JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 11. NO. 3 FEBRUARY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i3.7462

PERFORMANCE EVALUATION OF LIGHTWEIGHT DEEP LEARNING MODELS FOR BORAX-CONTAMINATED MEATBALL IMAGE CLASSIFICATION

Aryo Michael[1*]; Irene Devi Damayanti[1]

Program Studi Teknik Informatika[1]
Universtas Kristen Indonesia Toraja, Tana Toraja, Indonesia[1]
https://ukiroraja.ac.id[1]
aryomichael@ukitoraja.ac.id* irenedamayanti@ukitoraja.ac.id

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

The creation is distributed under the Creative Commons Attribution-NonCommercial 4.0 International License.

Abstract*—Food safety, particularly concerning the use of illegal additives such as borax in processed meat products like meatballs, remains a critical issue in Indonesia. This study analyzes the performance of several lightweight deep learning models based on Convolutional Neural Networks (CNN) and Transformers to classify images of meatballs containing borax, enabling their deployment on resource-constrained devices such as smartphones. Data collection involved capturing 1,429 images of meatballs with and without borax using a smartphone camera under varying lighting conditions and shooting angles. The five main architectures evaluated were ConvNeXt-Nano, Swin-Tiny, ViT-Tiny, MobileViT-XS, and EfficientNet-B0. Hyperparameter optimization was conducted using Optuna, followed by training with a 5-fold cross-validation scheme. Model evaluation metrics included accuracy, precision, recall, F1 score, and inference speed. The results show that MobileViT-XS was the best-performing architecture, achieving 65.93% accuracy, 0.703 precision, 0.706 recall, 0.659 F1 score, and efficient memory consumption (45.94 MB). These findings indicate that a hybrid approach combining the strengths of CNNs and Transformers can achieve an optimal balance between detection accuracy and computational efficiency. Therefore, such models have the potential to be applied as food safety detection systems on devices with limited resources.*

Keywords*: Deep Learning, Food Safety, Image Classification, Lightweight Model.*

Intisari—*Keamanan pangan, khususnya terkait penggunaan bahan tambahan ilegal seperti boraks dalam produk daging olahan seperti bakso, tetap menjadi masalah kritis di Indonesia. Studi ini menganalisis kinerja beberapa model deep learning ringan berbasis Jaringan Saraf Konvolusional (CNN) dan Transformers untuk mengklasifikasikan gambar bakso yang mengandung boraks, sehingga memungkinkan penerapan model tersebut pada perangkat dengan sumber daya terbatas seperti smartphone. Pengumpulan data melibatkan pengambilan 1.429 gambar bakso dengan dan tanpa boraks menggunakan kamera smartphone dalam kondisi pencahayaan dan sudut pengambilan gambar yang bervariasi. Lima arsitektur utama yang dievaluasi adalah ConvNeXt-Nano, Swin-Tiny, ViT-Tiny, MobileViT-XS, dan EfficientNet-B0. Optimasi hiperparameter dilakukan menggunakan Optuna, diikuti dengan pelatihan menggunakan skema validasi silang 5-fold. Metrik evaluasi model meliputi akurasi, presisi, recall, skor F1, dan kecepatan inferensi. Hasil menunjukkan bahwa MobileViT-XS merupakan arsitektur dengan kinerja terbaik, mencapai akurasi 65,93%, presisi 0,703, recall 0,706, skor F1 0,659, dan konsumsi memori yang efisien (45,94 MB). Temuan ini menunjukkan bahwa pendekatan hibrida yang menggabungkan keunggulan CNN dan Transformers dapat mencapai keseimbangan optimal antara akurasi deteksi dan efisiensi komputasi. Oleh karena itu, model-model semacam ini berpotensi diterapkan sebagai sistem deteksi keamanan pangan pada perangkat dengan sumber daya terbatas.*

Kata Kunci*: deep learning, keamanan pangan, klasifikasi citra, model ringan.*

INTRODUCTION

Food safety has become a global concern that cannot be overlooked, particularly in efforts to ensure that all individuals have access to healthy and nutritious food [1], [2]. The World Health Organization (WHO) estimates that unsafe food causes approximately 600 million cases of illness and 420,000 deaths annually worldwide, primarily due to foodborne pathogens [3]. Therefore, it is essential for individuals and society as a whole to understand the associated risks and collaborate to implement effective preventive measures, ensuring the availability of safe food for consumption.

In the modern era, shifts in lifestyle that prioritize practicality and efficiency, combined with the integration of technology in food production, have made fast food and street food the primary choices for daily consumption [4]. This trend not only reflects a transformation in eating habits but also significantly impacts the local economy, public health, and culinary culture. One example of a traditional food that remains very popular among Indonesians is "Bakso." Bakso is made from a mixture of ground meat and flour, providing protein and essential amino acids necessary for the human body [5]. However, meat as the main ingredient is susceptible to contamination by pathogens such as bacteria, which accelerate spoilage [6]. This vulnerability has led some producers to use illegal food additives, such as sodium tetraborate (Borax), to improve texture and extend the shelf life of meatballs. Although this practice aims to prevent economic losses caused by premature spoilage, it violates government regulations.

Consumers have limited ability to visually detect hazardous contaminants in food, necessitating a real-time monitoring system to protect public health. In this context, smartphones widely owned by the public offer significant potential as computer vision-based platforms for food safety detection. The advantages of smartphones in this application include high accessibility, enabling widespread use without reliance on specialized tools; continuously improving processing capabilities supported by modern processor technology; and high-quality cameras capable of capturing the visual details required for classification tasks. However, implementing automatic detection systems on smartphones presents significant challenges. Limited computational resources such as memory capacity, processing speed, and battery life pose major obstacles. To run optimally on mobile devices, an efficient and lightweight model is essential. This model must maintain high detection accuracy without compromising device performance. Therefore, the development of smartphone-based food safety detection systems must carefully balance technical requirements with device limitations.

Deep learning has made significant advances in object classification, including applications in the food sector [7]. Improvements in computing technology have facilitated the use of deep learning models to analyze food composition [8]. Detection of borax contamination in meatballs has been carried out using various technical approaches as documented in previous studies. Sensor-based methods, including resistance sensors that measure changes in electrical conductivity in meatball samples, enable direct chemical analysis but require direct physical interaction with the product [9]. As an alternative, computer vision techniques offer a non-invasive and rapid solution for screening meatballs without direct contact.

