Zero : Jurnal Sains, Matematika, dan Terapan
E-ISSN : 2580-5754; P-ISSN : 2580-569X
Volume 10, Number 1, 2026
DOI: 10.30829/zero.v10i1.28735
Page: 224-236

Geometric Data Augmentation with a Two-Stage Fine-Tuning Strategy for EfficientNetB3-Based Fruit Condition Classification

[1]Hana Sajida Azhurra
Department of Mathematics, Ahmad Dahlan University, Yogyakarta, 55191, Indonesia

[2]Sugiyarto Surono
Department of Mathematics, Ahmad Dahlan University, Yogyakarta, 55191, Indonesia

[3]Aris Thobirin
Department of Mathematics, Ahmad Dahlan University, Yogyakarta, 55191, Indonesia

Article Info

Article history:

Accepted 25 March 2026

Keywords:

Deep Learning;
EfficientNetB3;
Fruit Condition Classification;
Geometric Data Augmentation;
Two-Stage Fine-Tuning.

ABSTRACT

Accurate fruit condition classification is essential for automated food safety assessment, particularly due to health risks associated with chemical contaminants such as formalin. However, reliable generalization in automated inspection systems remains challenging because limited visual variation in image datasets often leads to overfitting in deep learning models. To address this challenge, this study proposes an EfficientNetB3-based framework that integrates geometric data augmentation with a structured two-stage fine-tuning strategy to improve robustness and training stability. The proposed model achieved 99% test accuracy with consistent cross-dataset performance. The framework also demonstrated stable optimization behavior across training stages, indicating improved generalization capability. From a practical perspective, the proposed approach may support scalable food quality monitoring and automated sorting in agricultural supply chains, as well as preliminary food safety screening in large-scale inspection processes.

Corresponding Author:

Hana Sajida Azhurra,
Department of Mathematics,
Ahmad Dahlan University
Email: 2200015015@webmail.uad.ac.id

1. INTRODUCTION

Recent advances in data availability and computational power have significantly accelerated the development of deep learning methods for automated visual analysis, particularly in computer vision applications [1]. In the food industry, computer vision techniques are increasingly applied to support food quality and safety monitoring through automated inspection systems [2]. Digital fruit images containing variations in color, texture, and surface conditions have therefore become an important resource for automated classification tasks in food quality monitoring [3]. To effectively analyze complex visual characteristics such as fruit color and texture, deep learning, particularly Convolutional Neural Networks (CNNs), has become the dominant approach for extracting hierarchical visual features that are often difficult to detect through manual inspection [4][5][6].

Beyond quality assessment, identifying fruit conditions is critical for food safety, particularly regarding illegal formalin preservation [7][8]. Since contaminated fruits often mimic fresh products, manual inspection is frequently insufficient in resource-limited regions [9][10]. Consequently, automated visual inspection systems is essential to prevent hazardous products from entering the supply chain and to support scalable regulatory monitoring [11][12][13].

To effectively implement such automated visual inspection systems, model robustness and generalization are critical considerations. Limited visual variation in image datasets can lead to overfitting in CNN-based models, thereby reducing their ability to generalize to unseen data [14][15]. Geometric data augmentation is widely used to address this issue by introducing controlled spatial transformations that increase visual diversity without expanding the dataset size [16][17]. Additionally, when target datasets differ substantially from large-scale pretraining datasets such as ImageNet, careful adaptation strategies are required to ensure stable knowledge transfer [18]. A two-stage fine-tuning strategy can therefore be applied to gradually adapt pretrained feature representations to the target dataset while preserving useful pretrained features [19][20].

From an architectural perspective, EfficientNet represents a significant advancement in CNN architecture by achieving a balance between accuracy and computational efficiency through compound scaling of network depth, width, and resolution [21][22]. Among its variants, EfficientNetB3 provides a favorable trade-off between representational capacity and computational cost, making it suitable for image classification tasks with moderate dataset sizes [23]. Previous studies have reported strong performance of EfficientNetB3 in fruit image classification tasks, including apple and tomato recognition [24].

However, most previous studies have primarily focused on distinguishing visual quality differences between fresh and rotten fruit [25][26], leaving chemically treated samples and the integration of structured two-stage fine-tuning strategies underexplored. To address these gaps, this study proposes an EfficientNetB3-based fruit condition classification framework that integrates geometric data augmentation with a structured two-stage fine-tuning strategy to improve model generalization and training stability. The proposed framework is designed to facilitate automated visual screening of fruit conditions, including chemically treated samples, thereby supporting scalable food safety monitoring and automated sorting processes.

2. RESEARCH METHOD

This study employed a quantitative experimental design using a computational deep learning approach. The objective was to evaluate the performance of an EfficientNetB3-based classification model enhanced with geometric data augmentation and a two-stage fine-tuning strategy. The experimental framework was designed to ensure reproducibility through systematic data preprocessing, controlled training procedures, and standardized evaluation metrics. The method of this study consists of several steps, start from collecting the data, splitting the data, and preprocessing phases such as normalization, resizing, and applying geometric augmentation to the data. The EfficientNetB3 model is trained using a two-stage fine-tuning strategy, followed by a model evaluation. The complete workflow of this research is illustrated in Figure 1.

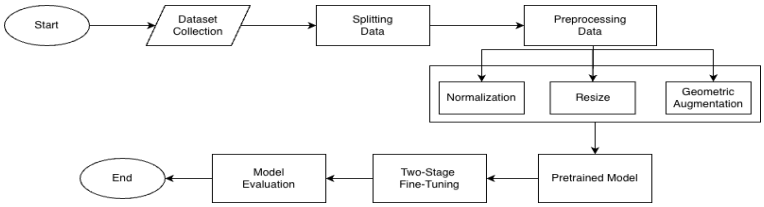

Figure 1. Research Flow

2.1 Dataset Collection

The dataset consisted of 10154 fruit images obtained from a public repository on Mendeley Data. The dataset included five fruit types: apples, bananas, oranges, grapes, and mangoes. Each fruit type was categorized into three condition classes: fresh, formalin-mixed, and rotten. Representative examples of the dataset are presented in Figure 2.

