Vol. 6 No. 1, 2025, pp. 81-91 **Jurnal Pepadun**
©2025 Computer Science Unila Publishing Network all rights reserved
DOI: 10.23960/pepadun.v6i1.267

A Comparative Study of CNN Architectures: ConvNeXt, MobileNetV3, and EfficientNet for Oral Disease Diagnosis

[1]Ferli Malkan Amien, [2]Didik Kurniawan, [3]Akmal Junaidi and [4]Bambang Hermanto

[1,2,3,4]Department of Computer Science, University of Lampung,
Jl. Prof. Dr. Ir. Sumantri Brojonegoro No. 1, Bandar Lampung, Lampung Province, Indonesia, 35145
e-mail: [*1]**ferli.malkan21@students.unila.ac.id**, [2]didik.kurniawan@fmipa.unila.ac.id,
[3]akmal.junaidi@fmipa.unila.ac.id, [4]bambang.hermanto@fmipa.unila.ac.id

Abstract - *Oral diseases are a global health issue affecting billions of people worldwide. Early detection using artificial intelligence technology, particularly Convolutional Neural Networks (CNN), can enhance the accuracy of oral disease diagnosis. This study compares the performance of three CNN architectures, namely ConvNeXt Tiny, MobileNetV3 Large, and EfficientNet B0, in classifying oral diseases based on a medical image dataset. The evaluation considers accuracy, computational efficiency, and processing time. The results show that EfficientNet B0 achieved the highest accuracy (98.21%) with fewer parameters than ConvNeXt Tiny. However, ConvNeXt Tiny demonstrated the best absolute accuracy (96.96%) despite higher computational resource consumption. MobileNetV3 Large exhibited high efficiency in parameter usage with competitive accuracy (96.44%). Thus, the choice of CNN architecture depends on the trade-off between high accuracy and computational efficiency in clinical implementation.*

Keywords: *Convolutional Neural Network (CNN); ConvNeXt; MobileNetV3; EfficientNet; Oral Disease.*

1. INTRODUCTION

Oral diseases represent a global health concern, affecting more than 3.5 billion people worldwide as of 2019. These conditions include dental caries, periodontal disease, tooth loss, oral cavity cancer, dental trauma, noma, and congenital anomalies such as cleft palate and cleft lip [1][2]. According to the WHO's Global Oral Health Status Report: Towards Universal Health Coverage for Oral Health by 2030, oral diseases, especially dental caries and periodontal disease, are highly prevalent in low- and middle-income countries due to limited access to dental care services. These diseases significantly impact individual quality of life, causing difficulties in speaking and chewing, increasing the risk of infection, and potentially progressing to more severe conditions such as oral cancer. Furthermore, oral diseases share common risk factors with other non-communicable diseases, such as poor dietary habits, tobacco use, and alcohol consumption. This intersection emphasizes the need for integrated healthcare strategies to address oral and general health concurrently [2].

In recent years, the field of medical imaging has leveraged the rapid progress in machine learning to enhance healthcare delivery, resulting in substantial advancements in diagnostic capabilities. Ongoing improvements in deep learning architectures have markedly boosted both the accuracy of diagnoses and the efficiency of data processing [3][4][5]. These technological advancements are particularly relevant in the context of oral health, where the substantial burden of oral diseases and the restricted availability of dental services underscore the urgent need for advanced and accessible diagnostic solutions.

The substantial burden of oral diseases and the restricted availability of dental services underscore the urgent need for advanced and accessible solutions. Traditional healthcare approaches often fall short in addressing these challenges effectively, particularly in under-resourced settings. In response, the rapid development of artificial intelligence (AI) and deep learning presents promising opportunities to transform medical diagnostics. These technologies enable efficient and accurate analysis of medical data, which in turn supports early detection, improves treatment planning, and enhances healthcare outcomes at scale [6]. In addition to accelerating the diagnostic process, AI-based systems can generate data-driven insights that improve clinical decision-making and support personalized healthcare services, contributing to more targeted and effective treatments [7].

One of the most prominent advancements in AI-driven diagnostics is the application of deep learning techniques, particularly convolutional neural networks (CNNs). CNNs have become a cornerstone in medical image analysis due to their ability to automatically extract and learn relevant features from visual data, making them highly effective for pattern recognition and image classification tasks [8]. These models offer high accuracy and have been successfully deployed in various medical domains, including tumor detection, retinal image analysis, and skin lesion classification, demonstrating their adaptability and diagnostic value [9].