However, most related studies rely on complex deep learning architectures, which, although effective, require significant computing and memory resources. These limitations make them less suitable for low-power devices such as smartphones [10]. Therefore, there is a research gap in the development of efficient deep learning models that balance high accuracy and computational optimization for mobile applications in borax detection in meatballs.

Convolutional Neural Networks (CNNs) have demonstrated high effectiveness in digital image classification, although they demand substantial computational resources and large, diverse datasets [11]. As an alternative, Vision Transformers (ViTs) provide optimal performance across various scales through self-attention mechanisms that capture global dependencies in images [12], while hybrid models that integrate attention mechanisms with CNNs offer enhanced memory and parameter efficiency by combining local feature extraction with global context modeling [13].

This study evaluates five different deep learning architectures for detecting borax contamination in meatball images. The architectures studied include Convolutional Neural Networks (CNNs), Vision Transformers (ViTs), and hybrid models that combine self-attention mechanisms with CNNs. The two CNN architectures explored are EfficientNet-B0 and ConvNeXt-Nano: EfficientNet-B0 uses combined scaling (depth, width, resolution) to balance computational complexity and efficiency, while ConvNeXt-Nano integrates design principles from ViTs into traditional CNN structures. For Transformer-based architectures, ViT-Tiny was tested as a pure model representation that processes images through multi-head self-attention layers to capture global dependencies. Meanwhile, hybrid models such as Swin-Tiny and MobileViT-XS combine the advantages

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

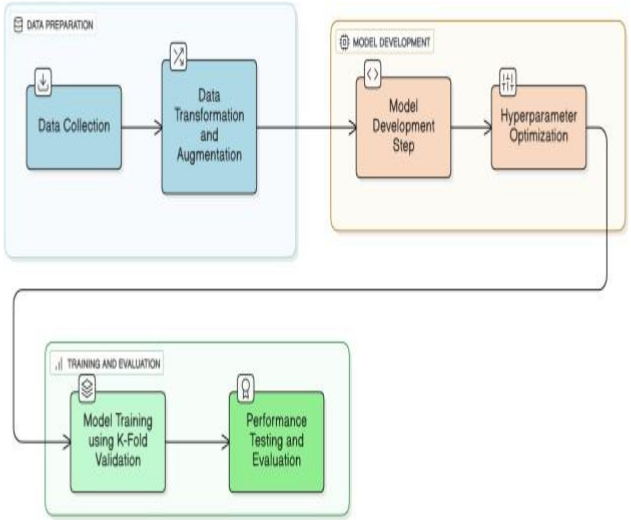
VOL. 11. NO. 3 FEBRUARY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i3.7462

of local feature extraction (CNN) with global context modeling (Transformer), with MobileViT-XS specifically optimized for mobile and edge devices.

This study aims to evaluate 3 primary architectures CNN, Transformers, and their hybrid models designed for resource-limited devices. The evaluation is based on three key criteria: (1) detection accuracy, (2) inference speed, and (3) memory usage efficiency. The results are expected to provide a foundation for developing a practical, accurate, and sustainable borax detection system for resource-constrained devices, thereby supporting the optimization of food safety in Indonesia through measurable technological solutions.

MATERIALS AND METHODS

This research was conducted through six main stages as illustrated in Figure 1, starting from data collection; data transformation and augmentation; deep learning model development; hyperparameter optimization using Optuna; model training with k-fold validation; and performance testing and evaluation.

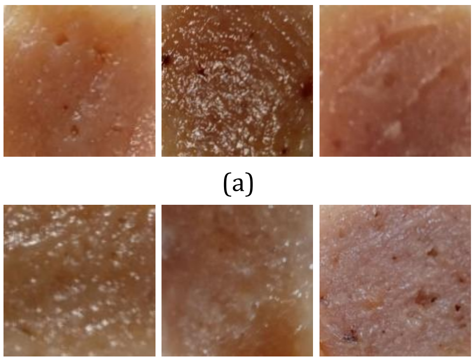

DATA PREPARATION
Data Collection
Data Transformation and Augmentation
MODEL DEVELOPMENT
Model Development Step
Hyperparameter Optimization
TRAINING AND EVALUATION
Model Training using K-Fold Validation
Performance Testing and Evaluation

Source: (Research Results, 2025)

Figure 1. Research Workflow

1. Data Collection

Data collection was conducted using smartphone cameras to capture a variety of shooting angles and lighting conditions. This approach aimed to address the challenge of classifying meatballs containing Borax in real-world environments, where visual similarities between classes are often subtle, and variations in lighting and background increase complexity. A total of 1,429 images were collected and meticulously labeled by category to support the training of the classification model.



Source: (Research Results, 2025)

Figure 2. Sample image (a) Meatballs Contain Borax (b) Meatballs Without Contain Borax

For the training set, 600 images of meatballs containing borax and 600 images of meatballs without borax were utilized, resulting in a perfectly balanced training dataset of 1,200 images. The testing set comprised 145 images of meatballs containing borax and 84 images of meatballs without borax, totaling 229 test images.

2. Data Transformation and Augmentation

Data transformation and dataset configuration in this study were divided into two categories: training and testing data. For training data, a series of transformation processes are applied, including resizing images to 224 × 224 pixels, as well as data augmentation through random horizontal and vertical flipping with a probability of 0.5. This will expand the training data set by generating variations in image orientation, and will help the model learn visual features better and not easily overfit [14]. After augmentation, normalization using mean values [0.485, 0.456, 0.406] and standard deviations [0.229, 0.224, 0.225], based on the standard parameters of the pre-trained model [15].

Meanwhile, the test data only underwent two processes: resizing to 224 × 224 pixels and normalization with identical parameters. Data augmentation was deliberately excluded at this stage to ensure that model evaluation was consistent and objective, reflecting performance in a real-world environment without artificial intervention. This strategy is designed to validate the generalization capabilities of models that have been improved through augmentation in the training phase, while avoiding bias due to excessive data modification. This configuration ensures that models not only excel in trained conditions but are also able to adapt effectively to previously unseen data variations, providing an important foundation for practical applications such as food safety inspection or contaminant detection.

3. Model Development

After preparing the entire dataset, the modeling process proceeded using CNN and Transformer architectures. In this study, five deep learning models based on CNN and Transformer architectures were specifically developed for resource-limited devices.

a. ConvNeXt-Nano

ConvNeXt is a modern CNN architecture designed to rival the performance of Vision Transformer (ViT) models in image classification tasks [16]. It incorporates contemporary design elements inspired by ViT, such as patch embedding, normalization, and depthwise convolution, while maintaining the computational efficiency characteristic of CNNs. The latest version, ConvNeXt V2, integrates masked autoencoders and Global Response Normalization, which enhance feature representation and improve accuracy [17].

