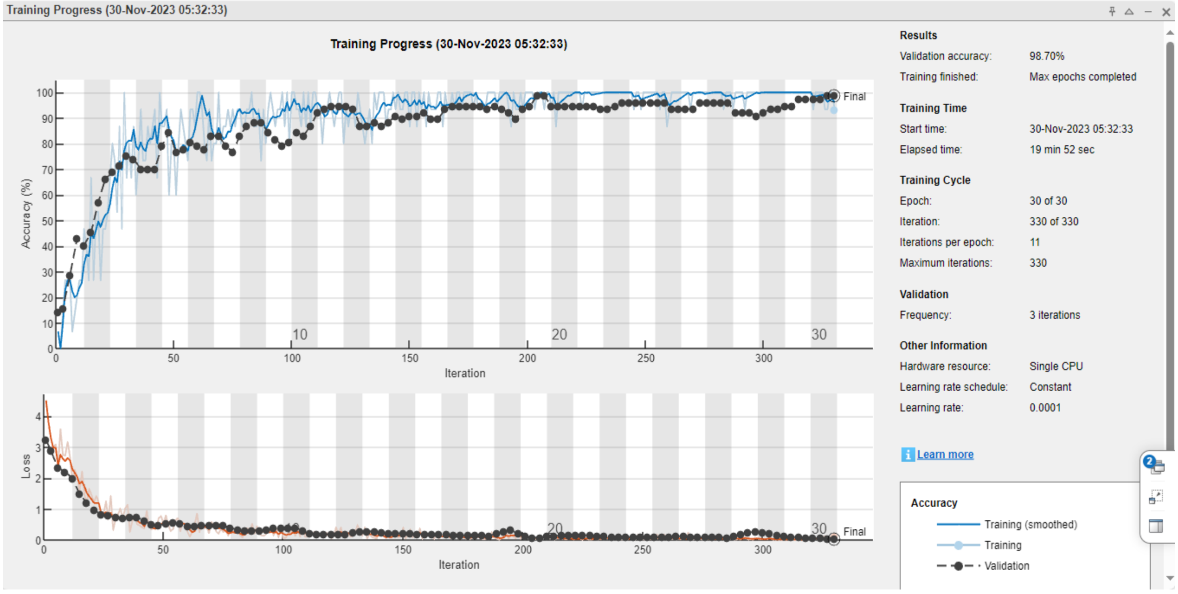
Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Fig. 10. Training progress for Alexnet network

Table 3 summarizes the key differences between the training processes for AlexNet and GoogLeNet. GoogLeNet outperforms AlexNet on bacteria classification, achieving a higher final validation accuracy and demonstrating less overfitting tendency.

Table 3. Comparison of alexnet and googlenet networks training process

Feature	AlexNet	GoogLeNet
Training loss progression	Rapid decrease, then plateau	Gradual decrease over all epochs
Validation accuracy progression	Rapid increase, then plateaus, then slow increase	Gradual increase over all epochs
Overfitting tendency	More prone to overfitting	Less prone to overfitting
Final validation accuracy	96.1%	98.7%
Computational complexity	Higher	Lower
Feature extraction capabilities	Extracts features at different scales	Extracts features from multiple dimensions
Overall performance	Good on image classification tasks	Better on ImageNet and other large-scale tasks



Results
Validation accuracy: 98.70%
Training finished: Max epochs completed

Training Time
Start time: 30-Nov-2023 05:32:33
Elapsed time: 19 min 52 sec

Training Cycle
Epoch: 30 of 30
Iteration: 330 of 330
Iterations per epoch: 11
Maximum iterations: 330

Validation
Frequency: 3 iterations

Other Information
Hardware resource: Single CPU
Learning rate schedule: Constant
Learning rate: 0.0001

Fig. 11. Training progress for Googlenet network

4.3. Visualize Activations

In this section, we analyze the behavior of a convolutional neural network by feeding it a testing image and observing the activations at different layers of the network [68]. The testing example used in this analysis is an image of Candida albicans, a species of bacteria, as shown in Fig. 12. To perform this analysis, we input the Candida albicans image into the convolutional neural network. As the image propagates through the network, each layer performs a series of convolutions, activations, and pooling operations to extract and transform features. At each layer, we capture and examine the activations, which represent the response of the network to specific patterns and structures in the input image [69], [70].