Despite the growing adoption of CNNs in medical applications, selecting an architecture that balances classification accuracy with computational efficiency remains a major challenge. This is especially crucial for real-time healthcare systems or mobile diagnostic tools, where computational resources may be limited. Therefore, this study aims to evaluate and compare the performance of three well-known CNN architectures, namely ConvNeXt, MobileNetV3, and EfficientNet, that are recognized for their efficiency in image classification tasks. Each of these architectures applies unique optimization strategies to reduce computational cost while preserving or improving classification performance.

ConvNeXt incorporates modern techniques such as depth-wise convolution and enhanced layer normalization to refine the convolutional process and improve learning efficiency [10]. MobileNetV3 enhances performance through the use of inverted residual blocks and squeeze-and-excitation modules, offering a lightweight solution that maintains competitive accuracy [11]. EfficientNet, on the other hand, utilizes compound scaling to balance depth, width, and resolution in a way that maximizes model performance while minimizing resource consumption [12].

The dataset utilized in this study originates from the work of Piyarathne et al. [13], published in the article titled “A Comprehensive Dataset of Annotated Oral Cavity Images for Diagnosis of Oral Cancer and Oral Potentially Malignant Disorders.” This publicly available dataset, hosted on the Zenodo repository, comprises diverse and high-quality oral cavity images, thereby enabling the evaluated models to generalize well across various clinical scenarios. This research focuses on assessing the classification accuracy and computational efficiency of the ConvNeXt, MobileNetV3, and EfficientNet architectures in diagnosing oral diseases through image data. The goal is to identify the most effective and practical deep learning solution for real-world clinical applications, particularly in resource-constrained environments that demand both speed and reliability in the diagnostic process.

2. RESEARCH METHODS

This study adopts the OSEMN framework, introduced by Hilary Mason and Chris Wiggins in their article titled “A Taxonomy of Data Science,” as the guideline for the research workflow. The OSEMN framework comprises five sequential stages: Obtain (data collection), Scrub (data cleaning), Explore (data exploration), Model (modeling), and Interpret (result interpretation) [14]. These stages guide the end-to-end process of building and evaluating the machine learning pipeline. The workflow is visualized in Figure 1.

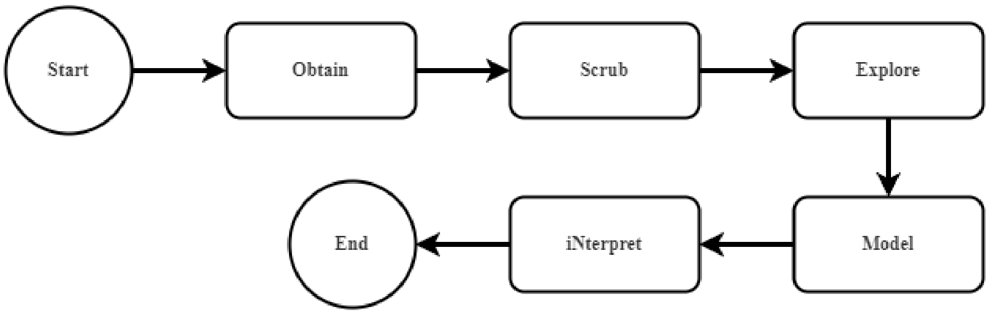

Figure 1. OSEMN framework.

2.1. Obtain

The dataset used in this study contains 3,000 oral cavity images divided into four categories: Healthy, Benign, Oral Potentially Malignant Disorders (OPMD), and Oral Cavity Cancer (OCA). This dataset was obtained from the research conducted by Piyarathne et al. [13]. Each image is labeled with its corresponding category, allowing for supervised learning. Figure 2 displays sample images from each class, while Figure 3 presents the class distribution, which is notably imbalanced.