ConvNeXt-Nano is a lightweight variant of the ConvNeXt architecture designed for image classification tasks, offering high computational efficiency without compromising accuracy. It provides advantages such as fewer parameters, low memory consumption, and fast inference speed, while maintaining competitive accuracy [18]. Additionally, this model has demonstrated effectiveness in handling limited data and small objects.

b. Swin-Tiny

Swin Transformer is a vision transformer architecture that employs a shifted window mechanism to achieve high efficiency and accuracy in image classification [19]. Swin-Tiny, a lightweight variant of the Swin Transformer, offers advantages in parameter efficiency and the ability to capture extensive spatial dependencies, making it well-suited for devices with limited resources [20].

c. ViT-Tiny

The Vision Transformer (ViT) model was introduced by Dosovitskiy et al. [21]. ViT-Tiny is a lightweight variant of the ViT architecture specifically designed for computational efficiency and optimal performance, particularly on limited datasets [20]. While retaining the core principles of ViT such as processing images as sequences of patch tokens and utilizing self-attention mechanisms ViT-Tiny has significantly fewer parameters and lower complexity, thereby reducing the computational load. This design enables ViT-Tiny to balance efficiency and accuracy, making it a practical solution for systems with computational constraints that still require reliable visual analysis capabilities.

d. MobileViT-XS

MobileViT is a deep learning architecture that integrates CNN and vision transformers. It has proven effective for implementation on mobile devices, delivering performance comparable to pure transformers while remaining computationally efficient [22]. MobileViT-XS is a lightweight variant of MobileViT that combines the strengths of CNNs and vision transformers for image classification on mobile or edge devices.

e. EfficientNet-B0

EfficientNet is a Convolutional Neural Network (CNN) model developed by Google to address challenges in object and image recognition [23]. This model is designed with eight different architectures, namely EfficientNet-B0 to EfficientNet-B7, where each version has increasing complexity along with an increase in parameters. As a consequence of the increase in the number of parameters, the accuracy of the model also tends to increase proportionally. It employs a compound scaling technique that systematically balances the network's depth, width, and resolution to achieve high accuracy with low floating-point operations (FLOPs), making it one of the most efficient models for image classification with an optimal balance between performance and computational complexity [24].

4. Hyperparameter Optimizing

Hyperparameter optimization is a crucial factor in developing a robust deep learning model [25].In this study, hyperparameter tuning was conducted using Optuna, a Sequential Model-Based Optimization (SMBO) library that adaptively explores the hyperparameter search space [26]. Table 2 presents the parameters optimized for each model.

Table 2. Hyperparameter Tuning Configuration

Hyperparameter	Value
Learning Rate	0.00001 - 0.01
Batch size	8, 16, 32
Optimizier	Adam, AdamW, SGD
Weight decay	0.000001 - 0.01

Source: (Research Results, 2025)

Each experiment conducted using the Optuna library represents a unique combination of hyperparameters, which is then evaluated based on its performance on the validation set.

5. Model Training

The optimal parameter configuration identified during the hyperparameter optimization stage is used to train the final model employing a 5-fold cross-validation scheme with 30 epochs to ensure convergence of model parameters. The final model performance is evaluated based on the average validation results across all folds.

This approach offers a more reliable estimate of generalization performance while minimizing the bias associated with a single data split. This study aims to evaluate the performance of lightweight deep learning

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 3 FEBRUARY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i3.7462

models in solving classification problems. Additionally, hyperparameter tuning was performed to identify the optimal configuration for each model tested. All experiments and analyses were conducted using five lightweight deep learning models designed to support computational efficiency on resource- limited devices.

6. Testing and Evaluation

Model evaluation covers important performance aspects such as accuracy, precision, sensitivity, F1-score, and inference speed, thereby providing recommendations on the most suitable model to be implemented in resource-constrained computing systems.

$$Accurassy = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$f1 - score = 2 \text{ x } \frac{Precision \text{ x } Recall}{Precision + Recall} \quad (4)$$

Inference speed measurements were performed to evaluate the computational efficiency of the model. The measurement process was carried out by calculating the difference in model execution time in a single batch of test data. The time difference produced in each iteration was then converted into milliseconds and averaged to obtain the average inference value in ms/batch.

All calculations and modeling in this study were implemented using the PyTorch deep learning framework. To train and evaluate the model, the Google Colab platform was chosen as the main computing environment, supported by an NVIDIA Tesla T4 GPU. The use of the T4 GPU played an important role in accelerating the computing process, while also increasing the efficiency of the experiment by utilizing optimal hardware resources.

RESULTS AND DISCUSSION

Hyperparameter Optimization

Hyperparameter optimization is a crucial factor in developing an effective classification model. The optimal hyperparameters for each model are presented in Table 3.

Table 3. Optimized hyperparameters for all models

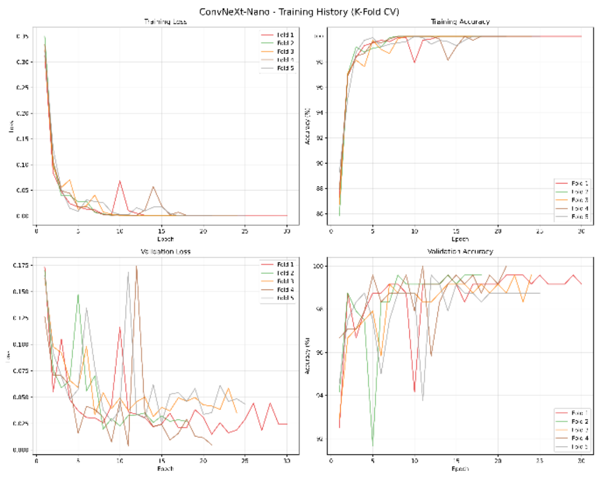
Model	Learning Rate	Batch size	Optimizier	Weight decay
ConvNeXt-Nano	0.000012	16	Adam	0.00096
Swin-Tiny	0.000208	32	AdamW	0.000001
ViT-Tiny	0.000024	32	Adam	0.000002
MobileViT-XS	0.000169	32	AdamW	0.001299
EfficientNet-B0	0.000353	32	AdamW	0.000018

Source: (Research Results, 2025)

Training Model

After completing the hyperparameter optimization process, the model was trained using cross-validation techniques.