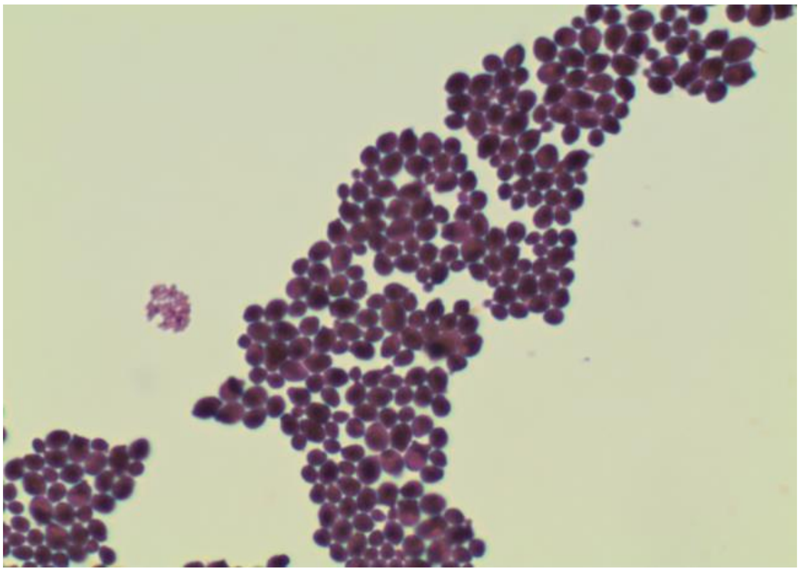
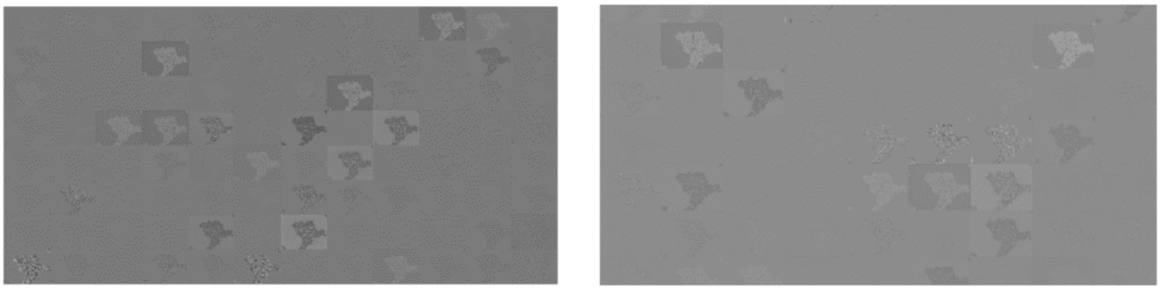


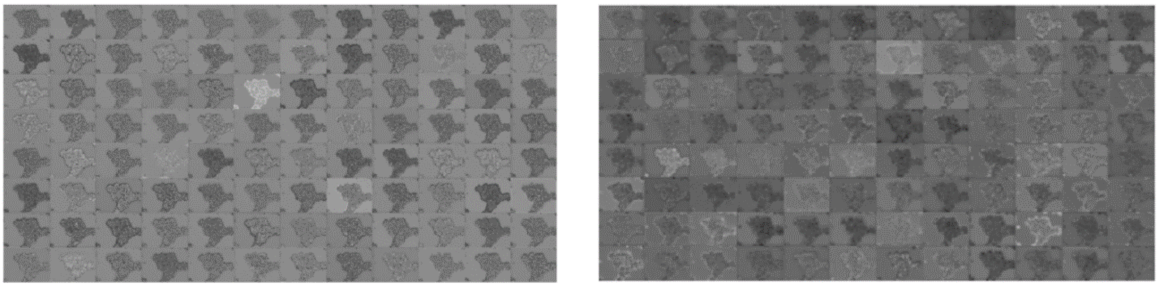
Fig. 12. Candida albicans bacteria

By examining the activation areas in the convolutional layers and comparing them to the corresponding regions in the original images, we can investigate the features learned by a convolutional neural network. This involves analyzing which areas become active when an image is processed by the network and comparing those areas to the relevant parts of the original image. Through this investigation, we can gain insights into how the network detects and represents important features in the input images. In Fig. 13, the activations of the initial and final convolutional layers in both AlexNet and GoogleNet are depicted. This visualization allows us to observe the response patterns within these layers and gain insights into how the networks process the input data. The initial convolutional layer of both networks produces activations that are highly localized and responsive to low-level features such as edges and orientations.

This is not surprising, as these features are essential for building a foundation for more complex feature extraction in subsequent layers. The final convolutional layer of both networks produces activations that are more global and responsive to higher-level features such as textures and shapes. This is because the final convolutional layer is the culmination of many layers of feature extraction, and it is able to leverage the activations of all previous layers to generate a comprehensive representation of the input image.



(i) AlexNet Conv layer 1 (i) GoogleNet Conv layer 1



(ii) AlexNet Conv layer 5 (ii) GoogleNet inception_5b-1x1 layer

Fig. 13. Convolution layers activations

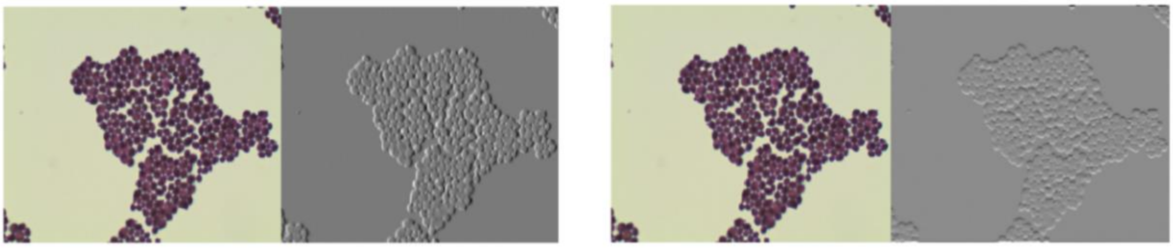
Fig. 14 showcases the strongest activation channels compared to the input image, can provide valuable insights into the features that are most salient and influential in the classification or analysis process. The image shows that the strongest activations are concentrated in the foreground of the image, where the Candida albicans cells are located. This suggests that the network is paying close attention to the morphological features of the cells, such as their shape, size, and arrangement. This is not surprising, as these features are essential for distinguishing Candida albicans from other bacteria species.

The figures show that convolutional neural networks (CNNs) learn to detect different types of features at different depths in the network. In the early layers, CNNs learn to detect simple features like edges and colors. In the deeper layers, CNNs learn to detect more complex and detailed features, such as textures and shapes. The later layers in the network build their features by combining features from earlier layers. This is why the channels in the later layers have stronger activations than the channels in the early layers. The stronger activations indicate that the network has learned more features from the input image.