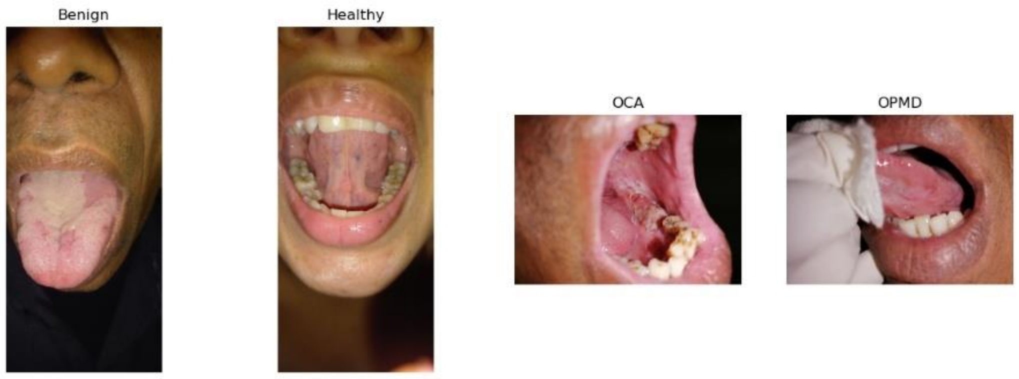

Figure 2. Sample images from each class.

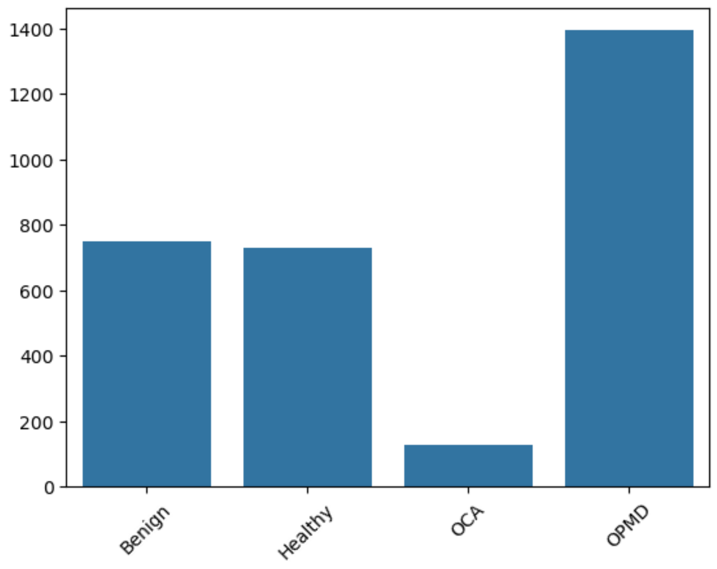

Figure 3. Class distribution in the dataset.

2.2. Scrub

2.2.1. Data Augmentation

Data augmentation is a technique that utilizes various transformations to generate synthetic data from an existing dataset. This technique plays a crucial role in training deep neural networks (DNNs), as it allows the model to receive a wider range of data variations without the need to collect new data [15]. Moreover, data augmentation is part of regularization methods aimed at improving the model's generalization ability [16]. In the context of this study, data augmentation is used to address the imbalance in the number of images across categories and to reduce the risk of overfitting. This is due to the highly uneven distribution of the initial dataset, particularly in the OCA category, which contains significantly fewer samples compared to other categories. The augmentation process was conducted in two stages:

a. First Stage
This stage focused on increasing the number of images in the OCA category from 129 to 500. The augmentation techniques used in this stage are presented in Table 1.

Table 1. Augmentation techniques used in the first stage.

No	Augmentation Type	Value
1	Random Resized Crop	224, scale = (0.8, 1.0)
2	Random Horizontal Flip	
3	Random Vertical Flip	
4	ToTensor	

b. Second Stage
After increasing the OCA category to 500 images, the second stage aimed to equalize the number of images in all categories (Healthy, Benign, and OCA) to match the largest category, OPMD, which contained 1,394 images. The augmentation techniques used in this stage are listed in Table 2. The results of augmentation and distribution balancing are summarized in Table 3.

Table 2. Augmentation techniques used in the second stage.

No	Augmentation Type	Value
1	Random Resized Crop	224, scale = (0.8, 1.0)
2	Random Horizontal Flip	
3	Random Vertical Flip	
4	Random Rotation	10
5	Color Jitter	brightness = 0.2, contrast = 0.2, saturation = 0.2, hue = 0.1
6	Random Affine	degrees =15, translate = (0.1, 0.1)
7	ToTensor	

Table 3. Data distribution per category before and after augmentation.

Class	Original	After Stage 1	After Stage 2
Healthy	729	729	1394
Benign	748	748	1394
OPMD	1394	1394	1394
OCA	129	500	1394
Total	3000	3371	5576

2.2.2. Dataset Splitting

To prepare for training and evaluation, the augmented dataset was split into training, validation, and test subsets. The split was done randomly using the train_test_split function from the scikit-learn library to avoid overlapping samples. Three different scenarios were tested with varying proportions, as presented in Table 4.

Table 4. Dataset splitting scenarios

Scenario	Train (%)	Test (%)	Validation (%)	Train	Test	Val
A	80	10	10	4460	560	556
B	70	15	15	3900	840	836
C	60	20	20	3344	1116	1116

2.2.3. Data Normalization

The final step in the scrubbing stage is normalization, which aims to standardize the pixel value distribution across all images. This process was carried out using the transforms. Normalize method, which adjusts pixel values based on the mean and standard deviation of the dataset. Normalization was applied to each color channel (R, G, B), resulting in pixel values with a mean close to zero and a standard deviation close to one for every image. The normalization parameters used are listed in Table 5.

Table 5. Data normalization parameters.

Parameter	Value
Mean	0.485, 0.456, 0.406
Standard Deviation	0.229, 0.224, 0.225

2.3. Explore

2.3.1. Data Distribution per Class

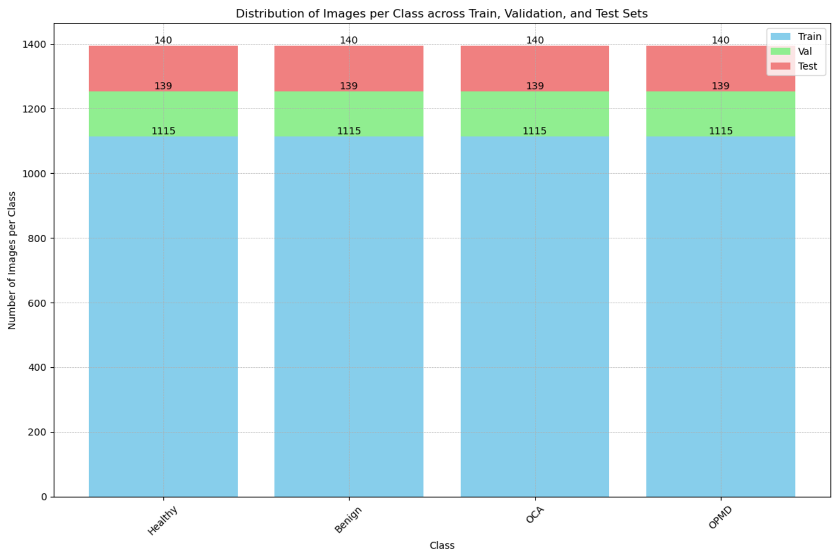

Figure 4. Data distribution chart after the Scrub process.

The class distribution was evaluated to ensure a balanced number of images across the following categories: Healthy, Benign, Oral Potentially Malignant Disorders (OPMD), and Oral Cancer (OCA). The results show that each of the four categories contains exactly 1,394 images, as illustrated in Figure 4. This balanced distribution is essential to avoid bias during training and to enhance the generalization capability of the model. A bar plot visualization reinforces this outcome, with all four bars appearing at equal height.

2.3.2. Sample Image Visualization

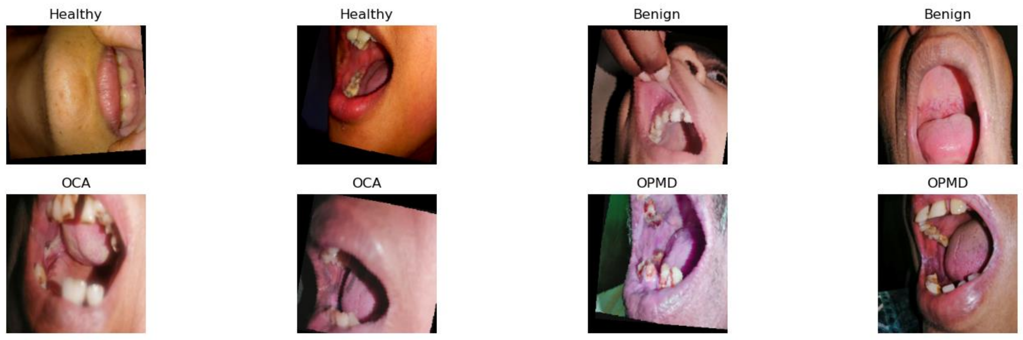

Figure 5. Sample images from each disease category after the Scrub stage.

To understand the visual differences between classes, sample images from each category were visualized after the preprocessing phase, as shown in Figure 5. Each class displays distinct visual characteristics: Healthy images show normal tissues, Benign images reveal mild abnormalities, OPMD images include lesions or discoloration, and OCA images exhibit severe tissue damage or ulcers. This visualization confirms the diversity and quality of the dataset post-augmentation.

2.3.3. Model Architecture Exploration

This study employs three CNN architectures: ConvNeXt, MobileNetV3, and EfficientNet, each with the following strengths:

a. ConvNeXt: Integrates advantages of Vision Transformers (ViT) with traditional CNNs. It uses large kernel convolutions (7×7), LayerNorm, GELU, and AdamW to strike a balance between high performance and computational efficiency.

b. MobileNetV3: Optimized for mobile devices through Neural Architecture Search (NAS), incorporating components such as inverted residual blocks, Squeeze-and-Excitation, and hard-swish activation. It is ideal for real-time applications and edge computing.

c. EfficientNet: Implements Compound Scaling to proportionally balance network depth, width, and input resolution. This approach enhances accuracy without imposing excessive computational cost.

These three models offer an ideal trade-off between accuracy and efficiency, making them suitable for a wide range of deep learning-based image recognition tasks.

2.4. Model

2.4.1. Model Training

Model development was performed in Visual Studio Code using a virtual environment configured via Anaconda. GPU acceleration was enabled using CUDA on an NVIDIA RTX 3060. All three CNN models, namely ConvNeXt Tiny, MobileNetV3 Large, and EfficientNet B0, were trained using a transfer learning approach by initializing with ImageNet 1K pretrained weights. Two training strategies were applied:

a. Without fine-tuning: Only the final layers were retrained, while the rest of the layers remained frozen.
b. With fine-tuning: All layers of the model were retrained to better adapt to the characteristics of the dataset.

Hyperparameters such as number of epochs, batch size, learning rate, and optimizer type were fine-tuned iteratively. Techniques such as EarlyStopping and ReduceLROnPlateau were employed to optimize the training process and prevent overfitting.

2.4.2. Model Performance Evaluation

After training, all models were evaluated using the test set to assess classification performance on previously unseen data. Evaluation metrics included final test accuracy, as well as visual analyses of the training and validation performance trends. Performance evaluation was conducted based on four training scenarios, combining the use of data augmentation and fine-tuning as follows:

i. Without augmentation, without fine-tuning
ii. Without augmentation, with fine-tuning
iii. With augmentation, without fine-tuning
iv. With augmentation, with fine-tune

Tables 6, 7, and 8 present the accuracy results for the ConvNeXt Tiny, MobileNetV3 Large, and EfficientNet B0 architectures, respectively.

Table 6. Accuracy results of ConvNeXt Tiny architecture.

Scenario	No Aug, No FT	No Aug, With FT	With Aug, No FT	With Aug, With FT
A	57,807%	74,419%	78,036%	96,964%
B	60,706%	70,640%	77,500%	96,905%
C	57,072%	67,554%	76,434%	93,996%

Table 7. Accuracy results of MobileNetV3 Large architecture.

Scenario	No Aug, No FT	No Aug, With FT	With Aug, No FT	With Aug, With FT
A	55,483%	65,449%	72,321%	96,439%
B	58,278%	67,770%	73,210%	95,714%
C	55,241%	63,727%	70,699%	93,548%

Table 8. Accuracy results of EfficientNet B0 architecture.

