Indonesian Journal of Data and Science

Volume 6 Issue 2
ISSN 2715-9936
https://doi.org/10.56705/ijodas.v6i2.218

Research Article

Performance Comparison of MobileNet and EfficientNet Architectures in Automatic Classification of Bacterial Colonies

I Putu Alfin Teguh Wahyudi [1], I Gede Iwan Sudipa [2], Luh Gede Bevi Libraeni [3], Made Leo Radhitya[4,*], I Made Dwi Putra Asana[5]

[1] Institut Bisnis dan Teknologi Indonesia (INSTIKI), Denpasar, Indonesia, alfinteguh16@gmail.co
[2] Institut Bisnis dan Teknologi Indonesia (INSTIKI), Denpasar, Indonesia, iwansudipa@instiki.ac.id
[3] Institut Bisnis dan Teknologi Indonesia (INSTIKI), Denpasar, Indonesia, bevi.libraeni@instiki.ac.id
[4] Institut Bisnis dan Teknologi Indonesia (INSTIKI), Denpasar, Indonesia, leo.radhitya@instiki.ac.id
[5] Institut Bisnis dan Teknologi Indonesia (INSTIKI), Denpasar, Indonesia, dwiputraasana@instiki.ac.id

Correspondence should be addressed to Made Leo Radhitya; leo.radhitya@instiki.ac.id

Received 10 February 2025; Accepted 11 May 2025; Published 31 July 2025

Abstract:

Bacterial colony classification is crucial in microbiology but remains labor-intensive and time-consuming when performed manually. Deep learning, particularly Convolutional Neural Networks (CNNs), enables automated classification, improving accuracy and efficiency. This study compares MobileNetV2 and EfficientNet-B0 for bacterial colony classification, evaluating the impact of data augmentation on model performance. Using the Neurosys AGAR dataset, preprocessing techniques such as histogram equalization, gamma correction, and Gaussian blur were applied, while data augmentation (rotation, noise addition, luminosity adjustments) improved model generalization. The dataset was split (80% training, 20% testing), and models were trained with learning rates (0.0001, 0.001) and epochs (100, 150, 200). Results show EfficientNet-B0 outperforms MobileNetV2, achieving higher validation accuracy and stability, with optimal performance at 150–200 epochs and a lower learning rate (0.0001). Data augmentation significantly improved accuracy and reduced overfitting. While MobileNetV2 remains a lightweight alternative, its performance is heavily reliant on augmentation. These findings highlight EfficientNet-B0 as the superior model, supporting the automation of microbiological diagnostics. Future research should explore hybrid CNN architectures, Vision Transformers (ViTs), and real-time implementation for improved classification efficiency.

Keywords: Bacterial Colony Classification; Convolutional Neural Networks (CNNs); MobileNetV2; EfficientNet-B0; Image Classification.
Dataset: https://agar.neurosys.com/

1. Introduction

Bacteria grow on solid surfaces as colonies, which are visible clusters of microorganisms that all stem from a single progenitor cell. As a result, each colony is made up of bacteria that are genetically identical, essentially forming a clone of the original cell. A pure bacterial colony consists of billions of clonal cells derived from a single cell. The initial phase of colony formation, where cells form a single layer and remain individually identifiable, is referred to as a microcolony. Microbial colonies are groups of cells originating from the same organism. Within a developing colony, cells communicate, transfer information to their offspring, and assume roles based on their spatial and temporal distribution [1].

By looking at the morphology and appearance of the formed colonies, the trained specialist may infer presumptive pathogens identification. Traditionally, the identification and classification of bacterial colonies are conducted through visual inspection by a skilled professional, such as a laboratory clinician. Detecting bacterial colonies is labor-intensive and time-consuming [2]. The conventional approach to classification bacterial colonies may lead to inaccurate recognition of bacteria [3]. Advancements in Artificial Intelligence, coupled with deep learning techniques, have made it possible to classify bacterial colonies without the need for manual feature extraction or human intervention. Recently, CNN-based deep learning methods have replaced traditional techniques by automating feature extraction, improving both accuracy and real-time performance. However, detecting small targets remains challenging for CNN models, despite significant advancements in detection algorithms [4]. CNN is a deep learning approach designed to process images by assigning learnable weights and biases to differentiate one input image from another. The input image undergoes repeated convolution with filters or kernels to extract features. CNN architectures consist of several layers, including convolutional layers, activation layers, pooling layers, and fully connected layers [5].