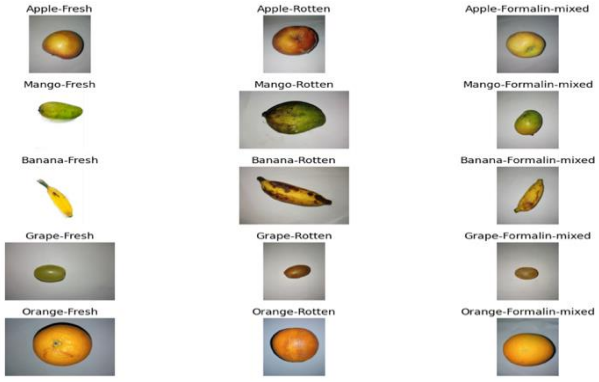

Figure 2. Sample fruit images by type and condition category

Although the primary dataset was obtained from a public repository [28], several methodological strategies were implemented to mitigate potential dataset limitations and to evaluate model robustness under environmental variability. According to the dataset documentation, the images were captured using multiple mobile cameras under varying environmental conditions, including differences in lighting, camera angles, image resolution, and weather conditions, which introduce natural variability in image acquisition [28]. The dataset metadata further provides quantitative indicators of acquisition diversity. Specifically, the images were captured using three different mobile devices (Apple iPhone 15 Pro Max, Redmi POCO M2 Reloaded, and Redmi Note 9 Pro), with image resolutions ranging approximately from 780 × 960 pixels to 3000 × 3000 pixels. Environmental metadata also indicates that images were acquired under different weather conditions (primarily sunny and foggy environments) with recorded temperatures ranging from approximately 24°C to 34°C across multiple collection locations in Sylhet, Bangladesh. These characteristics provide measurable evidence of environmental variability in terms of acquisition devices, illumination conditions, and image quality within the dataset. To further quantify visual variability within the dataset, additional image-level statistics were computed, including brightness, edge density, and entropy. Brightness provides a numerical indicator of lighting variation, while edge density and entropy quantify structural and background complexity within the images. The summary of these quantitative indicators is presented in Table 1.

Table 1. Quantitative Analysis of Image Characteristics Across Fruit Conditions

Condition	Brightness (mean ± std)	Edge Density (mean ± std)	Entropy (mean ± std)
Formalin-mixed	154.73 ± 11.04	0.004 ± 0.002	4.49 ± 0.15
Fresh	168.65 ± 35.60	0.003 ± 0.002	3.91 ± 1.20
Rotten	159.99 ± 17.73	0.011 ± 0.011	4.49 ± 0.66

Table 1 shows measurable differences in visual characteristics across fruit condition classes. Fresh fruit exhibits the highest mean brightness (168.65 ± 35.60), indicating relatively stronger lighting conditions and surface reflectance. Rotten fruit displays the highest edge density (0.011 ± 0.011), compared with formalin-mixed (0.004 ± 0.002) and fresh fruit (0.003 ± 0.002). This higher edge density reflects increased structural variability and surface irregularities, which are commonly associated with visual degradation in spoiled fruit such as wrinkles, dark spots, and uneven textures. In contrast, fresh fruit generally presents lower edge density due to smoother and more uniform surface characteristics. Entropy values further indicate variations in visual complexity within the images. Rotten and formalin-mixed samples exhibit higher entropy values (4.49 ± 0.66 and 4.49 ± 0.15) compared with fresh fruit (3.91 ± 1.20), suggesting more heterogeneous visual patterns caused by irregular textures, color variations, and local structural changes. In addition to lighting conditions and texture variability, viewpoint diversity was examined through an estimated camera angle distribution analysis in Table 2.

Table 2. Estimated Camera Angle Distribution

Camera Angle	Images	Percentage
Top view	102	85.0%
Oblique view	10	8.3%
Side view	8	6.7%

A random sample of 120 images (approximately 1% of the dataset) was selected to provide a representative subset while maintaining computational feasibility for manual inspection. For each sampled image, the fruit object was enclosed using a fruit bounding box, defined as the minimum rectangular region that fully contains the fruit in the image. The camera angle was approximated using the aspect ratio of the bounding box, computed as the ratio between the shorter and longer sides. Because most fruits exhibit approximately spherical shapes, top-view images tend to produce nearly circular projections with similar width and height values. Based on this geometric observation, heuristic thresholds were applied to categorize viewpoints: ratios greater than 0.90 indicate top-view images, ratios between 0.75–0.90 indicate oblique views, and ratios below 0.75 correspond to side views where the projected shape becomes increasingly elongated. The results show that approximately 85.0% of the images correspond to top-view angles, while 8.3% represent oblique views and 6.7% side views. Although the dataset is dominated by top-view images, the presence of oblique and side views indicates natural viewpoint variation within the image collection.

To further evaluate model generalization beyond the five fruit types used during training, an independent external dataset consisting of 100 fruit images was collected from heterogeneous sources, including publicly available web images and self-captured photographs obtained under a standardized distance of approximately 25 cm with a fixed top-down camera angle. This external dataset includes fruit varieties not present in the training data, such as pears, melons, and strawberries, enabling cross-variety evaluation of the model's ability to learn condition-related visual patterns rather than fruit-specific characteristics. Due to biosafety and regulatory

considerations associated with handling chemically treated food samples, the formalin-mixed class was not included in the external dataset. Preparing or handling fruit samples containing formalin requires controlled laboratory conditions and certified chemical safety procedures, which were beyond the scope of this study. Therefore, the external evaluation focuses on visually observable conditions (fresh and rotten), which represent the most common quality degradation patterns encountered in retail markets and agricultural supply chains.

Importantly, the external dataset also enables an experimental comparison across environments, as the images originate from heterogeneous acquisition conditions different from the primary dataset. The inclusion of web-sourced images and independently captured photographs introduces variations in background composition, lighting conditions, and camera characteristics. Evaluating the trained model on this external dataset therefore provides an indirect assessment of model robustness under diverse environmental conditions beyond the original training distribution. To ensure evaluation independence, the external dataset was manually screened to avoid any overlap with the primary dataset used for training, validation, and testing. Together, geometric data augmentation, controlled hyperparameter selection, and cross-dataset evaluation serve as mitigation strategies to reduce dataset bias and assess the robustness of the proposed model.

2.2 Data Splitting Procedure

The fruit dataset was split in two steps. Initially, 90% of the data was allocated to the training and validation sets, and the other 10% was allocated to the test set. Out of the 90% allocated to the training and validation sets, 80% was allocated to the training set and 20% was allocated to the validation set. This resulted in a split of 72% training, 18% validation, and 10% testing. This method has a final count of 7310 images for training, 1828 images for validation, and 1016 images for testing. Splitting in two steps hierarchically ensures class balance is preserved for all subsets of the data to maintain balance and proportion. It should be noted that this data splitting procedure applies only to the primary dataset. The external dataset used for cross-dataset evaluation was kept entirely separate and was not included in the training, validation, or testing partitions.