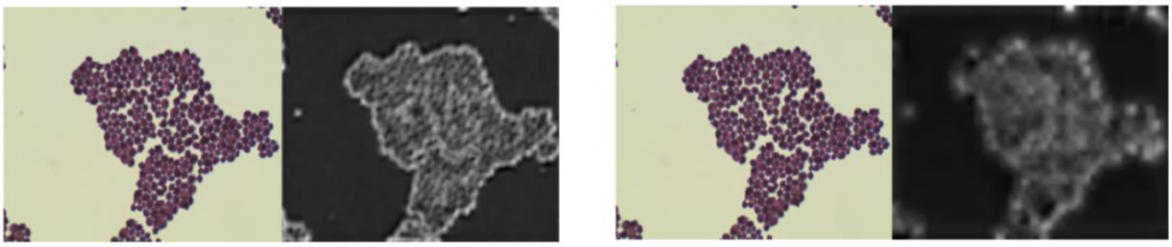
4.4. Rectified Linear Unit (ReLU) Layer Effect

By comparing the activation patterns in the last convolutional layer and the followed ReLU activation function layer in AlexNet and GoogleNet as shown in Fig. 15, some notable differences can be observed. In AlexNet, the activations in the relu5 layer are more localized, indicating its strength in extracting low-level features such as edges and orientations. On the other hand, GoogleNet's inception_5b-3x3_relu layer shows activations that are more global, suggesting its proficiency in extracting high-level features like textures and shapes.

These differences can be attributed to the architectural variances between the two networks. AlexNet is a shallower network with fewer layers, enabling it to extract features from a smaller receptive field and focus on localized features. In contrast, GoogleNet's deeper architecture allows it to extract features from a larger receptive field, facilitating the capture of more global features. Additionally, the overall appearance of the activation images differs between the two networks. AlexNet's relu5 layer exhibits more noise in the activations, possibly due to its traditional pooling layer, which can introduce noise. Conversely, GoogleNet's inception_5b-3x3_relu layer, which employs an inception module, shows reduced sensitivity to noise.



(i) AlexNet Conv layer 1 (i) GoogleNet conv1-7x7_s2 layer

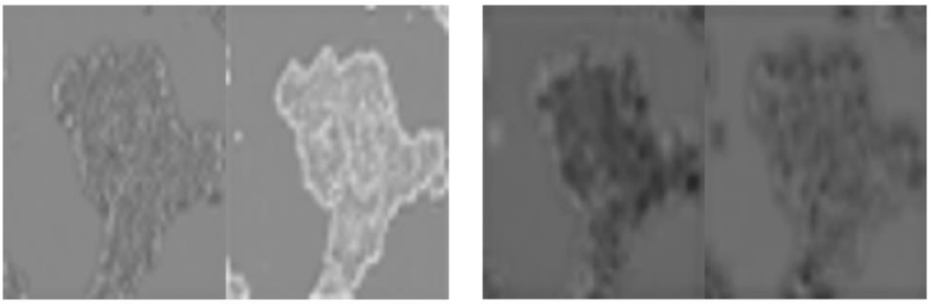


(ii) AlexNet Conv layer 5 (ii) GoogleNet inception_5b-1x1 layer

Fig. 14. Strongest activation channels

The ReLU activation function used in both AlexNet and GoogleNet plays a crucial role. By discarding negative activations, the function allows the networks to focus on positive activations and learn more complex patterns in the data. This enhances the networks' ability to extract important features and ignore less significant ones. The last convolutional layer and the subsequent ReLU activation function layer in AlexNet and GoogleNet are responsible for extracting high-level features in the input image. The ReLU activation function aids in making these features more non-linear and expressive, facilitating the learning of complex patterns and making accurate predictions. In summary, the activation patterns and the utilization of ReLU activation functions highlight the distinctive characteristics and strengths of AlexNet and GoogleNet in feature extraction and pattern recognition. Fig. 16 and Fig. 17 shows the confusion matrix of both the networks results.

Table 4 summarizes a comparison of the confusion matrices for CNN comparasion. In all four criteria, as you can see, GoogleNet performs better than AlexNet. This is due to the fact that GoogleNet can recognize more intricate patterns in the data than AlexNet can since it is a deeper and more complex network. GoogleNet outperforms AlexNet in almost all classes. The only exception is the Lactobacillus class, where AlexNet has a slightly higher accuracy. However, this is likely due to the small number of Lactobacillus images in the dataset. Overall, the comparison of the two confusion matrices shows that GoogleNet is a more powerful classification model than AlexNet. It is able to classify a wider range of images with greater accuracy.



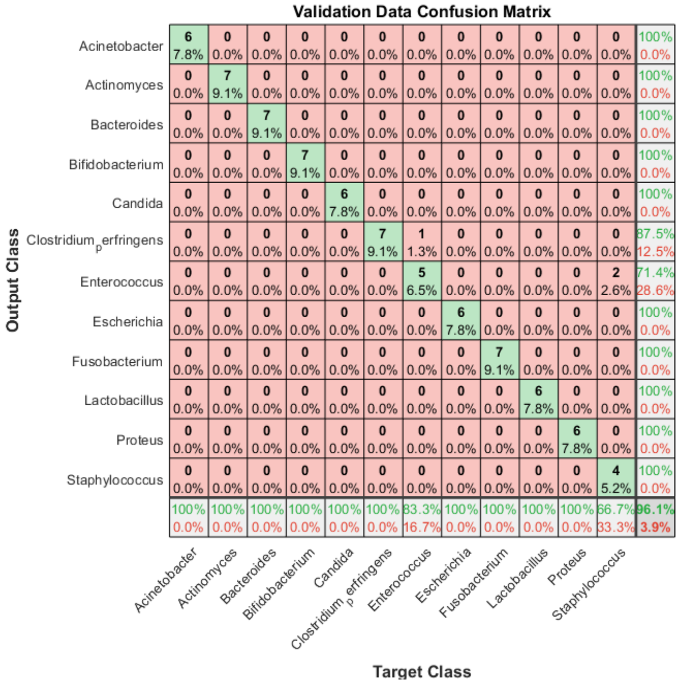
(i) AlexNet Conv layer 5 CH no. 30 & 44 (i) GoogleNet inception_5b-1x1 layer CH no. 30 & 44