ConvNeXt-Nano

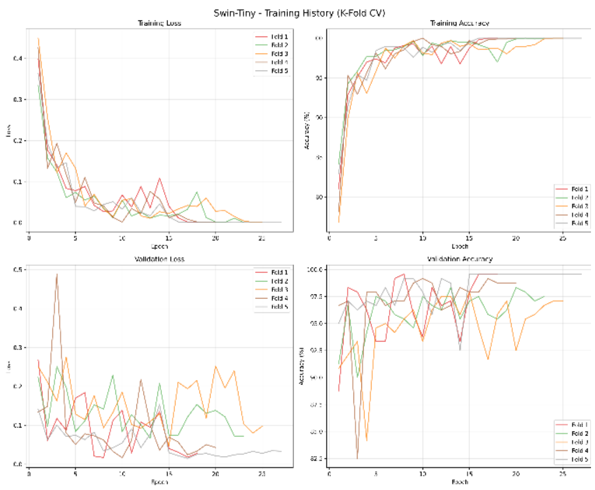


Source: (Research Results, 2025)
Figure 3. Training history of ConvNeXt-Nano model across 5-fold over 30 epochs

The training history of ConvNeXt-Nano in Figure 3 shows that the Training Loss graph indicates a drastic decrease in the first five epochs, from around 0.35 to close to 0.02 across all folds. This decrease indicates that the ConvNeXt-Nano model has excellent learning capacity for the training dataset. After the 5th epoch, the loss value remains stable at around zero, indicating that the model has reached optimal convergence on the training data. In terms of Training Accuracy, there is a significant increase in line with the decrease in loss. Accuracy increased rapidly from around 88% to nearly 100% in the first five epochs, then remained consistent at a perfect score (100%) until the 30th epoch. This reinforces the impression that the model is able to understand patterns in the training data very effectively. Meanwhile, Validation Loss shows a different pattern from training loss. Although there was a significant decrease at the beginning of training (epochs 0–5), considerable fluctuations were observed throughout the process, with values ranging from 0.01 to 0.175. These fluctuations, especially in Fold 4 and Fold 5, which peaked at nearly 0.175 around epoch 10, indicate instability in the model's generalization ability. For Validation Accuracy, the figures ranged from 92% to 100% with an inconsistent pattern. Although some folds (such as Fold 1 and Fold 2) showed relative stability above 99% after epoch 10, other folds (especially Folds 3, 4, and 5) experienced significant volatility. Sharp declines at certain points, such as Fold 1 dropping to 94% around epoch 10, indicate the model's sensitivity to variations in the validation data. Overall, these results confirm that ConvNeXt-Nano performs strongly in training, but still requires adjustments to improve generalization stability on unseen data.

Swin-Tiny

Based on the Swin-Tiny model training history graph in Figure 4, using 5-Fold Cross Validation, several important patterns emerge in the model learning process.

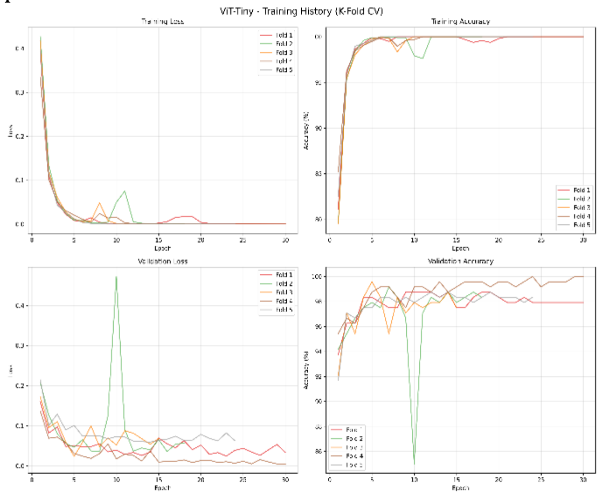


Source: (Research Results, 2025)
Figure 4. Training history of Swin-Tiny model across 5-fold over 30 epochs

In the training loss and training accuracy graphs, the model shows excellent ability in learning the training data. The training loss decreases sharply, from about 0.45 in early epochs to nearly 0 in the 5–10 epoch range. At the same time, training accuracy rises rapidly from about 78% to almost 100% across all folds. This rapid convergence shows the model can effectively recognize patterns in the training data. However, the model's performance on the validation data tells a different story. The validation loss graph shows significant fluctuations across all folds, with several spikes indicating instability. Unlike the smooth downward trend in training loss, validation loss does not show a consistent pattern, but rather tends to fluctuate throughout the training process. Meanwhile, validation accuracy ranges from 92% to 99%, with considerable variation between epochs. Although these figures are generally good, there is a clear gap between training accuracy (reaching 100%) and validation accuracy (a maximum of 99%). These patterns indicate overfitting, where the model is too specific to the training data and loses its ability to generalize to data it has never seen before. This is reflected in the significant difference in performance between training and validation, as well as unstable fluctuations in validation metrics. This condition needs to be addressed, for example, through regularization techniques, data augmentation, or model architecture adjustments to improve generalization capabilities.

ViT-Tiny

The training history graph of the ViT-Tiny (Vision Transformer) model in Figure 5 shows impressive performance in the 5-Fold Cross Validation scenario.



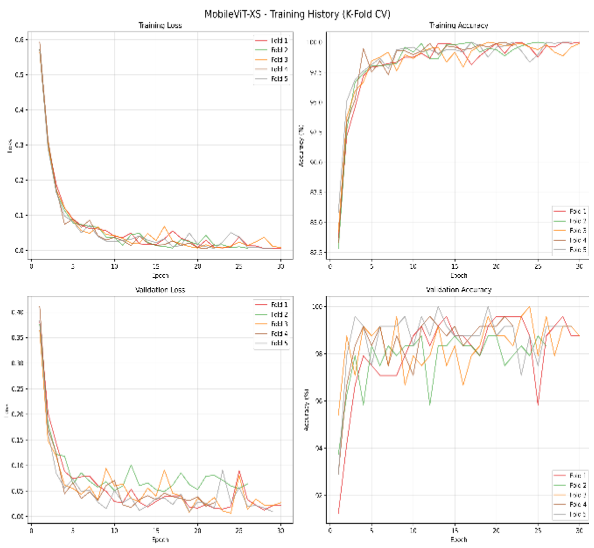
Source: (Research Results, 2025)
Figure 5. Training history of ViT-Tiny model across 5-fold over 30 epochs

The loss decreased from 0.42 to almost zero in 10 epochs. Training accuracy increased from 80% to almost 100% and remained stable. Almost all folds show smooth learning curves. Only Fold 3 experienced slight fluctuations. In validation, the loss remained low (0.02–0.08), and the accuracy was very high (98–100%) in most folds. Fold 3 experienced a spike in loss and a drop in accuracy to 85%, but it recovered quickly. The small difference between training and validation accuracy indicates strong generalization. The model did not show significant overfitting. These results confirm the consistency and effectiveness of the ViT-Tiny model in rigorous evaluation.