2.3 Pre-processing

The images in the training set were preprocessed to align with the specifications of EfficientNetB3 and to ensure stable training. The preprocessing process involved several steps. First, the images were resized to match the model's required input resolution. Next, geometric data augmentation was applied, followed by data normalization.

a. Resize

All images were resized to 300 × 300 pixels to conform to the input resolution required by the EfficientNetB3 architecture [27]. This resizing step ensured uniform input dimensions across the dataset and enabled the model to process the images consistently during training and evaluation. Standardizing the image size also facilitated optimal feature extraction by aligning the spatial resolution of the input data with the pretrained network configuration.

b. Geometric Data Augmentation

Geometric transformations were applied to the training images to increase visual diversity without altering the class labels [28]. These transformations did not increase the total number of images in the dataset, as they were generated on-the-fly during the training process through the data loader mechanism, meaning that no additional images were permanently stored. Consequently, the total number of images remained constant, while visual variability was continuously introduced through random transformations at each training iteration. To improve the model's ability to generalize to variations in real-world image conditions, geometric augmentation was used to modify the position, scale, and rotation of objects within the images [29]. By introducing controlled spatial transformations, this strategy helps mitigate spatial environmental bias by exposing the model to diverse object orientations, positions, and scales. As a result, the model becomes less sensitive to camera angle and framing differences while preserving semantic consistency across classes. The parameter values and application probabilities used for geometric data augmentation are summarized in Table 3.

Table 3. Geometric Data Augmentation Parameters and Probability Values

Transformation	Parameters	Probability
Horizontal Flip	-	0.5
Vertical Flip	-	0.3
Random Rotate 90	-	0.3
Shift Scale Rotate	shift_limit = 0.05, scale_limit = 0.10, rotate_limit = 20°	0.5

The probability values for Horizontal Flip, Vertical Flip, and Random Rotate 90 indicate how frequently each transformation is applied to the images during training [30]. The Horizontal Flip operation mirrors the image along the vertical axis, while the Vertical Flip mirrors the image along the horizontal axis. Horizontal Flip was assigned a higher probability (p = 0.5) because horizontal orientation changes commonly occur in real-world image acquisition. In contrast, Vertical Flip was applied with a lower probability (p = 0.3) to avoid generating unrealistic object orientations that rarely occur in natural fruit photography. These transformations help expose the model to different object orientations while preserving the semantic structure of the fruit images. The Random Rotate 90 transformation randomly rotates the image by multiples of 90 degrees, introducing additional orientation variability without distorting object geometry. In the Shift Scale Rotate transformation, the parameters control the maximum limits of positional shift, scaling variation, and rotation angle that can be randomly applied to the images during training. The shift limit of 0.05 allows the object position to vary within 5% of the image dimensions, simulating small camera framing variations commonly observed during image capture. The scale limit of 0.10 enables moderate object size variation to account for differences in camera distance while avoiding unrealistic magnification or shrinking of the fruit object. The rotation limit of 20° introduces moderate orientation variability while preserving the natural appearance of the fruit images, since extreme rotations are less common in typical fruit photography settings. The hyperparameter values for these geometric transformations were determined based on preliminary experiments and commonly adopted practices in image-based deep learning studies. This configuration aims to introduce sufficient spatial variability while preserving realistic fruit geometry and preventing excessive distortions that could negatively affect model learning.

c. Data Normalization

Each image pixel value was normalized using mean values of (0.485, 0.456, 0.406) and standard deviation values of (0.229, 0.224, 0.225), following the preprocessing scheme applied to the ImageNet dataset [31]. Z-score normalization was employed, where each pixel value was transformed as shown in (1) [32]:

$$X_{new} = \frac{X - \mu}{\sigma} \tag{1}$$

Where X_{new} represented the normalized pixel value, while X denoted the original pixel value. The symbol μ referred to the mean value of each color channel (R, G, B), and σ represented the standard deviation of each channel. To ensure that each pixel followed the distribution defined by the mean and standard deviation of the EfficientNetB3 input, normalization was applied independently to each channel.

2.4 Pre-trained Model

This study utilized a pretrained model to facilitate the extraction of more complex features and enhance visual pattern understanding within the dataset. The selected model, EfficientNetB3, consisted of multiple convolutional layers and MBConv blocks arranged in various configurations. This architecture employed a compound scaling technique that proportionally adjusted the network depth, width, and input resolution, resulting in more efficient feature extraction compared to conventional CNN architectures [21]. The incorporation of Squeeze-and-Excitation (SE) modules within the MBConv blocks enhanced the model's sensitivity to intricate visual variations by adaptively recalibrating channel-wise feature responses [33]. With these pretrained feature representations, EfficientNetB3 provided a strong initialization prior to the application of the two-stage fine-tuning process.

2.5 Two-Stage Fine-Tuning

The model training process employed a two-stage fine-tuning approach to ensure more stable adaptation to the characteristics of the target dataset. Training was conducted using the AdamW optimizer together with an early stopping mechanism to prevent overfitting when the validation performance no longer improved [34]. In Stage-1, the entire EfficientNetB3 backbone (feature extractor) was frozen, allowing weight updates to occur only in the classifier layer, which had been modified to output three classes corresponding to the fruit condition categories in the target dataset. At this stage, parameter updates conceptually followed the fundamental gradient-based optimization rule [35], expressed in (2) and (3):

$$x'_{backbone} = x_{backbone} \tag{2}$$

$$x'_{cls} = x_{cls} - \eta_1 \nabla_{x_{cls}} \tag{3}$$

From equation (2), $x'_{backbone}$ denoted the backbone parameter value after an optimizer step, while $x_{backbone}$represented the original backbone parameter. This indicated that no updates were applied to the backbone parameters. From (3), only the classifier parameters were optimized using a learning rate of 1×10^{-3}, where x'_{cls} denoted the updated classifier parameters after the optimization step, η_1 represented the learning rate in Stage-1, and $\nabla_{x_{cls}} L$ denoted the gradient of the loss function with respect to the classifier parameters.

Freezing the backbone in Stage-1 allowed the classifier layers to adapt to the target data without altering the general visual features learned during ImageNet pretraining. In Stage-2, all EfficientNetB3 layers were unfrozen so that the model could learn feature representations more specific to the target dataset. Training at this stage employed a smaller learning rate of 1×10^{-4}. Parameter updates continued to follow the fundamental gradient-based optimization rule [35], and the update formulation became (4):

$$x'_{cls} = x_{cls} - \eta_2 \nabla_{x_{cls}} L \tag{4}$$

Here, η_2 represented the learning rate in Stage-2, where $\eta_2 < \eta_1$. The use of a smaller learning rate prevented excessively large parameter updates and helped preserve useful feature representations learned during pretraining. This two-stage strategy effectively refines the model's feature extraction capability while ensuring stable classification across the three categories. By initially freezing the backbone, parameter updates are constrained to the classifier layer, which regulates early-stage adaptation and stabilizes the learning dynamics. This approach prevents abrupt shifts in the learned feature space and preserves pretrained representations. Such a controlled mechanism promotes stable convergence before transitioning to Stage-2, where deeper feature refinement is performed under a reduced learning rate.