(ii) AlexNet ReLU_layer CH no. 30 & 44 (ii) GoogleNet inception_5b-3x3_ReLU layer CH no. 30 & 44

Fig. 15. ReLU layer effect

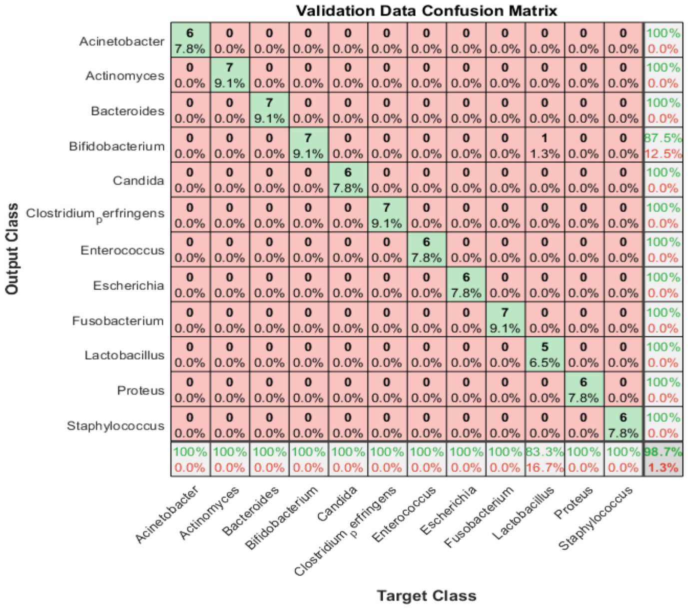
Fig. 16. CNN Comparison

Model	Training Accuracy	Time (minute)	Precision	Recall	F1 Score	MSE
Purposed Method AlexNet	96.1	14.6	98.3	98.3	98.7	0.00813
Purposed Method GoogLeNet	98.7	19.52	99.5	99.5	99.45	0.00351
Resnet8 [9]	98.3	9.45	99.1	99.1	98.91	0.00473
VGG16 [10]	97.65	40.13	98.7	98.6	98.65	0.00751
SqueezeNet [11]	98.1	7.05	99.5	99.5	99.8	0.00851



	Acinetobacter	Actinomyces	Bacteroides	Bifidobacterium	Candida	Clostridium$_p$erfringens	Enterococcus	Escherichia	Fusobacterium	Lactobacillus	Proteus	Staphylococcus	
Acinetobacter	6 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Clostridium$_p$erfringens	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 9.1%	1 1.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	87.5% 12.5%
Enterococcus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 6.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 2.6%	71.4% 28.6%
Escherichia	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Fusobacterium	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 9.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Lactobacillus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.8%	0 0.0%	0 0.0%	100% 0.0%
Proteus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.8%	0 0.0%	100% 0.0%
Staphylococcus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 5.2%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	83.3% 16.7%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	66.7% 33.3%	96.1% 3.9%

Fig. 17. AlexNet's confusion matrix



	Acinetobacter	Actinomyces	Bacteroides	Bifidobacterium	Candida	Clostridium$_p$erfringens	Enterococcus	Escherichia	Fusobacterium	Lactobacillus	Proteus	Staphylococcus	
Acinetobacter	6 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.3%	0 0.0%	0 0.0%	87.5% 12.5%
Candida	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Clostridium$_p$erfringens	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 9.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Enterococcus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	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%	0 0.0%	6 7.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Fusobacterium	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 9.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
Lactobacillus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 6.5%	0 0.0%	0 0.0%	100% 0.0%
Proteus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.8%	0 0.0%	100% 0.0%
Staphylococcus	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 7.8%	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%	100% 0.0%	83.3% 16.7%	100% 0.0%	100% 0.0%	98.7% 1.3%

Fig. 18. GoogleNet's confusion matrix

5. Conclusion

Bacterial categorization is an area where machine learning techniques have seen significant use. The use of these methods has enhanced the discipline of microbiology. Microbiologists benefit from machine learning (ML) techniques in a variety of ways, including the identification and categorization of bacteria as well as the general automation of these procedures. In this study, the application of transfer learning to the classification of bacteria using two well-known CNN architectures—GoogLeNet and AlexNetis shown. The bacteria were trained and evaluated to identify DIBaS pictures. The bulk of the primary research employ academic datasets, according to the results as well. In addition, the most often utilized performance assessment criteria were recall, accuracy, precision, F1-score, and confusion matrix. Researchers can benefit from this study since it helps to find areas in which they can make contributions. The training procedure takes between 14.6 and 19.52 minutes, with results in the range of 96.1% to 98.7% for accuracy and validation. The best accuracy errors were found at MSE 0.00351. Transfer learning with GoogLeNet emerges as a powerful tool for bacteria classification, offering improved accuracy and efficiency compared to transfer learning with AlexNet. Its deeper architecture, enhanced feature extraction capability, and reduced computational complexity make GoogLeNet a valuable asset for automating microbiology diagnostics, facilitating faster and more accurate identification of bacteria species, and ultimately aiding in disease prevention and treatment. Subsequent research endeavors ought to employ microscopic slide images procured from multi-center laboratories and employ DenseNet201 or another deeper model for training.