MobileViT-XS

Based on the training history graph in Figure 6, the MobileViT-XS model shows stable learning dynamics during the training phase, with more significant variations in the validation phase.

VOL. 11. NO. 3 FEBRUARY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i3.7462



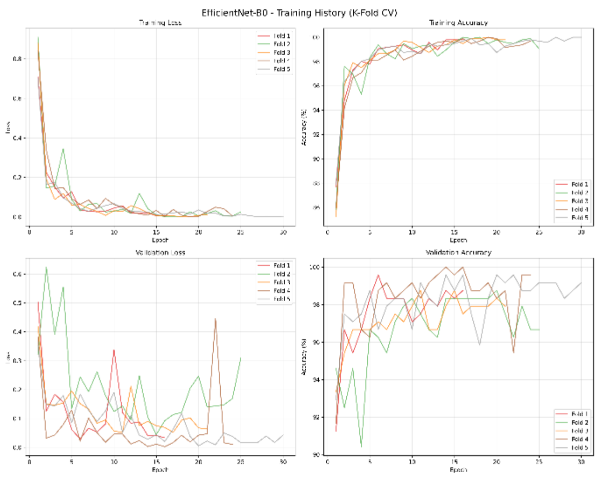
Source: (Research Results, 2025)
Figure 6. Training history of MobileViT-XS model across 5-fold over 30 epochs

In the training loss plot, all folds experienced a consistent decline from an initial value of around 0.5 to below 0.05 at epoch 30. This indicates an efficient convergence process, demonstrating the model's ability to effectively minimize training errors. The increase in training accuracy also reflects the model's effectiveness in understanding data patterns. The accuracy value increases sharply from around 85% at the beginning of training to nearly 100% after the 20th epoch and remains stable until the end of the training phase.

This stability indicates that the model has reached an optimal level of understanding of the training data. During the validation phase, validation loss showed fluctuations between folds, with most values ranging from 0.05 to 0.15 throughout the epoch. However, there was a temporary spike to 0.4 in several folds (mainly around epochs 10 to 15), which may be related to differences in data characteristics between folds. Nevertheless, validation accuracy remained competitive, ranging from 88% to 100%. Most folds achieved an accuracy above 95% after epoch 10 and maintained it until the end of training.

EfficientNet-B0

As shown in the training history graph in Figure 7, the performance of the EfficientNet-B0 model based on the training history graph with K-Fold Cross-Validation shows that the model converges rapidly and efficiently.



Source: (Research Results, 2025)
Figure 7. Training history of EfficientNet-B0 model across 5-fold over 30 epochs

The training loss decreases sharply from around 0.9 to close to 0.0 in the first 10 epochs, while the training accuracy reaches 98–100% from the 5th epoch and remains stable until the end of training. This shows that the model is able to learn the training data patterns optimally. On the validation side, although the validation loss is in a low range (0.0–0.3), there is higher fluctuation compared to training, and the validation accuracy fluctuates between 90% and 100%, with most folds reaching 96–99% in the final epoch. The variability between folds, especially in Fold 5, which shows significant fluctuations, indicates sensitivity to data partitioning and potential heterogeneity in the dataset. The striking difference between the training and validation metrics after the 10th epoch is an indicator of overfitting, where the model shows near-perfect performance in training but is less consistent in generalizing to new data.

Table 4 presents a summary of the training results from five model architectures using three main indicators: average accuracy, average training time, and average memory usage. These three metrics not only describe the predictive capabilities of each model, but also the computational efficiency required during the training process.

Table 4. Model Training Results

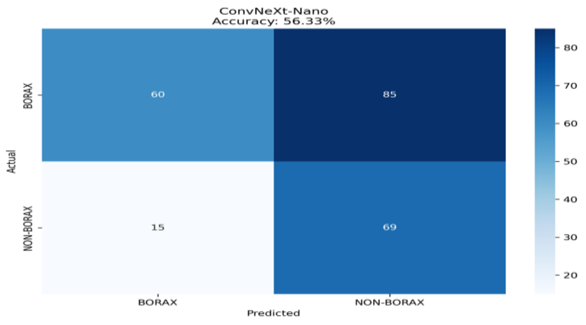
Model	Average Accuracy	Average Training Time	Average Memory Usage
ConvNeXt-Nano	99.67%	308.82s	255.96 MB
Swin-Tiny	98.92%	326.44s	451.03 MB
ViT-Tiny	99.25%	212.51s	100.64 MB
MobileViT-XS	99.58%	274.17s	45.94 MB
EfficientNet-B0	99.33%	222.43s	79.01 MB

Source: (Research Results, 2025)

The ConvNeXt-Nano model achieved the highest accuracy of 99.67%, with a training time of 308.82 seconds and memory usage of 255.96 MB. Next, Swin-Tiny attained an accuracy of 98.92% with a training time of 326.44 seconds, but exhibited higher memory usage at 451.03 MB. The ViT-Tiny model demonstrated balanced performance, achieving 99.25% accuracy, a training time of 212.51 seconds, and memory usage of 100.64 MB. In contrast, MobileViT-XS was the most memory-efficient, using only 45.94 MB, while maintaining an accuracy of 99.58% and a training time of 274.17 seconds. Meanwhile, EfficientNet-B0 recorded an accuracy of 99.33% with a training time of 222.43 seconds and memory usage of 79.01 MB.

Testing and Evaluation

After completing the model training process, testing was conducted using a pre-prepared test dataset consisting of 229 images samples (145 borax samples and 84 non-borax samples).