Scenario	No Aug, No FT	No Aug, With FT	With Aug, No FT	With Aug, With FT
A	57,475%	65,449%	68,750%	98,214%
B	56,733%	67,770%	68,690%	97,619%
C	56,073%	63,727%	66,398%	93,548%

The results in Table 6, 7, and 8 clearly indicate that, across all architectures, the highest accuracy was consistently achieved under the scenario that applied both data augmentation and fine-tuning, particularly under Scenario A. Conversely, the lowest accuracy was observed when neither augmentation nor fine-tuning was used. These notable differences underscore the crucial role of augmentation and fine-tuning in enhancing model performance. Data augmentation increases input image variability, thereby improving the model's generalization ability, while fine-tuning enables the model to better adapt to the specific characteristics of the dataset.

3. RESULTS AND DISCUSSION

This section represents the interpret stage of the research, where the results obtained from the model training and evaluation are analyzed and discussed in detail. The goal is to interpret the performance of each CNN architecture across different experimental scenarios, focusing on accuracy, training and testing efficiency, and computational resource usage. These insights serve as the basis for identifying the most suitable model for oral disease image classification.

3.1. Accuracy Comparison of Each Architecture

Model performance was evaluated by comparing the classification accuracy of three CNN architectures, namely ConvNeXt Tiny, MobileNetV3 Large, and EfficientNet B0, across four scenarios: without augmentation and before fine-tuning, without augmentation and after fine-tuning, with augmentation and before fine-tuning, and with augmentation and after fine-tuning. The evaluation results are presented in Table 9.

Table 9. Test accuracy comparison across architectures.

Augmentation	Fine-Tuning	ConvNeXt Tiny	MobileNetV3 Large	EfficientNet B0
No	Before	57,807%	55,483%	57,475%
No	After	74,419%	55,483%	65,449%
Yes	Before	78,036%	72,321%	68,750%
Yes	After	96,964%	96,439%	98,214%

The results demonstrate a significant improvement in accuracy when both data augmentation and fine-tuning are applied. EfficientNet B0 achieved the highest accuracy at 98.21%, followed by ConvNeXt Tiny (96.96%)

and MobileNetV3 Large (96.44%). Without augmentation or fine-tuning, all models showed relatively low accuracy, emphasizing the critical role of these techniques in enhancing model generalization.

3.2. Training and Testing Time Efficiency Evaluation

Computational efficiency is an important consideration when selecting models, particularly for real-time applications or deployment in resource-constrained environments. Tables 10 and 11 present the average training and testing times for each architecture under the four experimental scenarios.

Table 10. Average training time per architecture (in seconds).

Augmentation	Fine-Tuning	ConvNeXt Tiny	MobileNetV3 Large	EfficientNet B0
No	Before	24,45	18,18	20,26
No	After	30,97	21,98	25,53
Yes	Before	46,60	23,71	29,99
Yes	After	92,00	28,01	38,99

Table 11. Testing time per architecture (in seconds).

Augmentation	Fine-Tuning	ConvNeXt Tiny	MobileNetV3 Large	EfficientNet B0
No	Before	1,60	1,49	1,34
No	After	1,87	1,15	1,22
Yes	Before	3,02	2,68	2,32
Yes	After	3,20	2,92	2,33

MobileNetV3 Large demonstrated the fastest training time under optimal conditions (with augmentation and fine-tuning), at 28.01 seconds. However, EfficientNet B0 recorded the shortest testing time, at 2.33 seconds, suggesting that despite its architectural complexity, it remains highly efficient during inference. These results highlight the trade-offs between training duration and inference efficiency across different CNN architectures, offering valuable insights for selecting an appropriate model based on specific deployment needs.

3.3. Evaluation of Computational Resource Efficiency

Further evaluation was conducted by comparing architectural complexity based on the number of parameters and FLOPs (Floating Point Operations). FLOPs indicate the number of floating-point operations required to execute a neural network during inference, serving as a key metric for assessing computational efficiency. A lower FLOPs value generally implies faster computation and reduced cost, enabling the deployment of models on less powerful hardware without compromising performance [17].

Table 12. Parameter count and FLOPs per architecture.

Architecture	Parameters (Million)	FLOPs (Million)
ConvNeXt Tiny	28,6	4.500
MobileNetV3 Large	5,4	219
EfficientNet B0	5,3	390

Table 13. Accuracy vs. parameter count and FLOPs per architecture.

Architecture	Parameters (Million)	FLOPs (Million)	Accuracy Efficiency (% per million parameters)
ConvNeXt Tiny	28,6	4.500	3,39
MobileNetV3 Large	5,4	219	17,86
EfficientNet B0	5,3	390	18,53

Table 12 presents the number of parameters and FLOPs for each model architecture. The parameter count indicates the model's capacity to learn complex features, while FLOPs reflect the computational load required during the inference process. ConvNeXt Tiny has the highest parameter count at 28.6 million and the highest FLOPs at 4,500 million, indicating that it is a high-complexity model with substantial computational demands. In comparison, MobileNetV3 Large and EfficientNet B0 are significantly lighter, each with approximately 5.4 and 5.3 million parameters, and with much lower FLOPs values of 219 million and 390 million, respectively. These figures show that both MobileNetV3 Large and EfficientNet B0 are designed with computational efficiency in mind.

Table 13 further illustrates the relationship between accuracy, parameter count, and computational load through the accuracy-to-parameter ratio. This ratio, expressed as percentage accuracy per million parameters, allows for a more precise assessment of a model's efficiency in converting its capacity into performance. EfficientNet B0 achieves the highest accuracy-to-parameter ratio of 18.53%, highlighting its superior balance between accuracy and resource utilization. MobileNetV3 Large follows closely with a ratio of 17.86%, indicating that it also performs efficiently, particularly in scenarios where hardware limitations must be considered. Although ConvNeXt Tiny demonstrates high classification accuracy, it does so at a significantly higher computational cost, reflected in its much lower efficiency ratio of 3.39%.

These findings suggest that EfficientNet B0 is the most efficient architecture among the three, providing strong performance while maintaining relatively low computational demands. This makes it especially suitable for deployment in real-world environments with limited computational resources. In contrast, ConvNeXt Tiny, despite its robust accuracy, may be less practical in such scenarios due to its high complexity and resource requirements.

3.4. Training and Testing Efficiency Relative to Model Complexity

To evaluate training and testing efficiency in relation to model complexity, an analysis of time per parameter was conducted, as shown in Tables 14 and 15.

Table 14. Training time vs. parameters and FLOPs.

Architecture	Max Training Time (s)	Parameters (Million)	FLOPs (Million)	Efficiency (Time/Parameter, s per million)
ConvNeXt Tiny	92,00	28,6	4.500	3,22
MobileNetV3 Large	28,01	5,4	219	5,19
EfficientNet B0	38,99	5,3	390	7,35

Table 15. Comparison of testing time, number of parameters, and FLOPs for each architecture.

Architecture	Max Testing Time (s)	Parameters (Million)	FLOPs (Million)	Efficiency (Time/Parameter, s per million)
ConvNeXt Tiny	3,20	28,6	4.500	0,00071
MobileNetV3 Large	2,92	5,4	219	0,01333
EfficientNet B0	2,33	5,3	390	0,00597

ConvNeXt Tiny achieved the lowest time per parameter ratio during both training and testing, indicating high computational efficiency per unit of model complexity. However, its overall training and testing durations were the longest, which may limit its suitability in time-sensitive applications. MobileNetV3 Large had the shortest training time, making it advantageous for rapid development cycles and deployment in environments with limited computational resources, despite a higher time per parameter ratio. EfficientNet B0 demonstrated a balanced profile by offering moderate training time and the fastest testing time. This combination, along with its competitive accuracy, positions it as a well-rounded choice for practical applications.

4. CONCLUSION

This study evaluated and compared the performance of three CNN architectures, namely ConvNeXt Tiny, MobileNetV3 Large, and EfficientNet B0, for oral disease classification based on medical images, focusing on both accuracy and computational efficiency. The results showed that EfficientNet B0 achieved the highest accuracy (98.214%) while maintaining computational efficiency, making it the most suitable choice for deployment in resource-constrained clinical settings. ConvNeXt Tiny demonstrated robust performance across scenarios, albeit with higher resource demands, whereas MobileNetV3 Large offered a lightweight solution with competitive accuracy. The consistent improvements observed through data augmentation and fine-tuning further emphasize their essential role in enhancing model generalization. These findings provide valuable insights into selecting appropriate CNN architectures for real-world medical image diagnosis systems.

LITERATURE

[1] Institute for Health Metrics and Evaluation (IHME), “Global Burden of Disease Collaborative Network Global burden of disease study 2019 (GBD 2019) results,” 2020.

[2] World Health Organization, “Global oral health status report: Towards universal health coverage for oral health by 2030,” 2022.

[3] Z. Cao, J. Huang, X. He, and Z. Zong, “BND-VGG-19: A deep learning algorithm for COVID-19 identification utilizing X-ray images,” *Knowl Based Syst*, vol. 258, Dec. 2022, doi: 10.1016/j.knosys.2022.110040.

[4] A. Ahmed, D. Velayudhan, T. Hassan, M. Bennamoun, E. Damiani, and N. Werghi, “Enhancing security in X-ray baggage scans: A contour-driven learning approach for abnormality classification and instance segmentation,” *Eng Appl Artif Intell*, vol. 130, p. 107639, Apr. 2024, doi: 10.1016/j.engappai.2023.107639.

[5] D. R. Beddiar, M. Oussalah, U. Muhammad, and T. Seppänen, “A Deep learning based data augmentation method to improve COVID-19 detection from medical imaging,” *Knowl Based Syst*, vol. 280, Nov. 2023, doi: 10.1016/j.knosys.2023.110985.

[6] P. Fawaz, P. El Sayegh, and B. Vande Vannet, “What is the current state of artificial intelligence applications in dentistry and orthodontics?,” *J Stomatol Oral Maxillofac Surg*, vol. 124, no. 5, p. 101524, Oct. 2023, doi: 10.1016/j.jormas.2023.101524.

[7] M. Vodanović, M. Subašić, D. Milošević, and I. Savić Pavičin, “Artificial Intelligence in Medicine and Dentistry,” *Acta Stomatol Croat*, vol. 57, no. 1, pp. 70–84, Mar. 2023, doi: 10.15644/asc57/1/8.

[8] Y. Kasnanda Bintang and H. Imaduddin, “Pengembangan Model Deep Learning untuk Deteksi Retinopati Diabetik Menggunakan Metode Transfer Learning,” *Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. 9, no. 3, pp. 1442–1455, 2024, doi: 10.29100/jipi.v9i3.5588.

[9] F. Carrillo-Perez, O. E. Pecho, J. C. Morales, R. D. Paravina, A. D. Bona, R. Ghinea, R. Pulgar, M. del Mar Pérez, and L. J. Herrera, “Applications of artificial intelligence in dentistry: A comprehensive review,” *Journal of Esthetic and Restorative Dentistry*, vol. 34, no. 1, pp. 259–280, Jan. 2022, doi: 10.1111/jerd.12844.

[10] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.03545

[11] A. Howard, M. Sandler, G. Chu, L. C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, “Searching for MobileNetV3,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.02244

[12] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” May 2019, [Online]. Available: http://arxiv.org/abs/1905.11946

[13] N.S. Piyarathne, S.N. Liyanage, R.M.S.G.K. Rasnayaka, P.V.K.S. Hettiarachchi, G.A.I. Devindi, F.B.A.H. Francis, D.M.D.R. Dissanayake, R.A.N.S. Ranasinghe, M.B.D. Pavithya, I.B. Nawinne, R.G. Ragel, and R.D. Jayasinghe, "A comprehensive dataset of annotated oral cavity images for diagnosis of oral cancer and oral potentially malignant disorders," *Oral Oncol*, vol. 156, Sep. 2024, doi: 10.1016/j.oraloncology.2024.106946.

[14] H. Mason and C. Wiggins, "A Taxonomy of Data Science." Accessed: Sep. 19, 2024. [Online]. Available: https://web.archive.org/web/20160220042455/dataists.com/2010/09/a-taxonomy-of-data-science/

[15] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Text Data Augmentation for Deep Learning," *J Big Data*, vol. 8, no. 1, p. 101, Dec. 2021, doi: 10.1186/s40537-021-00492-0.

[16] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, Dec. 2022, doi: 10.1016/j.array.2022.100258.

[17] J. Chen, S. Kao, H. He, W. Zhuo, S. Wen, and C. H. Lee, "Run, Don't Walk: Chasing Higher FLOPS for Faster Neural Networks," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. [Online]. Available: https://doi.org/10.1109/CVPR52729.2023.01157.