6. Limitations and Future Work

One disadvantage of our suggested technique with regard to the dataset utilized is that, while it showed encouraging results for the DIBaS dataset used in this work, its performance could vary when applied to other datasets. This variation can be linked to the distinct qualities and difficulties that different datasets bring, which may not coincide with the particulars of the DIBaS dataset.

Our study might be expanded to include a wider variety of datasets than only the DIBaS dataset as a possible direction for future research. This addition would solve the constraint of possible performance variability identified previously and enable us to evaluate the resilience and adaptability of our suggested strategy across datasets with different characteristics and problems. We may learn more about the generalizability and efficacy of the technique in various circumstances by examining a variety of datasets. More training data may yield a greater resilience and accuracy for the purposes of training a deep learning architecture. Subsequent research in this field can entail developing a different dataset specifically tailored for the application or using a more sophisticated data augmentation technique. Moreover, the models investigated in this work mostly depend on modifying the architectures and architectures of conventional CNNs. With the incorporation of image processing techniques like vision transformers, deep learning networks are already exhibiting advances in their structure and layout as the fields of deep learning and image processing advance. A comparison study may be helpful in determining optimality with reference to industry-specific applications when newer, more sophisticated algorithms are created.

Author Contribution: Ahmed J. Abougarair, A. Oun and S. Sawan wrote the manuscript. The final manuscript has been reviewed by Alfian Ma'arif.

Funding: We declare that the funding for this project was provided solely by the researchers, with no external financial assistance.

Acknowledgments: This study was carried out as part of a Research Project at the Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Tripoli. The authors extend their sincere thanks to the Universitas Ahmad Dahlan for their invaluable support during the research phase. Also, the authors express their gratitude to the original proprietors of the Digital Images of Bacteria Species (DIBaS) collection for making it available to the public. This dataset's availability has tremendously aided their study.

Conflict of Interest: The authors declare that they have no conflict of interest.

References

[1] S. Trattner, H. Greenspan, G. Tepper and S. Abboud, “Automatic identification of bacterial types using statistical imaging methods,” *IEEE Transactions on Medical Imaging*, vol. 23, no. 7, pp. 807-820, 2004, https://doi.org/10.1109/TMI.2004.827481.

[2] Y. Yuliza, R. Muwardi, M. Rhozaly, L. Lenni, M. Yunita, G. E. Yehezkiel, “Fast Human Recognition System on Real-Time Camera,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 4, pp. 895-903, 2023, http://dx.doi.org/10.26555/jiteki.v9i4.27009.

[3] A. Plichta, “Methods of classification of the genera and species of bacteria using decision tree,” *Journal of Telecommunications and Information Technology*, no. 4, pp. 74-82, 2019, https://doi.org/10.26636/jtit.2019.137419.

[4] S. Visitsattaponge, 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, https://doi.org/10.1109/ACCESS.2024.3358671.

[5] S. Poomrittigul, W. Chomkwah, T. Tanpatanan, S. Sakorntanant and T. Treebupachatsakul, “A Comparison of Deep Learning CNN Architecture Models for Classifying Bacteria,” *2022 37th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 290-293, 2022, https://doi.org/10.1109/ITC-CSCC55581.2022.9894986.

[6] Z. Wang *et al.*, “Deep neural networks offer morphologic classification and diagnosis of bacterial vaginosis,” *Journal of Clinical Microbiology*, vol. 59, no. 2, pp. 1-13, 2021, https://doi.org/10.1128/jcm.02236-20.

[7] D. -T. Mai and K. Ishibashi, “Bacteria Shape Classification using Small-Scale Depthwise Separable CNNs,” *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 2940-2943, 2021, https://doi.org/10.1109/EMBC46164.2021.9630658.

[8] S. A. Akbar, K. H. Ghazali, H. Hasan, Z. Mohamed and W. S. Aji, “An Enhanced Classification of Bacteria Pathogen on Microscopy Images Using Deep Learning,” *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 119-123, 2021, https://doi.org/10.1109/ISRITI54043.2021.9702809.

[9] C. Garcia-Perez, K. Ito, J. Geijo, R. Feldbauer, N. Schreiber, and W. Z. Castell, “Efficient detection of longitudinal bacteria fission using transfer learning in deep neural networks,” *Frontiers Microbiology*, vol. 12, p. 645972, 2021, https://doi.org/10.3389/fmicb.2021.645972.

[10] H. Sajedi, F. panah, and A. Pashaei, "Image-processing based taxonomy analysis of bacterial micromorphology using machine-learning models," *Multimedia Tools and Applications*, vol. 79, pp. 32711–32730, 2020, https://doi.org/10.1007/s11042-020-09284-9.

[11] T. Shaily and S. Kala, "Bacterial Image Classification Using Convolutional Neural Networks," *2020 IEEE 17th India Council International Conference (INDICON)*, pp. 1-6, 2020, https://doi.org/10.1109/INDICON49873.2020.9342356.

[12] A. Ferrari, S. Lombardi, and A. Signoroni, "Bacterial colony counting with convolutional neural networks in digital microbiology imaging," *Pattern Recognition*, vol. 61, pp. 629–640, 2017, https://doi.org/10.1016/j.patcog.2016.07.016.

[13] B. D. Satoto, M. I. Utoyo, R. Rulaningtyas, and E. B. Koendhori, "Classification of features shape of Gram-negative bacterial using an extreme learning machine," *IOP Conference Series: Earth and Environmental Science*, vol. 524, no. 1, p. 012005, 2020, https://doi.org/10.1088/1755-1315/524/1/012005.

[14] I. Apostolopoulos and T. Bessiana, "COVID-19: Automatic detection from X-ray images utilizing transfer learning with convolutional neural networks," *Physical and Engineering Sciences in Medicine*, vol. 43, pp. 635–640, 2020, https://doi.org/10.1007/s13246-020-00865-4.

[15] S. Patel, "Bacterial colony classification using atrous convolution with transfer learning," *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 4, pp. 1428-1441, 2021, http://annalsofrscb.ro/index.php/journal/article/view/2650.

[16] M. Talo, "An automated deep learning approach for bacterial image classification," *arXiv preprint arXiv*, 2019, https://arxiv.org/abs/1912.08765.

[17] N. E. M. Khalifa, M. H. N. Taha, A. E. Hassanien, S. Elghamrawy, "Detection of coronavirus (COVID-19) associated pneumonia based on generative adversarial networks and a fine-tuned deep transfer learning model using chest X-ray dataset," *Proceedings of the 8th International Conference on Advanced Intelligent Systems and Informatics 2022*, pp. 234-247, 2020, https://doi.org/10.1007/978-3-031-20601-6_22.

[18] S. Kotwal, P. Rani, T. Arif, J. Manhas, "Machine Learning and Deep Learning Based Hybrid Feature Extraction and Classification Model Using Digital Microscopic Bacterial Images," *SN Computer Science*, vol. 4, no. 5, p. 687, 2023, https://doi.org/10.1007/s42979-023-02138-9.

[19] C. Chen , A. Abdullah, S. H. Kok, D. T. K. Tien, "Review of Industry Workpiece Classification and Defect Detection using Deep Learning," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, pp. 329-340, 2022, https://dx.doi.org/10.14569/IJACSA.2022.0130439.

[20] S. A. Nagro *et al*., "Automatic Identification of Single Bacterial Colonies Using Deep and Transfer Learning," *IEEE Access*, vol. 10, pp. 120181-120190, 2022, https://doi.org/10.1109/ACCESS.2022.3221958.

[21] M. F. Wahid, M. J. Hasan, M. S. Alom and S. Mahbub, "Performance Analysis of Machine Learning Techniques for Microscopic Bacteria Image Classification," *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1-4, 2019, https://doi.org/10.1109/ICCCNT45670.2019.8944597.

[22] Y. Brima, M. Atemkeng and S. Djiokap, J. Ebiele, F. Tchakounte, "Transfer learning for the detection and diagnosis of types of pneumonia including pneumonia induced by COVID-19 from chest X-ray images," *Diagnostics*, vol. 11, no. 8, p. 1480, 2021, https://doi.org/10.3390/diagnostics11081480.

[23] H. Wang *et al*., "Early detection and classification of live bacteria using time-lapse coherent imaging and deep learning," *Light: Science & Applications*, vol. 9, no. 1, p. 118, 2020, https://doi.org/10.1038/s41377-020-00358-9.

[24] A. Casado-García *et al*., "MotilityJ: An open-source tool for the classification and segmentation of bacteria on motility images," *Computers in Biology and Medicine*, vol. 136, p. 104673, 2021, https://doi.org/10.1016/j.compbiomed.2021.104673.

[25] R. G. García, S. J. Rodríguez, B. B. Martínez, C. H. Gracidas, and R. M. Torres, "Efficient deep learning architectures for fast identification of bacterial strains in resource-constrained devices," *Multimedia Tools and Applications*, vol. 81, no. 28, pp. 39915-39944, 2022, https://doi.org/10.1007/s11042-022-13022-8.

[26] K. Kristensen, L. M. Ward, M. L. Mogensen, and S. L. Cichosz, “Using image processing and automated classification models to classify microscopic gram stain images,” *Computer Methods and Programs in Biomedicine Update*, vol. 3, p. 100091, 2023, https://doi.org/10.1016/j.cmpbup.2022.100091.

[27] J. P. Chaudhari, H. Mewada, A. V. Patel, and K. Mahant, “Automated bacteria genera classification using histogram-oriented optimized capsule network,” *Engineering Science and Technology, an International Journal*, vol. 46, p. 101500, 2023, https://doi.org/10.1016/j.jestch.2023.101500.

[28] I. Naz, J. H. Shah, M. H. U. Rehman, M. Rafiq and G. S. Choi, “Quantum Mechanism-Based Convolution Model for the Classification of Pathogenic Bacteria,” *IEEE Access*, vol. 11, pp. 137747-137757, 2023, https://doi.org/10.1109/ACCESS.2023.3339127.

[29] H. P. Desai, A. P. Parameshwaran, R. Sunderramanand and M. Weeks, “Comparative Study Using Neural Networks for 16S Ribosomal Gene Classification,” *Journal of Computational Biology*, vol. 27, no. 2, pp. 248-258, 2020, https://doi.org/10.1089/cmb.2019.0436.

[30] D. Sutaji and H. Rosyid, “Convolutional Neural Network (CNN) Models for Crop Diseases Classification,” *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 7, no. 2, pp.187-196, 2022, https://doi.org/10.22219/kinetik.v7i2.1443.

[31] H. Eldem, E. Ulker and O. Y. Iskl, “Alexnet architecture variations with transfer learning for classification of wound images,” *Engineering Science and Technology, an International Journal*, vol. 45, p. 101490, 2023, https://doi.org/10.1016/j.jestch.2023.101490.

[32] S. Lu, Z. Lu, Y.-D. Zhang, “Pathological brain detection based on AlexNet and transfer learning,” *Journal of Computational Science*, vol. 30, pp. 41-47, 2019, https://doi.org/10.1016/j.jocs.2018.11.008.

[33] H. Chen *et al.*, “AlexNet convolutional neural network for disease detection and classification of tomato leaf,” *Electronics*, vol. 11, no. 6, p. 951, 2022, https://doi.org/10.3390/electronics11060951.

[34] V. S. Kanimozhi, M. Balasubramani, R. Anuradha, “Hierarchal Bayes model with AlexNet for characterization of M-FISH chromosome images,” *Medical & Biological Engineering & Computing*, vol. 59, no. 7, pp. 1529-1544, 2021, https://doi.org/10.1007/s11517-021-02384-0.

[35] H. Eldem, E. Ülker and O. Işkl, “Effects of Training Parameters of AlexNet Architecture on Wound Image Classification,” *Traitement du Signal*, vol. 40, no. 2, pp. 811-817, 2023, https://doi.org/10.18280/ts.400243.

[36] R. Anand and T. Shanthi, M. S. Nithish, S. Lakshman, “Face Recognition and Classification Using GoogleNET Architecture,” *International Conference on Soft Computing for Problem Solving*, pp. 261-269, 2020, https://doi.org/10.1007/978-981-15-0035-0_20.

[37] V. Sunanthini *et al.*, “Comparison of CNN Algorithms for Feature Extraction on Fundus Images to Detect Glaucoma,” *Journal of Healthcare Engineering*, vol. 2022, 2022, https://doi.org/10.1155/2022/7873300.

[38] G. Xu *et al.*, "A Deep Transfer Convolutional Neural Network Framework for EEG Signal Classification," in *IEEE Access*, vol. 7, pp. 112767-112776, 2019, https://doi.org/10.1109/ACCESS.2019.2930958.

[39] L. Yang, ”GoogLeNet based on residual network and attention mechanism identification of rice leaf diseases,” *Computers and Electronics in Agriculture*, vol. 204, p. 107543, 2023, https://doi.org/10.1016/j.compag.2022.107543.

[40] T. Treebupachatsakul and S. Poomrittigul, “Bacteria Classification using Image Processing and Deep learning,” *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pp. 1-3, 2019, https://doi.org/10.1109/ITC-CSCC.2019.8793320.

[41] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, 2010, https://doi.org/10.1109/TKDE.2009.191.

[42] A. J. Abougarair, “Neural Networks Identification and Control of Mobile Robot Using Adaptive Neuro Fuzzy Inference System,” *ICEMIS'20: Proceedings of the 6th International Conference on Engineering & MIS 2020*, pp. 1-9, 2020, https://doi.org/10.1145/3410352.3410734.

[43] N. D. Kathamuthu *et al.*, “A deep transfer learning-based convolution neural network model for COVID-19 detection using computed tomography scan images for medical applications,” *Advanced Engineering Software*, vol. 175, p. 103317, 2023, https://doi.org/10.1016/j.advengsoft.2022.103317.

[44] A. J. Abougarair, M. J. Abujarir, H. T. Tilamon, “Artificial pancreas control using optimized fuzzy logic based genetic algorithm,” *International Robotics & Automation Journal*, vol. 9, no. 2, pp. 89-97, 2023, https://doi.org/10.15406/iratj.2023.09.00270.

[45] A. J. Abougarair *et al*., “Blood Cells Cancer Detection Based on Deep Learning,” *Journal of Advances in Artificial Intelligence*, vol. 2, no. 1, pp. 108-121, 2024, https://doi.org/10.18178/JAAI.2024.2.1.108-121.

[46] M. Hacibeyoglu, M. H. Ibrahim, “Human Gender Prediction on Facial Mobil Images using Convolutional Neural Networks,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 6, no. 3, pp. 203-208, 2018, https://doi.org/10.18201/ijisae.2018644778.

[47] A. A. Almisreb, N. M. Tahir, S. Turaev, M. A. Saleh, S. A. M. A. Junid, “Arabic Handwriting Classification using Deep Transfer Learning Techniques,” *Pertanika Journal Science Technol*, vol. 30, no. 1, p. 641, 2022, https://doi.org/10.47836/pjst.30.1.35.

[48] A. S. Emhemmed, A. J. Abougarair, O. Salih and S. S. Alsamlqi, “Simulation Analysis and Control of Wireless Power Transfer for Implantable Medical Devices,” *2022 IEEE 21st international Ccnference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 18-23, 2022, https://doi.org/10.1109/STA56120.2022.10019046.

[49] B. Zieliński, A. Plichta, K. Misztal, P. Spurek, M. Brzychczy-Włoch, D. Ochońska, “Deep learning approach to bacterial colony classification,” *PLoS ONE*, vol. 12, no. 9, p. e0184554, 2017, https://doi.org/10.1371/journal.pone.0184554.

[50] Y. Guo, Z. Pang, J. Du, F. Jiang and Q. Hu, “An Improved AlexNet for Power Edge Transmission Line Anomaly Detection,” *IEEE Access*, vol. 8, pp. 97830-97838, 2020, https://doi.org/10.1109/ACCESS.2020.2995910.

[51] A. J. Abougarair, M. K. I. Aburakhis, M. M. Edardar, “Adaptive Neural Networks Based Robust Output Feedback Controllers for Nonlinear Systems,” *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 37-56, 2022, https://doi.org/10.31763/ijrcs.v2i1.523.

[52] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander and M. S. H. Sunny, “Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review,” *IEEE Access*, vol. 7, pp. 13960-13988, 2019, https://doi.org/10.1109/ACCESS.2019.2894819.

[53] A. J. Abougarair, “Adaptive Neural Networks Based Optimal Control for Stabilizing Nonlinear System,” *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, pp. 141-148, 2023, https://doi.org/10.1109/MI-STA57575.2023.10169340.

[54] S. E. Elwefati, A. J. Abougarair and M. M. Bakush, “Control of Epidemic Disease Based Optimization Technique,” *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA*, pp. 52-57, 2021, https://doi.org/10.1109/MI-STA52233.2021.9464453.

[55] A. J. Abougarair, S. E. Elwefati, “Identification and Control of Epidemic Disease Based Neural Networks and Optimization Technique,” *International Journal of Robotics and Control System*, vol 3, no. 4, pp. 780-803, 2023, https://doi.org/10.31763/ijrcs.v3i4.1151.

[56] A. J. Abougarair, M. Edardar, “Adaptive Neural Networks Based Robust Output Feedback Control for linearized System,” *WSEAS Transactions on Computer Research*, vol. 9, pp. 125-136, 2021, https://doi.org/10.37394/232018.2021.9.15.

[57] Z. Wang, C. Tang, X. Sima and L. Zhang, “Research on Application of Deep Learning Algorithm in Image Classification,” *2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)*, pp. 1122-1125, 2021, https://doi.org/10.1109/IPEC51340.2021.9421185.

[58] A. Abbas, M. M. Abdelsamea and M. M. Gaber, “4S-DT: Self-Supervised Super Sample Decomposition for Transfer Learning With Application to COVID-19 Detection,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 2798-2808, 2021, https://doi.org/10.1109/TNNLS.2021.3082015.

[59] A. J. Abougarair, M. Ellafi, A. Ma'Arif and O. Salih, “Analysis of Mobile Accelerometer Sensor Movement Using Machine Learning Algorithm,” *2023 IEEE 3rd International Maghreb Meeting of the*

Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA), pp. 46-51, 2023, https://doi.org/10.1109/MI-STA57575.2023.10169214.

[60] A. J. Abougarair, N. A. A. Shashoa, M. K. Aburakhis, “Performance of Anti-Lock Braking Systems Based on Adaptive and Intelligent Control Methodologies,” *Indonesian Journal of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 626-643, 2022, https://doi.org/10.52549/ijeei.v10i3.3794.

[61] M. B. Rozenwald, A. A. Galitsyna, G. V. Sapunov, E. E. Khrameeva, M. S. Gelfand, “A machine learning framework for the prediction of chromatin folding in Drosophila using epigenetic features,” *Peer Journal Computer Science*, vol. 6, p. e307, 2020, https://doi.org/10.7717/peerj-cs.307.

[62] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications,” *Journal of Big Data*, vol. 8, pp.1-74, 2021, https://doi.org/10.1186/s40537-021-00444-8.

[63] F. Zhuang *et al.*, “A Comprehensive Survey on Transfer Learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43-76, 2021, https://doi.org/10.1109/JPROC.2020.3004555.

[64] L. Muflikhah, F. A. Bachtiar, D. E. Ratnawati, R. Darmawan, “Improving Performance for Diabetic Nephropathy Detection Using Adaptive Synthetic Sampling Data in Ensemble Method of Machine Learning Algorithms,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 10, no. 1, pp. 123-137, 2024, https://doi.org/10.26555/jiteki.v10i1.28107.

[65] K. Koirunnisa A. M. Siregar, S. Faisal, “Optimized Machine Learning Performance with Feature Selection for Breast Cancer Disease Classification,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 4, pp. 1131-1143, 2023, http://dx.doi.org/10.26555/jiteki.v9i4.27527.

[66] A. J. Abougarair, M. K. Aburakhis, M. O. Zaroug, “Design and implementation of smart voice assistant and recognizing academic words,” *International Robotics & Automation Journal*, vol. 8, no. 1, pp. 27-32, 2022, https://doi.org/10.15406/iratj.2022.08.00240.

[67] A. J. Abougarair, W. Arebi, “Smart Glove for Sign Language Translation,” *International Robotics & Automation Journal*, vol. 8, no. 3, pp. 109-117, 2022, https://doi.org/10.15406/iratj.2022.08.00253.

[68] H. Isyanto, A. Arifin and M. Suryanegara, “Fast and Accurate Voice Biometrics with Deep Learning Algorithm of CNN Depth wise Separable Convolution Model and Fusion of DWT-MFCC Methods,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 8, no. 3, pp. 431-446, 2022, http://dx.doi.org/10.26555/jiteki.v8i3.24515.

[69] E. Wulczyn *et al.*, “Deep learning-based survival prediction for multiple cancer types using histopathology images,” *PLoS One*, vol. 15, no. 6, p. e0233678, 2020, https://doi.org/10.1371/journal.pone.0233678.

[70] A. J. Abougarair, H. M. Gnan, A. Oun and S. O. Elwarshfani, “Implementation of a Brain-Computer Interface for Robotic Arm Control,” *2021 IEEE 1st International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering MI-STA*, pp. 58-63, 2021, https://doi.org/10.1109/MI-STA52233.2021.9464359.