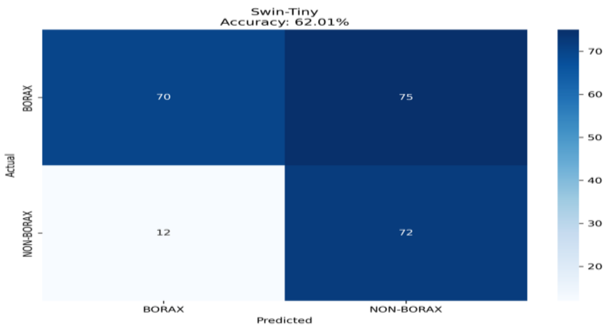


Source: (Research Results, 2025)
Figure 8. Confusion Matrix for ConvNeXt-Nano Model

The classification results of the ConvNeXt-Nano model are shown through the confusion matrix in Figure 8. The model successfully classified 60 borax samples correctly as borax (true positive) and 69 non-borax samples correctly as non-borax (true negative). However, there were 85 borax samples misclassified as non-borax (false negative) and 15 non-borax samples misclassified as borax (false positive).

Based on the confusion matrix, performance metrics were calculated using equations (1) through (4). Using equation (1), the overall model accuracy was obtained at 0.563 with the calculation (60 + 69)/229 = 129/229. For the borax class, equation (2) yielded a precision of 0.800 from 60/(60+15) = 60/75, equation (3) yielded a recall of 0.414 from 60/(60+85) = 60/145, and equation (4) yielded an F1-score of 0.545. Meanwhile, the non-borax class showed a precision of 0.448 from 69/(69+85) = 69/154, a recall of 0.821 from 69/(69+15) = 69/84, and an F1-score of 0.580.

Macro-average metrics that give equal weight to both classes yielded a macro precision of 0.624, macro recall of 0.618, and macro F1-score of 0.563.

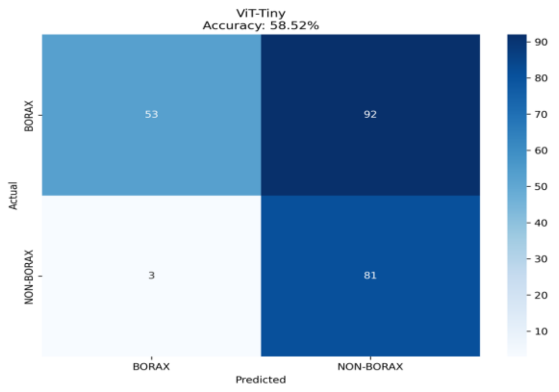


Source: (Research Results, 2025)
Figure 9. Confusion Matrix for Swin-Tiny Model

The classification results of the Swin-Tiny model on the test dataset are shown in the confusion matrix in Figure 9. The model successfully classified 70 borax samples correctly as borax (True Positive) and 72 non-borax samples correctly as non-borax (True Negative). However, there were 75 borax samples misclassified as non-borax (False Negative) and 12 non-borax samples misclassified as borax (False Positive).

Based on the confusion matrix, performance metrics were calculated using equations (1) through (4). Using equation (1), the overall model accuracy was obtained at 0.620 with the calculation (70 + 72)/229 = 142/229. For the borax class, equation (2) yielded a precision of 85.4% or 0.854 from 70/(70+12) = 70/82, equation (3) yielded a recall of 0.483 from 70/(70+75) = 70/145, and equation (4) yielded an F1-score of 0.614. Meanwhile, the non-borax class showed a precision of 0.490 from 72/(72+75) = 72/147, a recall of 0.857 from 72/(72+12) = 72/84, and an F1-score of 0.623.

Macro-average metrics that give equal weight to both classes yielded a macro precision of 0.672, macro recall of 67.0% or 0.670, and macro F1-score of 0.619.

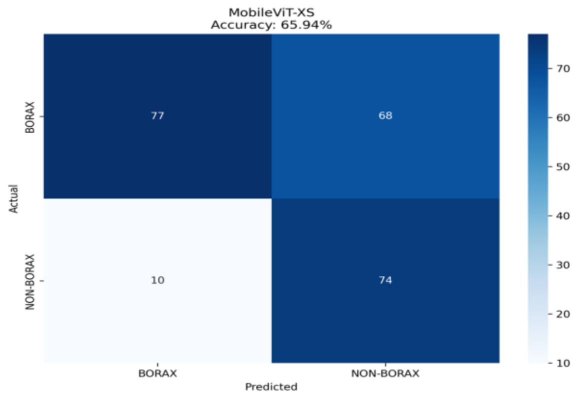


Source: (Research Results, 2025)
Figure 10. Confusion Matrix for ViT-Tiny Model

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 11. NO. 3 FEBRUARY 2026
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v11i3.7462

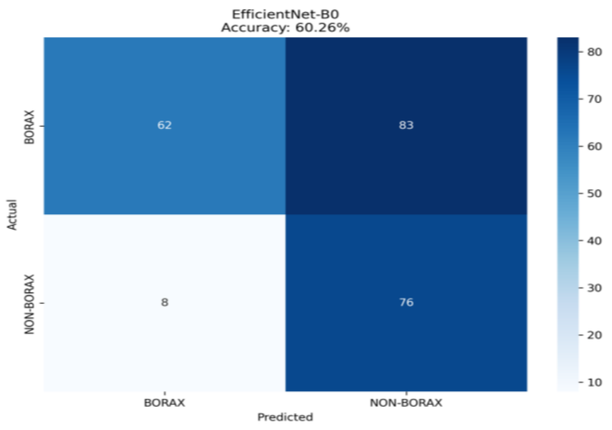
ViT-Tiny model classification results based on the confusion matrix in Figure 10. The model successfully classified 53 borax samples correctly as borax (true positive) and 81 non-borax samples correctly as non-borax (true negative). However, there were 92 borax samples misclassified as non-borax (false negative) and 3 non-borax samples misclassified as borax (false positive). Using equation (1), the overall model accuracy was obtained at 0.585 with the calculation (53 + 81)/229 = 134/229. For the borax class, equation (2) yielded a precision of 94.6% or 0.946 from 53/(53+3) = 53/56, equation (3) yielded a recall of 0.366 from 53/(53+92) = 53/145, and equation (4) yielded an F1-score of 0.527. Meanwhile, the non-borax class showed a precision of 0.468 from 81/(81+92) = 81/173, a recall of 0.964 from 81/(81+3) = 81/84, and an F1-score of 0.631. Macro-average metrics that give equal weight to both classes yielded a macro precision of 0.707, macro recall of 0.665, and macro F1-score of 0.579.