In addition to the training strategy described above, several training hyperparameters were specified to ensure stable optimization during the fine-tuning process. The model was trained using the AdamW optimizer with a batch size of 8 and a weight decay coefficient of 1×10^{-5} to improve regularization and reduce the risk of overfitting. The learning rates applied in Stage-1 and Stage-2 correspond to the values defined in the optimization formulations (η_1 and η_2) described earlier. Training was conducted for 10 epochs in Stage-1 and up to 5 epochs in Stage-2, with an early stopping mechanism applied to terminate training when the validation performance no longer improved. These hyperparameter settings help maintain stable gradient updates while allowing gradual adaptation of the pretrained EfficientNetB3 representations to the target dataset.

2.6 Model Evaluation

The model was evaluated using the test set, which consisted of images that were not used during the training or validation phases. Various evaluation metrics and a confusion matrix were employed to assess the model's performance. The confusion matrix comprised four components: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). These components enabled the assessment of prediction correctness and error distribution. Evaluation metrics were computed based on the TP, FP, and FN values for each class k, denoted as TP_k, FP_k, and FN_k. The formulas for precision and recall are presented in (5) and (6) [36]:

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} \tag{5}$$

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \tag{6}$$

Where $k = 1,2,3$. To obtain a more balanced evaluation between precision and recall [36], the F1-score was calculated as shown in (7):

$$\text{F1}_k = 2 \times \frac{\text{Precision}_k \times \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{7}$$

Accuracy was defined as the ratio of correctly predicted samples to the total number of evaluated instances in multi-class classification. Accordingly, accuracy was calculated as shown in (8):

$$\text{Accuracy} = \frac{\sum_{k=1}^{3} \text{TP}_k}{N} \tag{8}$$

Where TP_k denotes the number of correct predictions for class k, and N represents the total number of samples in the test set. This formulation is consistent with the general definition of accuracy as the proportion of correct predictions over the entire dataset [37], thereby enabling a fair evaluation of the model's performance in multi-class classification.

2.7 Software and Hardware Environment

The proposed model was implemented in Python using the PyTorch deep learning framework, with experiments conducted in Visual Studio Code. Training was performed on a laboratory workstation equipped with an NVIDIA CUDA-enabled GPU (16 GB memory) to accelerate computation. Stage-1 fine-tuning was conducted for 10 epochs (approximately 2.5 minutes per epoch), while Stage-2 fine-tuning required 5 epochs (approximately 6.5 minutes per epoch). Overall, the complete two-stage training process required approximately one hour. These results indicate that the EfficientNetB3-based two-stage fine-tuning approach is computationally feasible for research-scale implementation without requiring high-end infrastructure.

3. RESULT AND ANALYSIS

This section presents the results and discussion obtained from the series of experiments conducted throughout the study. The content is arranged systematically to illustrate the model's workflow from training to final evaluation. This section is structured into five main components: the two-stage fine-tuning training strategy, comparative performance analysis, augmented versus non-augmented evaluation, external validation for generalization assessment, and discussion of ethical and technical limitations. Each part is supported by relevant figures and tables to provide a comprehensive understanding of the findings.

3.1 Two-Stage Fine-Tuning

The EfficientNetB3 architecture was trained using a two-stage fine-tuning strategy to obtain more optimal feature representations. This approach divided the weight-adjustment process into two stages, allowing the model to gradually learn from general features to more specific visual patterns in the fruit images.

a. Stage-1 Fine-Tuning

In the first stage, as described in the methodology, only the classifier layer was trained, allowing the model to adapt gradually to the fundamental characteristics of the dataset. Figure 3 illustrates the progression of training and validation accuracy during Stage-1 fine-tuning, where only the classifier layer was updated while the pretrained backbone remained frozen. A gradual upward trend is observed in both curves, indicating stable adaptation of the classification layer to the fruit dataset.

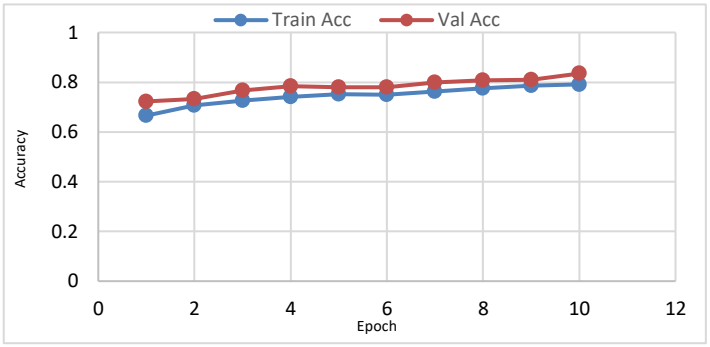

Figure 3. Training and validation accuracy during Stage-1 Fine-Tuning

At epoch 1, training and validation accuracy were 0.6656 and 0.7226, respectively, indicating that pretrained features already provided meaningful representations. From epoch 1 to 4, both metrics increased steadily, reflecting effective adaptation of the classifier layer. Because only a limited number of parameters were updated, training remained stable without abrupt fluctuations. Between epochs 5 and 10, accuracy continued to improve gradually, reaching 0.7912 (training) and 0.8348 (validation). The consistent upward trend and absence of oscillations indicate stable optimization. Validation accuracy remained slightly higher than training accuracy across most epochs, suggesting that Stage-1 fine-tuning did not induce overfitting and maintained good generalization despite limited parameter updates.

Figure 4 presents the training and validation loss curves during Stage-1 fine-tuning. The training loss exhibits a steady downward trend across epochs, decreasing from approximately 0.9355 in the first epoch to 0.5253 at epoch 10. This consistent reduction indicates that the classifier layer increasingly improved prediction confidence and reduced classification error.