Prominent Convolutional Neural Network (CNN) architectures like MobileNetV2 and EfficientNet-B0 have gained attention for their compact and efficient design. Google introduced MobileNetV2 in 2019, incorporating an improved module that features an inverted residual structure. A key modification was the removal of non-linearity in the narrow layers, allowing the model to enhance efficiency. When used for feature extraction, MobileNetV2 demonstrates strong performance in tasks such as object detection and image segmentation [6]. MobileNetV2 are lightweight models that are significantly smaller and faster in performance compared to many other popular models. MobileNetV2 are a class of compact, low-latency, and low-power models that can be effectively used for tasks such as object detection and classification, as well as other common functions that CNNs excel at. Due to their small size, MobileNets are considered ideal deep learning tools for use on mobile devices [7]. EfficientNet is a deep learning model architecture that incorporates an innovative compound scaling technique. This approach uniformly adjusts the width, depth, and resolution of a Convolutional Neural Network (CNN) using a scaling coefficient, enhancing both performance and efficiency. It is recognized for its ability to combine computational efficiency with high accuracy. In tests using the base model, EfficientNet-B0, on the ImageNet dataset, it achieved a mean Top-1 Accuracy of 77.1% and a mean Top-4 Accuracy of 93.3%. This demonstrates its strong performance, particularly in terms of accuracy, while still maintaining efficient use of resources [8]. Additionally, the architecture includes a mobile-friendly baseline model, EfficientNet-B0, which is optimized with MBConv blocks and Squeeze-and-Excitation modules [9].

Since both models share similarities in terms of performance and efficiency, this study aims to compare their accuracy, loss effectiveness in the classification of bacterial colonies, specifically focusing on species such as Staphylococcus Aureus, Escherichia Coli, and Bacillus Subtilis, to determine which model yields better results.

This study focuses on bacterial colony classification using only three bacterial species and does not include environmental factors that may influence colony growth. Additionally, the dataset used is limited to publicly available bacterial colony images, which may impact model generalization.

The contributions of this research include providing insights into the comparative performance of MobileNetV2 and EfficientNet-B0 in bacterial colony classification. The findings will assist in selecting efficient deep learning models for microbiological applications, thereby enhancing laboratory efficiency and diagnostic accuracy.

The remainder of this paper is organized as follows: Section 2 details the research methodology, including dataset selection and preprocessing techniques. Section 3 presents experimental results and discussions, followed by Section 4, which concludes the study and suggests directions for future research.

2. Method:

This section details the research methodology for bacterial colony classification utilizing Convolutional Neural Network (CNN) models, specifically MobileNet V2 and EfficientNet-B0 architectures. The process begins with the collection of bacterial colony image datasets, which undergo preprocessing and data augmentation to improve data quality and variability, aiming to enhance the models' generalization capabilities. The dataset is then partitioned, with 80% allocated for training and the remaining 20% reserved for testing. Following this, model training is conducted using both MobileNet V2 and EfficientNet-B0 architectures to classify bacterial colonies. Finally, the performance of each model is evaluated, assessing the accuracy and efficiency of MobileNet V2 and EfficientNet-B0 in executing the classification tasks.

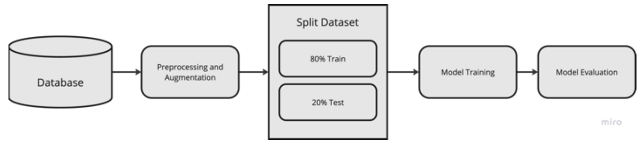

Figure 1. Model Building Flow

CNN for Classification

Convolutional neural networks (CNNs) are a particular type of artificial neural network designed to process grid-shaped data, such as images. CNNs consist of several layers, including a convolution layer, an activation layer, and a pooling layer. The convolution layer works to extract features by applying filters to the input, resulting in a feature map that highlights the important characteristics of the data. The pooling layer then reduces the dimensionality of the feature map, retaining essential information while reducing computational complexity. The combination of these layers enables CNNs to automatically learn a hierarchical representation of the data, a process that is particularly effective for tasks such as image classification and object detection.

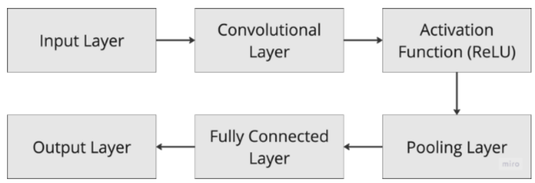

Figure 2. Convolutional Neural Networks (CNN)

MOBILENETV2

MobileNetV2 is a CNN architecture optimized for resource-constrained devices such as mobile phones. It introduces inverted residual blocks and linear congestion. In inverted residual blocks, instead of reducing the number of channels as in traditional residual blocks, MobileNetV2 expands the dimensions in the initial layer, then performs depth convolution, and finally projects back to smaller dimensions. This approach preserves important information during the transformation process, improving the efficiency and accuracy of the model. In addition, the use of a linear bottleneck with no nonlinear activation function in the output layer helps prevent information loss that can occur due to nonlinearity in a low-dimensional space [10].

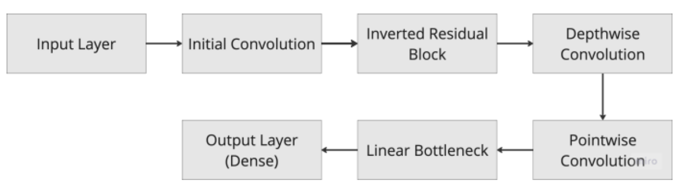

Figure 3. MobileNetV2 Architecture

EFFICIENTNET-B0

EfficientNet-B0 is the base model of the EfficientNet family, designed to achieve an optimal balance between accuracy and computational efficiency. It builds on the Mobile Inverted Bottleneck Convolution (MBConv) block used in MobileNetV2, with the addition of a squeeze-and-excitation (SE) module that provides an attention mechanism to adjust the importance of each feature. EfficientNet's uniqueness lies in its compound scaling method, which simultaneously increases the depth, width, and resolution of the network in a balanced manner, allowing the model to efficiently scale as needed without sacrificing performance [11].

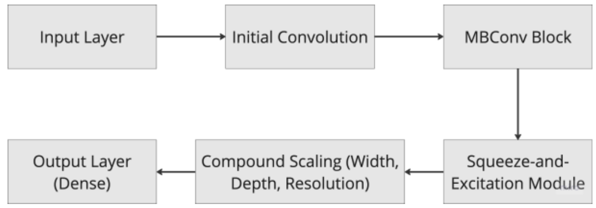

Figure 4. EfficientNet-B0 Architecture

Dataset Description

This study utilizes a dataset obtained from the Neurosys website (https://agar.neurosys.com/) under the title AGAR: a microbial colony detection SDK and a dataset for accurate deep learning bacteria detection. The data from the website serves as the primary data for this research, and the total of data in the datasets is 10 images and 10 JSON file, where the JSON data includes the X and Y coordinates of bacterial colonies, which are used for cropping the images in the dataset.

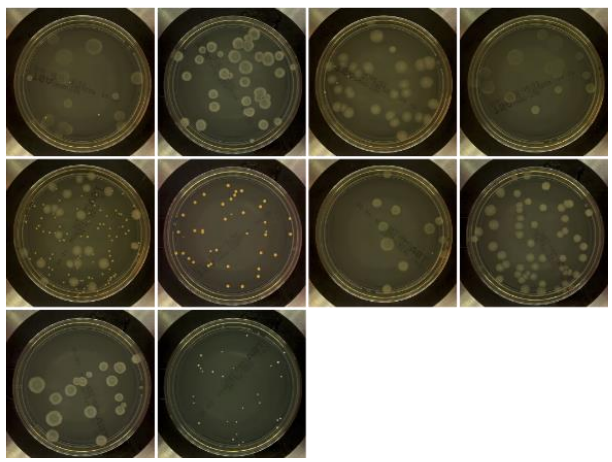

Figure 5. Images Datasets from Neurosys Datasets

Cropping Image

Images from the dataset are fetched and cropped based on a JSON file containing the coordinates for cropping the image, then the results are saved in a folder according to the class label, this process opens and reads the JSON file, then loads the images from the image path using OpenCV. After that, it checks whether a folder for each class already exists in the output folder path, and if not, a folder is created. For each entry in the JSON, it retrieves the image ID, class label, and size and position information (height, width, x, y) which is used to crop the relevant part of the image according to those coordinates. These image pieces are stored in a class folder with a name that includes the class label and image ID.

Pre-processing and Augmentation

The preprocessing section encompasses techniques aimed at enhancing image quality before entering the modeling phase, thereby maximizing data effectiveness for machine learning models. Preprocessing techniques include histogram equalization, which enhances contrast by evenly distributing pixel intensity, gamma correction, which adjusts image brightness to correct overly dark or bright appearances, and Gaussian blur, which smooths the image using a Gaussian kernel, reducing fine details or noise.

The augmentation section, on the other hand, serves to increase dataset diversity by artificially modifying images to enhance model resilience against varying conditions. Applied augmentation techniques include increasing and decreasing luminosity, which adjust brightness levels, as well as increasing and decreasing contrast, which alter contrast levels to introduce variation in lighting conditions. The rotate clockwise and rotate counter-clockwise functions provide image rotations to add angle variety, while add noise inserts random Gaussian noise to build resilience to visual interference. Lastly, sharpen image enhances image sharpness, aiding the model in recognizing essential details in objects. Preprocessing focuses on optimizing input quality, while augmentation enriches the dataset by providing variations, enabling the model to learn from more diverse image conditions during training.

Split Dataset

To split the image dataset into training and validation subsets using TensorFlow's ImageDataGenerator and apply preprocessing to enhance input data quality, the following steps are taken. First, the precision of numerical display in the pandas library is set to two decimal places to improve accuracy in numerical data analysis. The image resolution is specified as 224x224 pixels, and the dataset location is established in the training directory. During preprocessing, the code normalizes pixel values using the parameter rescale 1./255, adjusting the pixel scale from the 0-255 range to 0-1. Additionally, the parameter validation split 0.20 is employed to allocate 20% of the data as a validation subset, enabling the model to be evaluated with data excluded from training.

Data generators for the training and validation subsets are then initialized using ImageDataGenerator based on the specified parameters. This configuration enables both generators to randomly draw data from the same dataset directory, resizing images to the predefined target resolution. To verify that the data structure aligns with the model's requirements, the code retrieves a data batch from the training subset and prints its shape to display the dimensions of images and labels. Following this, the class indices are converted into an array of labels ordered by numerical index, with class names capitalized to facilitate identification. This code provides a comprehensive pipeline for dataset splitting and preprocessing, preparing the image data in training and validation subsets ready for machine learning model training.

Model Training

The model chosen for EfficientNet is EfficientNetB0, the smallest variant within the EfficientNet family, due to its efficient balance between computational resource usage and performance. This selection supports the research objective of evaluating model performance under time constraints, as larger EfficientNet models would require significantly more time to train. Additionally, MobileNetV2 was selected for its lightweight architecture, making it ideal for environments with limited resources while maintaining adequate accuracy. These two models fulfill the study's aim to assess model efficiency without compromising on training time or computational feasibility.

Model Evaluation

The comparative analysis of the two models, EfficientNetB0 and MobileNetV2, will focus on five key performance metrics: training accuracy, testing accuracy, training loss, validation accuracy, validation loss, and confusion matrix. These criteria will be assessed to evaluate each model's overall effectiveness and stability. The training and validation accuracies will provide insight into how well each model learns from the data and generalizes to unseen data. Loss metrics, including training and validation loss, will indicate the models' optimization efficiency and convergence. Additionally, comparing testing accuracy against training accuracy will help identify any potential overfitting, with higher variance suggesting overfitting and closer values indicating a better generalization. This comprehensive evaluation aims to determine which model best balances accuracy and efficiency for bacterial colony classification.

Confusion Matrix

The confusion matrix is a critical tool for evaluating the performance of classification models. It provides a detailed breakdown of a model's performance by comparing its predicted results with the actual class labels. The matrix is typically presented in a tabular format, with the rows representing the actual classes and the columns representing the predicted classes. For binary classification, the confusion matrix has four key components: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). These metrics allow practitioners to gain a deeper understanding of where the model succeeds and fails, making it an essential resource for diagnosing classification errors [12].

3. Results and Discussion

Results

Data preprocessing and augmentation are crucial steps in ensuring the quality and variability of the dataset before training the model. In this study, various preprocessing techniques were applied to enhance the quality of bacterial colony images, including histogram equalization, gamma correction, Gaussian blur, sharpening, contrast enhancement and reduction, as well as luminosity adjustment. These techniques aim to normalize lighting conditions, enhance feature clarity, and reduce noise, allowing the model to capture more representative patterns.

In addition to preprocessing, data augmentation was performed to improve the model's generalization capability against environmental variations. The augmentation techniques employed include clockwise and counter clockwise rotation, Gaussian noise addition, and luminosity adjustment. These transformations simulate real-world conditions such as different lighting angles, brightness levels, and visual noise interference.

Figures 6 illustrate the results of various preprocessing and augmentation techniques applied in this study. These methods are expected to assist the Convolutional Neural Network (CNN) in better recognizing bacterial colony features, improving prediction accuracy, and reducing the risk of overfitting. At the end the total of data for training is 2053 data and 515 data testing.

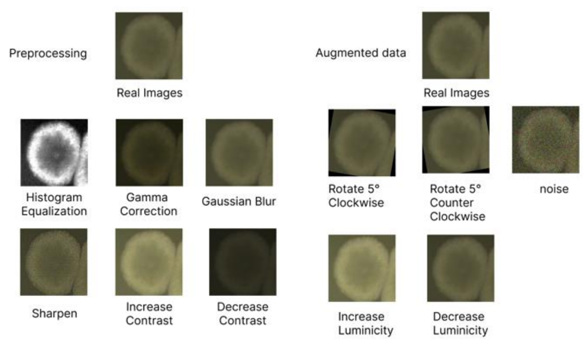

Figure 6. Preprocessing and Augmented Data

Table 1. Training Results

Model	Schema	*Learning Rate*	*Epoch*	*Accuracy*	*Loss*	Validation Accuracy	*Validation Loss*
Pretrained .MobileNet	1	0,001	100	0.9991	0.0039	0.3266	3.8578
	2	0,001	150	1.0000	0.0016	0.8661	0.9040
	3	0,001	200	1.0000	0.000073	0.9953	0.9953
	4	0,0001	100	1.0000	0.0014	0.4186	50.8993
	5	0,0001	150	1.0000	0.0009622	0.9639	0.1937
	6	0,0001	200	1.0000	0.0011	0.9707	0. 0.0653
Pretrained MobileNet + *Augemented Data*	1	0,001	100	0.99915	0.00812	0.9780	0.09063
	2	0,001	150	1.0000	0.000018264	1.0000	0.00062683
	3	0,001	200	0.99986	0.00016	1.0000	0.000104505
	4	0,0001	100	1.0000	0.00048	1.0000	0.00016
	5	0,0001	150	0.99984	0.00066	1.0000	0.0000715
	6	0,0001	200	0.9995	0.00354	1.0000	0.00014
Pretrained EfficientNet	1	0,001	100	0.993	0.021	0.372	1.695
	2	0,001	150	1.0000	0.0023	0.9429	0.2831
	3	0,001	200	1.0000	0.0000479	1.0000	0.0179

Model	**Schema**	***Learning Rate***	***Epoch***	***Accuracy***	***Loss***	**Validation Accuracy**	***Validation Loss***
	4	0,0001	100	1.0000	0.0023	0.4346	71.0605
	5	0,0001	150	1.0000	0.000840	0.9642	0.1621
	6	0,0001	200	1.0000	0.00344	1.0000	0.00678
Pretrained EfficienetNet + *Augemented Data*	1	0,001	100	0.9989	0.00687	0.9998	0.0010
	2	0,001	150	0.9936	0.0062	0.8895	2.8702
	3	0,001	200	1.0000	0.00013007	1.0000	0.00004837
	4	0,0001	100	0.99977	0.000829	0.99982	0.000684
	5	0,0001	150	0.99981	0.0004239	0.99981	0.0004519
	6	0,0001	200	0.9991	0.0672	0.9226	0.2503

The results of model training and evaluation are presented in **Table 1**, which compares the accuracy, loss, validation accuracy, and validation loss across different configurations of MobileNet and EfficientNet, both with and without data augmentation. The findings indicate that EfficientNet with augmented data consistently achieved the highest validation accuracy, nearing 1.000 at 150–200 epochs, demonstrating its superior generalization capability. Similarly, MobileNet with augmented data also exhibited significant improvements in validation accuracy compared to its non-augmented counterpart, emphasizing the effectiveness of data augmentation in reducing overfitting.

Furthermore, the analysis shows that a lower learning rate (0.0001) resulted in lower validation loss, particularly after 150–200 epochs, suggesting that a more gradual optimization process contributes to model stability. MobileNet, despite achieving high accuracy, displayed higher variance in validation accuracy when augmentation was not applied, indicating potential overfitting when trained with limited data diversity. EfficientNet, on the other hand, demonstrated more stable performance, benefiting from its compound scaling strategy that balances network depth, width, and resolution.

These results suggest that data augmentation plays a crucial role in improving the robustness and reliability of CNN-based bacterial colony classification models. The combination of EfficientNet with augmented data and an increased number of epochs (150–200) is recommended for optimal performance, ensuring both high accuracy and generalization capability. The findings also highlight that while MobileNet offers a lighter and computationally efficient alternative, EfficientNet remains the superior choice for applications where accuracy is the primary concern. The experimental outcomes, as visualized in **Table 1**, support the hypothesis that model architecture, training configurations, and data augmentation significantly impact the classification performance.

The results demonstrate that EfficientNet outperforms MobileNet in terms of classification accuracy and stability, particularly when data augmentation is applied. This suggests that EfficientNet's compound scaling method enables it to capture intricate patterns in bacterial colony images more effectively than MobileNet. The improved validation accuracy after augmentation indicates that enhancing dataset variability significantly boosts model generalization, preventing overfitting and making the model more robust to unseen data.

Additionally, the experiments confirm that increasing the number of epochs (from 100 to 200) leads to improved accuracy and lower validation loss, signifying better convergence of the model. The application of a lower learning rate (0.0001) also enhances stability, reducing the risk of erratic loss fluctuations. MobileNet, while achieving high accuracy, exhibited inconsistencies in validation accuracy, particularly in scenarios without augmentation, reinforcing the need for extensive data preprocessing and augmentation in CNN-based bacterial classification tasks.

EfficientNet consistently outperforms MobileNet in terms of validation accuracy and generalization, making it the preferred model for bacterial colony classification. The application of data augmentation significantly enhances model performance by reducing overfitting and improving the model's ability to generalize to unseen bacterial colony images. Additionally, a lower learning rate of 0.0001 contributes to better model stability, facilitating a smoother convergence process and resulting in lower validation loss. Training the models for 150–200 epochs has been found to yield optimal results, as increasing the training duration improves accuracy while maintaining a low validation loss.

Despite these findings, MobileNet remains a viable alternative for environments with computational constraints; however, its performance is highly dependent on the incorporation of data augmentation to achieve results comparable to EfficientNet.

The analysis shows that Pretrained EfficientNet with Augmented Data achieves the highest classification accuracy, particularly in schemas 2, 3, 4, 5, and 6, where all predictions are correct with zero misclassifications. In contrast, Pretrained MobileNet without augmentation exhibits more misclassifications, highlighting its limitations. Data augmentation significantly improves model performance, reducing errors and enhancing generalization. Among all models, EfficientNet with augmented data consistently outperforms others, achieving perfect classification due to it's compound scaling method and optimized training (150-200 epochs, 0.0001 learning rate). These findings confirm that EfficientNet with augmented data is the most reliable model for bacterial colony classification, emphasizing the importance of data augmentation and deep learning advancements in microbiological diagnostics.

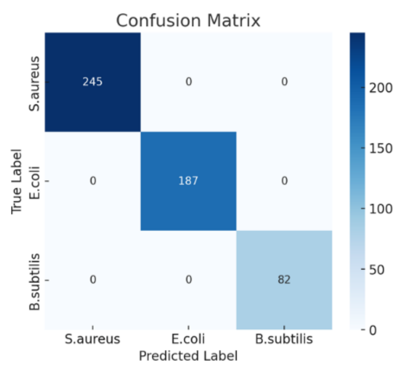

Figure 7. Result of Confusion Matrix on Schema 2, 3, 4, 5, and 6 in EfficientNet Models With Augmented Data

Discussion

The comparative analysis reveals that EfficientNet consistently outperforms MobileNet in terms of validation accuracy and generalization capabilities for bacterial colony classification tasks. This superior performance can be attributed to EfficientNet's compound scaling method, which balances network depth, width, and resolution, enabling it to capture intricate patterns more effectively. The application of data augmentation techniques further enhances model performance by increasing dataset diversity, thereby reducing overfitting and improving the model's ability to generalize to unseen data. Additionally, employing a lower learning rate of 0.0001 contributes to better model stability, facilitating a smoother convergence process and resulting in lower validation loss. Training the models for 150–200 epochs yields optimal results, as extending the training duration allows the models to learn more complex features without significant risk of overfitting. While MobileNet remains a viable alternative for environments with computational constraints, its performance is highly dependent on the incorporation of data augmentation to achieve results comparable to EfficientNet.

The findings of this study align with existing literature on image processing methods. For instance, a comparative study by Boudouh et al. (2024) [13] demonstrated that data augmentation significantly enhances the performance of deep learning models by mitigating the limitations imposed by limited training data and improving model accuracy. Furthermore, the superior performance of EfficientNet observed in this study corroborates the results reported by Kumar et al. (2024) [14], who found that EfficientNet-based models achieve higher classification accuracy due to their efficient feature extraction capabilities. These consistencies reinforce the validity of the current study's outcomes and highlight the effectiveness of EfficientNet and data augmentation in image classification tasks.

The demonstrated superiority of EfficientNet, particularly when combined with data augmentation, provides a robust framework for developing automated systems for bacterial colony classification. Implementing such systems can significantly reduce the time and labor associated with manual classification processes in microbiological laboratories, leading to increased efficiency and accuracy in pathogen identification. Moreover, the findings suggest that even in resource-constrained settings, where computational power may be limited, employing data augmentation techniques with lightweight models like MobileNet can still yield satisfactory performance, making advanced classification tools more accessible.

Despite the promising results, this study has certain limitations. The experiments were conducted on a specific dataset, which may not encompass the full diversity of bacterial colony morphologies encountered in different laboratory settings. Additionally, the study focused solely on MobileNet and EfficientNet architectures; other contemporary models were not evaluated, which could provide a more comprehensive understanding of the relative performance across different neural network designs. Furthermore, the computational resources available influenced the choice of model parameters, such as the number of epochs and learning rates, which may affect the generalizability of the findings.

Future research should aim to validate these findings across diverse and larger datasets to ensure the robustness and generalizability of the models. Exploring the performance of other state-of-the-art architectures, such as Vision Transformers or hybrid models, could provide deeper insights into the most effective designs for bacterial colony classification. Additionally, investigating the impact of different data augmentation strategies and hyperparameter tuning methods could further enhance model performance. Finally, developing real-time classification systems and evaluating their performance in practical laboratory environments would be a valuable step toward translating these research findings into operational tools.

4. Conclusion

This study compared the performance of MobileNetV2 and EfficientNet-B0 for bacterial colony classification, demonstrating that EfficientNet-B0 consistently outperforms MobileNetV2 in validation accuracy and generalization ability. The compound scaling approach of EfficientNet-B0 enables it to capture more intricate features, making it a superior choice for bacterial image classification. Meanwhile, MobileNetV2, while computationally efficient, exhibited lower accuracy without augmentation. The application of data augmentation significantly improved model robustness, reducing overfitting and enhancing generalization. Additionally, lower learning rates (0.0001) and extended training epochs (150–200) contributed to better model stability and performance. The research confirmed that EfficientNet-B0 is the superior model, validating the hypothesis that data augmentation enhances model generalization, and that hyperparameter tuning, particularly adjusting learning rates and training durations, optimizes CNN-based classification models. This study contributes to deep learning-based bacterial classification by empirically evaluating MobileNetV2 and EfficientNet-B0, demonstrating the role of data augmentation, and providing insights into hyperparameter tuning for model optimization. The findings support the automation of microbiological diagnostics, improving efficiency and accuracy in bacterial identification.