Source: (Research Results, 2025)
Figure 11. Confusion Matrix for MobileViT-XS Model

MobileViT-XS model classification results based on the confusion matrix in Figure 11. The model accurately identified 77 borax samples as borax (true positive) and 74 non-borax samples as non-borax (true negative). Nevertheless, there were 68 borax samples incorrectly classified as non-borax (false negative) and 10 non-borax samples wrongly identified as borax (false positive).

Applying equation (1), the overall model accuracy reached 65.9% or 0.659 with the computation (77 + 74)/229 = 151/229. For the borax class, equation (2) produced a precision of 0.885 from 77/(77+10) = 77/87, equation (3) generated a recall of 0.531 from 77/(77+68) = 77/145, and equation (4) resulted in an F1-score of 0.664. Conversely, the non-borax class demonstrated a precision of 0.521 from 74/(74+68) = 74/142, a recall of 0.881 from 74/(74+10) = 74/84, and an F1-score of 0.655. Macro-average metrics that assign equal weight to both classes produced a macro precision of 0.703, macro recall of 0.706, and macro F1-score of 0.659.



Source: (Research Results, 2025)
Figure 12. Confusion Matrix for EfficientNet-B0 Model

EfficientNet-B0 model classification results based on the confusion matrix in Figure 12. The model accurately identified 62 borax samples as borax (true positive) and 76 non-borax samples as non-borax (true negative). Nevertheless, there were 83 borax samples incorrectly classified as non-borax (false negative) and 8 non-borax samples wrongly identified as borax (false positive).

Applying equation (1), the overall model accuracy reached 0.603, with the computation (62 + 76)/229 = 138/229. For the borax class, equation (2) produced a precision of 0.886 from 62/(62+8) = 62/70, equation (3) generated a recall of 0.428 from 62/(62+83) = 62/145, and equation (4) resulted in an F1-score of 0.577. Conversely, the non-borax class demonstrated a precision of 0.478 from 76/(76+83) = 76/159, a recall of 0.905 from 76/(76+8) = 76/84, and an F1-score of 0.626. Macro-average metrics that assign equal weight to both classes produced a macro precision of 0.682, macro recall of 0.666, and macro F1-score of 0.601.

Table 5. Comparison of Performance of All Models

Model	Test Accuracy	Precision	Recall	F1-Score	Avg Inference Time
ConvNeXt-Nano	56.3%	0.624	0.617	0.562	126.94 ms/batch
Swin-Tiny	62.0%	0.672	0.670	0.620	30.10 ms/batch
ViT-Tiny	58.5%	0.707	0.665	0.578	14.36 ms/batch
MobileViT-XS	65.9%	0.703	0.706	0.659	24.66 ms/batch
EfficientNet-B0	60.2%	0.682	0.666	0.601	26.17 ms/batch

Source: (Research Results, 2025)

A comparative evaluation of five lightweight deep learning models for classifying borax-contaminated meatballs in Table 5 shows that MobileViT-XS stands out as the model with the highest performance, achieving an accuracy of 65.93%, surpassing Swin-Tiny (62.00%), EfficientNet-B0 (60.27%), ViT-Tiny (58.52%), and ConvNeXt-Nano (56.33%). The superiority of MobileViT-XS can be explained by its hybrid architecture, which combines local CNN convolutions and global Transformer attention, enabling detailed texture feature extraction and broad semantic context understanding.

This is reflected in the highest recall (0.706) and F1 score (0.659) achieved by this model, demonstrating consistency in balancing specific detection and contextual sensitivity. ViT-Tiny, despite having lower overall accuracy (58.52%), stands out with the highest precision (0.707), which is critical for minimizing false positives in hazardous contaminant detection scenarios. However, this compromise highlights the model's limitations in integrating high precision with comprehensive accuracy. On the other hand, ConvNeXt-Nano recorded the lowest performance (56.33%), confirming that adopting Transformer elements without systematic optimization does not improve detection performance, even after hyperparameter optimization with Optuna.

In terms of computational efficiency, ViT-Tiny is the fastest with an inference time of 14.36 ms/batch, followed by MobileViT-XS (24.66 ms/batch), EfficientNet-B0 (26.17 ms/batch), and Swin-Tiny (30.10 ms/batch). ConvNeXt-Nano shows inefficiency with an inference time of 126.94 ms/batch, in line with its low accuracy. MobileViT-XS stands out as the most balanced solution, combining high accuracy with competitive computational efficiency, making it suitable for deployment on low-power devices or real-time applications.

Overall, despite hyperparameter optimization across all models, MobileViT-XS consistently demonstrates superiority across all evaluation metrics. Model deployment recommendations depend on the context: MobileViT-XS for mobile or real-time applications, ViT-Tiny for systems prioritizing extreme precision with high speed, and Swin-Tiny for laboratory environments with adequate computational resources. The consistent performance of MobileViT-XS underscores its robustness in handling data variability, a critical aspect in food safety where classification errors can have significant impacts on public health.

CONCLUSION

This study evaluates five lightweight deep learning models for classifying images of meatballs containing borax: ConvNeXt-Nano, Swin-Tiny, ViT-Tiny, MobileViT-XS, and EfficientNet-B0. Hyperparameter optimization was conducted using Optuna, tuning parameters such as learning rate, batch size, and weight decay. The optimization yielded distinct optimal configurations for each model. Testing on 229 images revealed that MobileViT-XS achieved the best performance, withan accuracy of 65.93%, precision of 0.703, recall of 0.706, F1-score of 0.659, the lowest memory usage (45.94 MB), and a relatively high inference speed (24.66 ms per batch). This model effectively balances computational efficiency and classification quality due to its hybrid CNN–Transformer architecture and stable hyperparameter settings. Swin-Tiny ranked second, with an accuracy of 62.01% and the fastest inference speed (30.10 ms per batch), but it exhibited very high memory usage (451.03 MB). EfficientNet-B0 demonstrated a relatively balanced profile (60.27% accuracy, 79.01 MB memory), while ViT-Tiny achieved the highest precision (0.707) but lower overall accuracy (58.52%). Although ConvNeXt-Nano recorded the highest training accuracy (99.67%), it failed to generalize well, as indicated by its lowest test accuracy (56.33%). Overall, the results indicate that MobileViT-XS with optimized hyperparameters is the most suitable model for implementing a borax detection system on resource-limited devices such as smartphones. Future research should expand the dataset, evaluate the impact of extreme lighting conditions, and explore adaptive hyperparameter optimization techniques to enhance classification robustness in real-world scenarios.