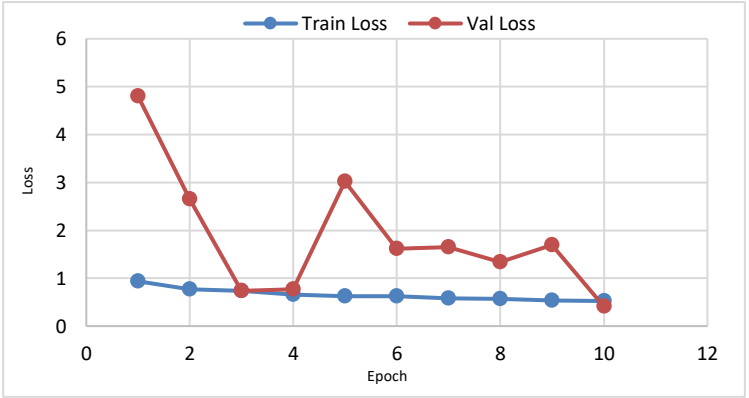

Figure 4. Training and validation loss during Stage-1 Fine-Tuning

The validation loss curve exhibits noticeable fluctuations during Stage-1. Initially, validation loss was relatively high before decreasing substantially to 0.7673 by epoch 4. Although a discernible gap between training and validation accuracy is observed, this discrepancy does not progressively widen across epochs, nor is it associated with a sustained increase in validation loss. Both accuracy metrics demonstrate consistent upward trends, while validation loss ultimately declines to 0.4098 by epoch 10, closely following the downward trajectory of the training loss. The temporary spike around epoch 5 likely reflects the constrained adaptability of the classifier under frozen backbone parameters. With feature representations fixed, the classifier layer adjusts decision boundaries within a limited feature space, which may result in transient instability when processing visually ambiguous samples. Importantly, the absence of sustained divergence between training and validation loss suggests that Stage-1 did not induce progressive overfitting. Instead, it can be interpreted as a controlled adaptation phase that stabilizes the classifier before full-network fine-tuning. From a transfer learning perspective, this boundary-level adjustment helps preserve pretrained representations while enabling gradual task-specific adaptation, thereby providing a stable foundation for Stage-2 refinement. Nevertheless, to further minimize potential overfitting risks, several regularization strategies could be considered in future implementations, such as increasing augmentation intensity, incorporating dropout in the classifier layer, adjusting weight decay parameters, or refining early stopping sensitivity. These approaches may provide additional control over training dynamics during early-stage adaptation.

b. Stage-2 Fine-Tuning

Figure 5 illustrates the progression of training and validation accuracy across the five epochs of Stage-2 fine-tuning. A consistent upward trend is observed in both curves, indicating stable optimization after unfreezing the backbone layers.

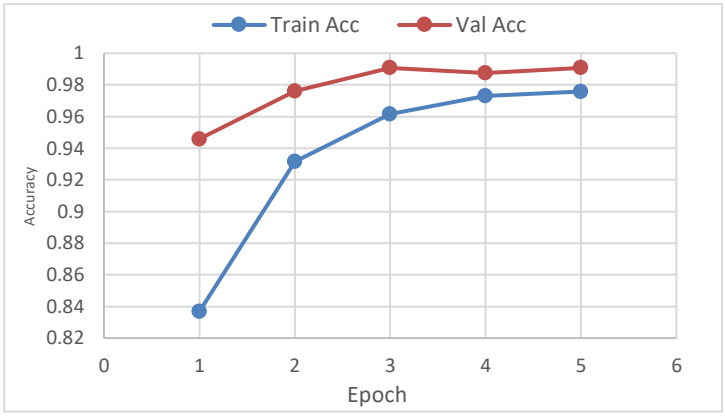

Figure 5. Training and validation accuracy during Stage-2 Fine-Tuning

At epoch 1, training and validation accuracy were 0.8348 and 0.9450, indicating strong pretrained feature representations from Stage-1. After unfreezing the backbone, training accuracy increased sharply to 0.9316 at epoch 2, reflecting rapid feature adaptation. From epoch 2 to 5, both metrics improved gradually, reaching 0.9758 (training) and 0.9907 (validation). The small final gap between training and validation accuracy suggests minimal overfitting and strong generalization. The smooth convergence pattern further indicates that the reduced learning rate in Stage-2 effectively stabilized parameter updates while refining deeper feature representations.

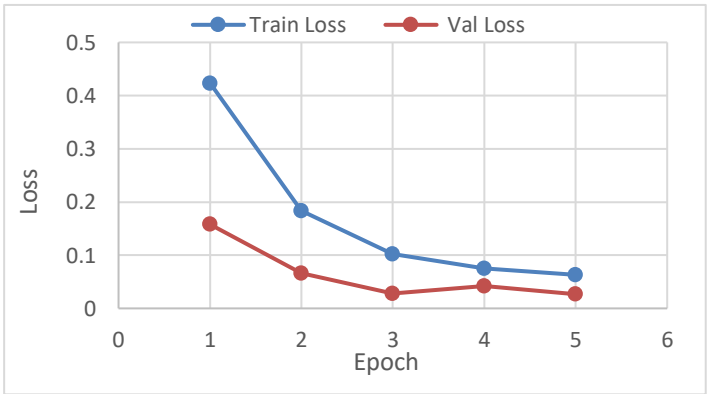

Figure 6. Training and validation loss during Stage-2 Fine-Tuning

Figure 6 illustrates the training and validation loss during Stage-2 fine-tuning, both exhibiting a consistent downward trend that confirms stable optimization. Training loss decreased sharply from 0.4236 to 0.1832 between epochs 1 and 2 after backbone unfreezing, indicating rapid early adaptation. Loss values continued to decline steadily, reaching 0.0631 (training) and 0.0267 (validation) by epoch 5. The small gap between training and validation loss, along with the absence of divergence, demonstrates minimal overfitting and stable parameter updates under the reduced learning rate. When interpreted alongside the increasing accuracy (Figure 5), the decreasing loss confirms improved prediction confidence, reduced classification error, and enhanced generalization performance. Overall, Stage-2 effectively refined feature representations, with rapid early convergence followed by smooth stabilization, highlighting the benefit of gradual unfreezing from the pretrained EfficientNetB3 backbone. A summary comparison of the training results from both stages is presented in Table 4.

Table 4. Final result of Two-Stage Fine-Tuning

Stage of Fine-Tuning	Epoch	Train Acc	Train Loss	Val Acc	Val Loss
Stage-1	10	0.7912	0.5253	0.8348	0.4098
Stage-2	5	0.9758	0.0631	0.9907	0.0267

Training accuracy increased from 0.7912 in Stage-1 to 0.9758 in Stage-2, while validation accuracy improved from 0.8348 to 0.9907. Concurrently, training loss decreased from 0.5253 to 0.0631, and validation loss decreased from 0.4098 to 0.0267, indicating more stable optimization and improved convergence during Stage-2. The small gap between training and validation accuracy suggests minimal overfitting and strong generalization performance. These findings confirm that gradual unfreezing combined with a reduced learning rate effectively stabilized parameter updates and enhanced deeper feature adaptation, thereby improving discriminative capability and overall generalization.

3.2 Model Accuracy Comparison

This section compares the performance of the augmented model with that of the non-augmented model to evaluate the impact of geometric data augmentation on overall classification performance. Both models were trained using the same two-stage fine-tuning strategy to ensure that performance differences were primarily attributed to the use of augmentation. Table 5 presents the accuracy comparison between the two models.

Table 5. Model Accuracy of Augmented and Non-Augmented Model

Use of Augmentation	Train Accuracy	Validation Accuracy	Test Accuracy
Augmentation	0.97	0.99	0.99
Non-Augmentation	0.98	0.97	0.98

The slightly lower training accuracy in the augmented model is expected because geometric data augmentation is applied only during training, increasing data variability and making the learning task more challenging. This acts as a regularization mechanism that encourages the model to learn more generalizable features rather than memorizing training patterns. Consequently, despite slightly lower training accuracy, the augmented model achieves higher validation and test performance than the non-augmented model.

To further examine whether the performance difference is statistically meaningful, the experiment was repeated five times under identical training configurations. The augmented model achieved a mean test accuracy of 0.9903 ± 0.0006, whereas the non-augmented model achieved 0.9815 ± 0.0043. A Welch's t-test confirmed that the difference between the two models is statistically significant ($p = 0.01$), indicating that the improvement obtained through geometric data augmentation is unlikely to be caused by random variation. The mean accuracy improvement was 0.0088 with a 95% confidence interval of [0.0034, 0.0142]. Moreover, the lower standard deviation observed in the augmented model suggests improved training stability and robustness across repeated runs Overall, the augmented model consistently achieved higher validation and test accuracy (0.99 and 0.99, respectively) compared to the non-augmented model (0.97 and 0.98). Although the numerical improvement in test accuracy is approximately 1%, this corresponds to about ten additional correctly classified samples in the test set, demonstrating a practically meaningful performance gain.

3.3 Model Performance Evaluation

The model's performance was further evaluated on the test set using precision, recall, and F1-score metrics for each class. Table 6 presents the comparison between the augmented and non-augmented models.

Table 6. Precision, Recall, and F1-Score Comparison

Class	Precision		Recall		F1-Score	
	Aug.	Non-Aug.	Aug.	Non-Aug.	Aug.	Non-Aug.
Formalin-mixed	0.98	0.99	1.00	0.97	0.99	0.98
Fresh	0.99	0.97	0.99	0.99	0.99	0.98
Rotten	1.00	0.98	0.98	0.97	0.99	0.98

The results in Table 6 indicate that the augmented model outperforms the non-augmented model in precision, recall, and F1-score, indicating the benefit of geometric data augmentation in improving classification stability. This improvement is further supported by the confusion matrix shown in Figure 7.

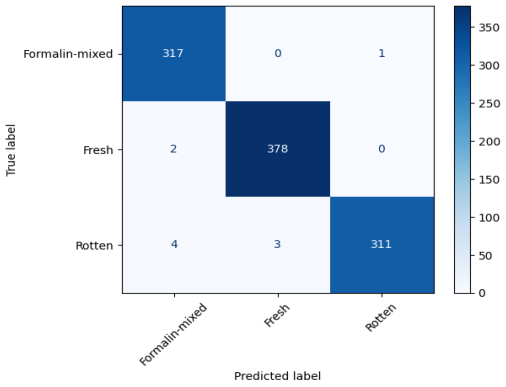

Figure 7. Confusion matrix

Figure 7 presents the confusion matrix analysis. Out of 1,016 test samples, 1,006 were correctly classified, yielding an overall accuracy of approximately 99.01%. The formalin-mixed and fresh classes achieved high class accuracies of 99.69% and 99.47%, respectively, with minimal misclassification. The rotten class showed slightly lower performance at 97.80%, with most errors occurring between rotten and formalin-mixed samples, likely due to overlapping visual features such as discoloration or surface irregularities. The macro-average accuracy reached 98.99%, indicating balanced performance across categories. These results align with the precision, recall, and F1-score values and confirm strong feature discrimination and generalization capability on unseen data.

3.4 External Validation for Generalization Assessment

To address the concern regarding cross-dataset validation and to assess robustness under diverse real-world conditions, the trained model was evaluated using an independent external dataset. The detailed classification performance is presented in Table 7.

Table 7. External Dataset Classification Results

Class	Total Samples	Correct Predictions	Accuracy
Fresh	50	50	1.00
Rotten	50	49	0.98
Overall	100	99	0.99

Beyond environmental variation, the external validation also implicitly evaluates cross-varietal generalization. Although the training dataset was limited to five fruit types (apples, bananas, oranges, grapes, and mangoes), the external dataset included fruit varieties not present during training, such as pears, melons, and strawberries. The model's consistent performance across these unseen fruit types suggests that the learned feature representations capture condition-related characteristics (fresh vs. rotten) rather than fruit-specific textures alone. This cross-variety evaluation partially mitigates concerns regarding limited dataset diversity and provides preliminary evidence of broader applicability. As shown in Table 7, the model achieved perfect classification performance for the Fresh class, correctly identifying all 50 samples (100% accuracy). For the Rotten class, one misclassification was observed, resulting in an accuracy of 98%. Overall, the model achieved an external accuracy of 99% across 100 independent samples. These results indicate that the proposed model maintains high classification performance when applied to previously unseen data collected under varied environmental conditions. The consistent performance across both classes suggests that the learned feature representations generalize well beyond the distribution of the primary dataset.

3.5 Discussion on Ethical and Technical Limitations

While this study represents a research-level prototype, ethical considerations must be addressed before real-world deployment. In food safety applications, misclassification may have significant consequences. False negatives (misclassifying contaminated fruit as safe) could allow hazardous products to reach consumers, posing potential health risks and undermining trust in automated inspection systems. Conversely, false positives (rejecting safe fruit) may lead to unnecessary economic losses and supply chain inefficiencies. Therefore, the proposed model should be regarded as a decision-support tool rather than a fully autonomous system, and its predictions should be complemented by laboratory testing or expert verification to support responsible and trustworthy deployment in food safety monitoring. Beyond these immediate operational considerations, the

proposed approach may also have broader economic and regulatory implications for food safety monitoring. Automated visual inspection systems have the potential to reduce reliance on labor-intensive manual inspection and improve scalability in large agricultural supply chains, thereby lowering operational costs and enabling earlier identification of potentially contaminated products. Early detection may help reduce downstream economic losses associated with large-scale product recalls, disposal of unsafe food, and reputational damage to producers and distributors. However, regulatory authorities typically require rigorous validation and regulatory compliance before automated systems can be incorporated into official food safety inspection protocols. In practice, chemical contamination detection is still primarily verified through laboratory-based analytical methods due to their established reliability and legal acceptance. Consequently, computer vision–based systems such as the proposed model should currently be considered complementary screening tools rather than replacements for laboratory verification.

Compared with previous computer vision studies on fruit quality assessment, several deep learning approaches have been proposed for detecting fresh and rotten fruits. For example, Sharma and Kumar [40] investigated a ResNet50 model to automatically classify fruit freshness, achieving an overall accuracy of 95%. Similarly, Reka et al. [26] applied a VGG16 model for rotten fruit detection and reported an accuracy of 95%. While these studies primarily focus on visual freshness or surface spoilage, the present study extends the scope to chemically treated fruit conditions, particularly formalin-mixed samples. In addition to classification performance, the proposed EfficientNetB3-based framework achieved a test accuracy of 99%, demonstrating competitive performance while also incorporating statistical validation, dataset variability analysis, and discussion of deployment-related considerations in real-world food safety monitoring scenarios. The improved performance of EfficientNetB3 compared with architectures such as VGG16 and ResNet50 can be attributed to its compound scaling strategy, which systematically balances network depth, width, and input resolution to enhance feature representation while maintaining computational efficiency. This architectural design enables the model to capture finer visual patterns and subtle texture variations in fruit images that may not be effectively represented by conventional CNN architectures. These comparisons indicate that the proposed EfficientNetB3-based framework achieves competitive performance while addressing a less explored problem in computer vision-based food safety monitoring. However, direct accuracy comparisons across studies should be interpreted cautiously due to differences in datasets, class distributions, and experimental settings.

Despite these contributions, several practical limitations should be considered when interpreting the results. Although the experimental setup employed a balanced dataset to support stable learning across classes, real-world food safety scenarios are unlikely to exhibit perfectly balanced distributions. Formalin contamination is expected to occur far less frequently than fresh or naturally rotten fruit, which may introduce frequency bias and increase the risk of undetected contamination. In practical deployment, maintaining high recall for the formalin-mixed category would therefore be particularly important due to its direct public health implications. Future implementations may address this challenge through cost-sensitive learning strategies, such as weighted cross-entropy loss, or through data-level techniques including targeted augmentation and controlled resampling.

4. CONCLUSION

This study proposed an EfficientNetB3-based framework for fruit condition classification that integrates geometric data augmentation and a two-stage fine-tuning strategy. The proposed approach achieved 99% test accuracy and demonstrated robust performance under visual variability while maintaining stable generalization. These findings suggest that the framework has potential as a scalable visual screening tool to support automated food quality inspection and preliminary food safety monitoring. Future work may explore several experimental directions to enhance the practical applicability of the proposed framework. First, to improve variety-level generalization, the dataset may be expanded to include additional fruit varieties, such as dragon fruit and durian, by integrating larger public fruit image repositories including Fruits-360. Second, to better represent real-world environmental variability, further evaluations may incorporate systematic lighting-controlled experiments ranging from 200 to 1000 lux as well as testing under dynamic backgrounds commonly observed in retail conveyor-based sorting environments. Third, the issue of class imbalance in formalin-treated samples may be addressed through cost-sensitive learning strategies, such as the adoption of Focal Loss to improve minority-class recall. Finally, future studies may investigate the feasibility of real-time deployment by evaluating the computational efficiency and inference latency of the proposed model under continuous image streams, such as camera-based fruit inspection systems. This may include performance benchmarking under high-frame-rate input scenarios and assessing the model's suitability for integration into automated visual inspection systems used in food quality monitoring.

5. REFERENCES

[1] N. L. Rane, M. Paramesha, S. P. Choudhary, and J. Rane, "Machine Learning and Deep Learning for Big Data Analytics: A Review of Methods and Applications," *Partners Universal International Innovation Journal*, vol. 2, no. 3, pp. 172–197, 2024, doi: 10.5281/zenodo.12271006.

[2] Y. E. Nagaty, "Digital Image Analysis (DIA) in Food Technology: An Overview," *Alexandria Journal of Food Science and Technology*, vol. 21, no. 2, pp. 29–34, 2024, doi: 10.21608/ajfs.2025.340104.1062.

[3] A. Bhargava and A. Bansal, "Fruits and vegetables quality evaluation using computer vision: A review," *Journal of King Saud University – Computer and Information Sciences*, vol. 33, no. 3, pp. 243–257, 2021, doi: 10.1016/j.jksuci.2018.06.002.

[4] I. H. Sarker, "Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, Art. no. 420, 2021, doi: 10.1007/s42979-021-00815-1.

[5] F. Badri, M. T. Alawiy, and E. M. Yuniarno, "Deep learning architecture based on convolutional neural network (CNN) in image classification," *Jurnal Ilmiah KURSOR*, vol. 12, no. 2, pp. 83–92, 2023, doi: 10.21107/kursor.v12i2.349.

[6] M. W. Sari, S. P. Sitorus, Rohani, and R. Pane, "Implementation of Convolutional Neural Network (CNN) Method in Determining the Level of Ripeness of Mango Fruit Based on Image," *Jurnal Penelitian Pendidikan IPA*, vol. 11, no. 5, pp. 419–428, 2025, doi: 10.29303/jppipa.v11i5.11436.

[7] M. Leslie, J. Vermaat, L. Haak, et al., "A FoodSafeR Perspective on Emerging Food Safety Hazards and Associated Risks," *Frontiers in Sustainable Food Systems*, vol. 9, 2025, Art. no. 1646792, doi: 10.3389/fsufs.2025.1646792.

[8] U. Agustina, R. M. Sari, M. V. Humairo, and E. O. P. Dewi, "Food Safety Monitoring: Formaldehyde Health Risk Assessment on Imported Fruits in Indonesia 2014-2022," *Journal of Global Research in Public Health*, vol. 8, no. 2, pp. 206–215, 2023, doi: 10.30994/jgrph.v8i2.477.

[9] H. Xu, H. Chen, Y. Li, T. Luo, D. Zhao, X. Chen, H. Zhang, X. Hu, H. Xu, Y. Wang, Y. Shentu, and Z. Tong, "Dietary formaldehyde: a silent aggravator of diabetes and cognitive impairments," *Nutrition & Diabetes*, vol. 15, Art. no. 35, 2025, doi: 10.1038/s41387-025-00390-x.

[10] A. Sattar, M. A. M. Ridoy, A. K. Saha, H. M. Hasan Babu, and M. N. Huda, "Computer vision based deep learning approach for toxic and harmful substances detection in fruits," *Heliyon*, vol. 10, no. 3, Art. no. e25371, 2024, doi: 10.1016/j.heliyon.2024.e25371.

[11] Y. K. Anagaw, W. Ayenew, L. W. Limenh, D. T. Geremew, M. C. Worku, T. A. Tessema, W. Simegn, and M. L. Mitku, "Food adulteration: Causes, risks, and detection techniques—review," *SAGE Open Medicine*, vol. 12, pp. 1–10, 2024, doi: 10.1177/20503121241250184.

[12] J. Rodrigues, C. Saraiva, J. García-Díez, J. Castro, and A. Esteves, "Evaluating the Effectiveness of Food Safety Policies in Portugal: A Stakeholder-Based Analysis of Challenges and Opportunities for Food Safety Governance," *Foods*, vol. 14, no. 9, Art. no. 1534, 2025, doi: 10.3390/foods14091534.

[13] I. Rojas Santelices, P. Gupta, C. Mallor, and M. Chandra, "Artificial vision systems for fruit inspection and classification," *Sensors*, vol. 25, no. 5, Art. no. 1524, 2025, doi: 10.3390/s25051524.

[14] K. Shehzad, U. Ali, and A. Munir, "Computer Vision for Food Quality Assessment: Advances and Challenges," *Global Journal of Machine Learning and Computing*, vol. 1, no. 1, pp. 76–92, 2025, doi: 10.70445/gjmlc.1.1.2025.76-92.

[15] A. Salari, A. Djavadifar, X. Liu, and H. Najjaran, "Object recognition datasets and challenges: A review," *Neurocomputing*, vol. 495, pp. 129–152, 2022, doi: 10.1016/j.neucom.2022.01.022.

[16] H. K. Dishar and L. A. Muhammed, "A Review of the Overfitting Problem in Convolution Neural Network and Remedy Approaches," *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 15, no. 2, pp. 155–165, 2023, doi: 10.29304/jqcm.2023.15.2.1240.

[17] A. Larasati, S. Surono, A. Thobirin, and D. A. Dewi, "Performance analysis of resampling techniques for overcoming data imbalance in multiclass classification," *JUITA: Jurnal Informatika*, vol. 13, no. 1, pp. 57–66, Mar. 2025, doi: 10.30595/juita.v13i1.25270.

[18] T. Islam, M. S. Hafiz, J. R. Jim, M. M. Kabir, and M. F. Mridha, "A systematic review of deep learning data augmentation in medical imaging: Recent advances and future research directions," *Healthcare Analytics*, vol. 5, Art. no. 100340, 2024, doi: 10.1016/j.health.2024.100340.

[19] H. Yuan, M. Zhu, R. Yang, H. Liu, I. Li, and C. Hong, "Rethinking Domain-Specific Pretraining by Supervised or Self-Supervised Learning for Chest Radiograph Classification: A Comparative Study Against ImageNet Counterparts in Cold-Start Active Learning," *Health Care Science*, vol. 4, no. 2, pp. 110–143, 2025, doi: 10.1002/hcs2.70009.

[20] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in Proc. *56th Annu. Meet. Assoc. Comput. Linguistics (ACL)*, Melbourne, Australia, vol. 1, pp. 328–339, 2018, doi: 10.18653/v1/P18-1031.

[21] H. Xu, S. Ebner, M. Yarmohammadi, A. S. White, B. Van Durme, and K. Murray, "Gradual fine-tuning for low-resource domain adaptation," in Proc. *2nd Workshop Domain Adaptation for NLP (Adapt-NLP) at EACL*, Kyiv, Ukraine, 2021, pp. 214–221.

[22] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. *36th Int. Conf. Mach. Learn. (ICML)*, Proc. Mach. Learn. Res., vol. 97, 2019, pp. 6105–6114.

[23] Khadijah, R. Kusumaningrum, Rismiyati, and N. Sabilly, "EfficientNet Model for Multiclass Classification of The Correctness of Wearing Face Mask," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 18–29, 2025, doi: 10.52549/ijeei.v13i1.5197.

[24] A. Alshoraihy, "EfficientNetB3 in Leukemia Detection: Advancements in Medical Imaging Analysis," *Medinformatics*, vol. 2, no. 3, pp. 219–225, 2025, doi: 10.47852/bonviewMEDIN52023293.

[25] J. Sharma, "Advanced Fruit Classification Using EfficientNetB3 Focusing on Apples and Tomatoes," in Proc. *3rd Int. Conf. Advancement Technol. (ICONAT)*, Goa, India, pp. 1–4, 2024, doi: 10.1109/ICONAT61936.2024.10774951.

[26] S. S. Reka, A. Bagelikar, P. Venugopal, V. Ravi, and H. Devarajan, "Deep Learning-Based Classification of Rotten Fruits and Identification of Shelf Life," *Computers, Materials and Continua*, vol. 78, no. 1, pp. 781–794, 2024, doi: 10.32604/cmc.2023.043369.

[27] S. S. S. Palakodati, V. R. R. Chirra, Y. Dasari, and S. Bulla, "Fresh and rotten fruits classification using CNN and transfer learning," *Revue d'Intelligence Artificielle*, vol. 34, no. 5, pp. 617–622, Oct. 2020, doi: 10.18280/ria.340512.

[28] M. H. I. Bijoy, S. Z. Tasnim, S. A. Awsaf, and M. Z. Hasan, "FruitVision: A benchmark dataset for fresh, rotten, and formalin-mixed fruit detection," *Data in Brief*, vol. 61, p. 111752, 2025, doi: 10.1016/j.dib.2025.111752.

[29] S. Mohammadi, S. Sattarpanah Karganroudi, M. Adda, and H. Ibrahim, "Toward smart railway maintenance: AI-enhanced Non-Destructive Evaluation using Vision Transformers and CNNs for fastener defect detection," *Green Energy and Intelligent Transportation*, vol. 5, p. 100332, 2026, doi: 10.1016/j.geits.2025.100332.

[30] A. A. A. El-Aziz, M. A. Mahmood, and S. A. El-Ghany, "Enhancing Early Detection of Oral Squamous Cell Carcinoma: A Deep Learning Approach with LRT-Enhanced EfficientNet-B3 for Accurate and Efficient Histopathological Diagnosis," *Diagnostics*, vol. 15, no. 13, Art. no. 1678, 2025, doi: 10.3390/diagnostics15131678.

[31] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 60, 2019, doi: 10.1186/s40537-019-0197-0.

[32] S. Park, J. Kim, S. Wang, and J. Kim, "Effectiveness of image augmentation techniques on non-protective personal equipment detection using YOLOv8," *Applied Sciences*, vol. 15, no. 5, Art. no. 2631, 2025, doi: 10.3390/app15052631.

[33] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, Art. no. 125, 2020, doi: 10.3390/info11020125.

[34] Henderi, T. Wahyuningsih, and E. Rahwanto, "Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (kNN) algorithm to test the accuracy of types of breast cancer," *International Journal of Informatics and Information System*, vol. 4, no. 1, pp. 13–20, 2021, doi: 10.47738/ijiis.v4i1.73.

[35] A. Mamadmurodov, S. Umirzakova, M. Rakhimov, A. Kutlimuratov, Z. Temirov, R. Nasimov, A. Meliboev, A. Abdusalomov, and Y. I. Cho, "A Hybrid Deep Learning Model for Early Forest Fire Detection," *Forests*, vol. 16, no. 5, Art. no. 863, 2025, doi: 10.3390/f16050863.

[36] M. K. Anam, S. Defit, Haviluddin, L. Efrizoni, and M. B. Firdaus, "Early Stopping on CNN-LSTM Development to Improve Classification Performance," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, 2024, doi: 10.47738/jads.v5i3.312.

[37] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[38] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," arXiv:2008.05756, 2020, doi: 10.48550/arXiv.2008.05756.

[39] J. Opitz, "A closer look at classification evaluation metrics and a critical reflection of common evaluation practice," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 820–836, 2024, doi: 10.1162/tacl_a_00675.

[40] J. Sharma and B. V. Kumar, "Automated classification of fresh and rotten fruits using ResNet50 for enhanced food quality control and waste reduction," in Proc. *Int. Conf. Pervasive Computational Technologies (ICPCT)*, Greater Noida, India, pp. 159–163, 2025, doi: 10.1109/ICPCT64145.2025.10939134.