Future research should expand dataset diversity, optimize EfficientNet-B0 for real-time applications, explore hybrid CNN models and Vision Transformers (ViTs), implement transfer learning from biomedical datasets, and validate models in real-world laboratory environments to assess their practical effectiveness. This study establishes a foundation for AI-driven bacterial colony classification, offering valuable insights for advancing microbiological diagnostics and automated identification systems.

References:

[1] L. Z. Pipe, “Understanding Evolving Bacterial Colonies,” in *Springer Handbook of Bio-/Neuroinformatics*, N. Kasabov, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 93–112. doi: 10.1007/978-3-642-30574-0_7.

[2] S. Patel, “Bacterial Colony Classification Using Atrous Convolution with Transfer Learning Authors,” *Ann Rom Soc Cell Biol*, pp. 1428–1441, Oct. 2021.

[3] T. Beznik, P. Smyth, G. de Lannoy, and J. A. Lee, “Deep learning to detect bacterial colonies for the production of vaccines,” *Neurocomputing*, vol. 470, pp. 427–431, 2022, doi: 10.1016/j.neucom.2021.04.130.

[4] Liu Shousheng *et al.*, “Bacterial colonies detecting and counting based on enhanced CNN detection method,” *E3S Web Conf.*, vol. 233, p. 2012, 2021, doi: 10.1051/e3sconf/202123302012.

[5] S. G. Sambandam, R. Purushothaman, R. U. Baig, S. Javed, V. T. Hoang, and K. Tran-Trung, “Intelligent surface defect detection for submersible pump impeller using MobileNet V2 architecture,” *The International Journal of Advanced Manufacturing Technology*, vol. 124, no. 10, pp. 3519–3532, 2023, doi: 10.1007/s00170-022-10386-x.

[6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “ MobileNetV2: Inverted Residuals and Linear Bottlenecks ,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* , Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[7] L. Mpova, T. Shongwe, and A. Hasan, “Classification and Detection of Cyanosis Images on Lightly and Darkly Pigmented Individual Human Skins using a Fine-Tuned MobileNet Architecture,” in *2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, 2023, pp. 1–5. doi: 10.1109/icABCD59051.2023.10220464.

[8] H. Pu and K. Yi, “A Comparative Analysis of EfficientNet and MobileNet Models’ Performance on Limited Datasets: An Example of American Sign Language Alphabet Detection,” *Highlights in Science, Engineering and Technology*, 2024.

[9] A. Matthew, A. A. S. Gunawan, and F. I. Kurniadi, “Diabetic Retinopathy Diagnosis System Based on Retinal Biomarkers Using EfficientNet-B0 for Android Devices,” in *2023 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, 2023, pp. 207–212. doi: 10.1109/COMNETSAT59769.2023.10420736.

[10] V.-D. and J. K.-H. Hoang Van-Thanh and Hoang, “Rethinking Mobile Inverted Bottleneck Convolution for EfficientNet,” in *Computational Intelligence Methods for Green Technology and Sustainable Development*, W.-J. and Q. H. A. and L. H.-G. and Q. H.-N. Huang Yo-Ping and Wang, Ed., Cham: Springer International Publishing, 2023, pp. 435–445.

[11] T. M. and J. K.-H. Thanh Hoang Van and Phuong, “A Compact Version of EfficientNet,” in *The 12th Conference on Information Technology and Its Applications*, H. and H. C.-P. and N. Q.-V. Nguyen Ngoc Thanh and Le-Minh, Ed., Cham: Springer Nature Switzerland, 2023, pp. 297–305.

[12] K. M. Ting, “Confusion Matrix,” in *Encyclopedia of Machine Learning and Data Mining*, G. I. Sammut Claude and Webb, Ed., Boston, MA: Springer US, 2017, p. 260. doi: 10.1007/978-1-4899-7687-1_50.

[13] T. Renchin, D. Janchiv, and B. Sanjaa, “Comparative study of some methods for image processing,” in *2008 Third International Forum on Strategic Technologies*, 2008, pp. 364–365. doi: 10.1109/IFOST.2008.4602858.

[14] A. A. Kumar, N. Lal, and R. N. Kumar, “A Comparative Study of Various Filtering Techniques,” in *2021 5th International Conference on Trends in Electronics and Informatics (ICOEI)*, 2021, pp. 26–31. doi: 10.1109/ICOEI51242.2021.9453068.