REFERENCE

[1] B. Patwardhan and B. S. Paranthaman, “Nutrition, food and global health,” Oct. 01, 2021, Elsevier B.V. doi: 10.1016/j.jaim.2021.11.003.
[2] B. Patwardhan and B. S. Paranthaman, “Nutrition, food and global health,” Oct. 01, 2021, Elsevier B.V. doi: 10.1016/j.jaim.2021.11.003.
[3] L. S. Jakobsen, M. Al Huthiel, F. Al Natour, and A. Agudo, “Health consequences of harmful chemicals in foods: insights from the WHO Global Burden of Foodborne Disease Study,” Eur J Public Health, vol. 34, no. Supplement_3, p. ckae144.633, Nov. 2024, doi: 10.1093/eurpub/ckae144.633.
[4] M. M. Sarhan, S. E. Alotaibi, N. A. Alharbi, S. A. Aljohani, Y. A. Alnazzawi, and M. A. M. Alwadi,

"Exploring the contributing factors of fast food consumption in daily life: a qualitative study of Saudi university students," BMC Public Health, vol. 25, no. 1, p. 2402, Jul. 2025, doi: 10.1186/s12889-025-23624-0.
[5] D. Rahardiyan, "Fortifying bakso (Restructured meat product) with potential encapsulated functional strategies – A mini review," Feb. 01, 2021, Rynnye Lyan Resources. doi: 10.26656/fr.2017.5(1).277.
[6] L. Shao, S. Chen, H. Wang, J. Zhang, X. Xu, and H. Wang, "Advances in understanding the predominance, phenotypes, and mechanisms of bacteria related to meat spoilage," Trends Food Sci Technol, vol. 118, pp. 822–832, Dec. 2021, doi: 10.1016/j.tifs.2021.11.007.
[7] S. P. Mohanty et al., "The Food Recognition Benchmark: Using Deep Learning to Recognize Food in Images," Front Nutr, vol. 9, no. May, pp. 1–13, 2022, doi: 10.3389/fnut.2022.875143.
[8] S. Kaushal, D. K. Tammineni, P. Rana, M. Sharma, K. Sridhar, and H.-H. Chen, "Computer vision and deep learning-based approaches for detection of food nutrients/nutrition: New insights and advances," Trends Food Sci Technol, vol. 146, p. 104408, 2024, doi: https://doi.org/10.1016/j.tifs.2024.104408.
[9] A. A. S. Pradhana et al., "Sensor Array System Based on Electronic Nose to Detect Borax in Meatballs with Artificial Neural Network," Journal of Electrical and Computer Engineering, vol. 2023, pp. 1–10, Sep. 2023, doi: 10.1155/2023/8847929.
[10] A. Michael, S. Palelleng, I. Devi Damayanti, and J. Rusman, "Kombinasi Pretrained Model dan Random Forest Pada Klasifikasi Bakso Mengandung Boraks dan Non-Boraks Berbasis Citra," Teknika, vol. 12, no. 1, pp. 27–32, Feb. 2023, doi: 10.34148/teknika.v12i1.591.
[11] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," Remote Sens (Basel), vol. 13, no. 22, pp. 1–51, 2021, doi: 10.3390/rs13224712.
[12] Maurício, I. Domingues, and J. Bernardino, "Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review," May 01, 2023, MDPI. doi: 10.3390/app13095521.
[13] X. Meng, J. Ma, F. Liu, Z. Chen, and T. Zhang, "An Interpretable Breast Ultrasound Image Classification Algorithm Based on Convolutional Neural Network and Transformer," Mathematics, vol. 12, no. 15, Aug. 2024, doi: 10.3390/math12152354.
[14] A. Agarwal, A. Dash, G. Galbale, and S. P. Singh, "Analyzing Impact of Data Augmentation Techniques in Computer Vision," in 2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), IEEE, Feb. 2025, pp. 733–737. doi: 10.1109/CICTN64563.2025.10932445.
[15] R. Poojary, R. Raina, and A. Kumar Mondal, "Effect of data-augmentation on fine-tuned CNN model performance," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 10, no. 1, p. 84, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp84-92.
[16] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11966–11976, 2022, doi: 10.1109/CVPR52688.2022.01167.
[17] S. Woo et al., "ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders," in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 16133–16142. doi: 10.1109/CVPR52729.2023.01548.
[18] H. Li, H. Hu, Z. Jin, Y. Xu, and X. Liu, "The Image Recognition and Classification Model Based on ConvNeXt for Intelligent Arms," in 2025 IEEE 5th International Conference on Electronic Technology, Communication and Information (ICETCI), 2025, pp. 1436–1441. doi: 10.1109/ICETCI64844.2025.11084094.
[19] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," Aug. 2021, [Online]. Available: http://arxiv.org/abs/2103.14030
[20] S. H. Lee, S. Lee, and B. C. Song, "Vision Transformer for Small-Size Datasets," Dec. 2021, [Online]. Available: http://arxiv.org/abs/2112.13492
[21] A. Dosovitskiy et al., "AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE," [Online]. Available: https://github.com/
[22] S. Mehta and M. Rastegari, "MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer," Mar. 2022, [Online]. Available: http://arxiv.org/abs/2110.02178
[23] H. Ali, N. Shifa, R. Benlamri, A. A. Farooque,

and R. Yaqub, "A fine tuned EfficientNet-B0 convolutional neural network for accurate and efficient classification of apple leaf diseases," Sci Rep, vol. 15, no. 1, p. 25732, Jul. 2025, doi: 10.1038/s41598-025-04479-2.

[24] S. Tripathy, R. Singh, and M. Ray, "Automation of Brain Tumor Identification using EfficientNet on Magnetic Resonance Images," Procedia Comput Sci, vol. 218, pp. 1551–1560, 2023, doi: https://doi.org/10.1016/j.procs.2023.01.133.

[25] J. Pardede and A. S. Purohita, "Hyperparameter Search for CT-Scan Classification Using Hyperparameter Tuning in Pre-Trained Model CNN With MLP," in 2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM), 2022, pp. 1–8. doi: 10.1109/ICOSNIKOM56551.2022.10034878.

[26] Ştefana Duţă and A. E. Sultana, "Optimizing Depression Classification Using Combined Datasets and Hyperparameter Tuning with Optuna," Sensors, vol. 25, no. 7, p. 2083, Mar. 2025, doi: 10.3390/s25072083

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri