Building of Informatics, Technology and Science (BITS)
Volume 8, No 1, June 2026 Page: 308−317
ISSN 2684-8910 (media cetak)
ISSN 2685-3310 (media online)
DOI 10.47065/bits.v8i1.9901

Classification of Swiftlet Nest Quality Based on SNI 8998:2021 Using Deep Learning

Ilmiati*, Siti Mutmainah, Khairunnas

Fakultas Teknik dan Ilmu Komputer, Program Studi Ilmu Komputer, Universitas Muhammadiyah Bima, Bima, Indonesia
Email: [1,*]ilmiatyarifin@gmail.com, [2]siti.mutmainah.id19@gmail.com, [3]assignment.khairunnas@gmail.com
Alamat Email Penulis Korespondensi: ilmiatyarifin@gmail.com
Submitted: **09/05/2026**; Accepted: **22/06/2026**; Published**: 23/06/2026**

Abstract−The quality of swiftlet nests is a key factor in determining the market value and quality standards of this commodity in both domestic and international markets. The quality classification process, which is currently dominated by manual methods, has fundamental weaknesses, namely high subjectivity and inconsistency in sorting results. This study aims to evaluate the performance of deep learning architectures in automatically classifying the quality of swiftlet nests based on visual characteristics. The main contribution of this study is to address the research gap in previous publications by strictly aligning quality class labels with the formal document of the Indonesian National Standard (SNI) 8998:2021, as well as presenting a cross-architecture comparative analysis to map model performance trade-offs. Evaluations were conducted on the MobileNetV2, and presents a cross-architecture comparative analysis to map model performance trade-offs. Evaluations were conducted on the MobileNetV2, ResNet50, and YOLOv8n-cls architectures using accuracy, precision, recall, and F1-score metrics. The research dataset includes visual images of swiftlet nests grouped into three quality classes (good, moderate, and poor) through self-documentation and augmentation techniques. Test results show that YOLOv8n-cls achieved the highest performance in this scenario with an accuracy of 99.5%, precision of 98.78%, recall of 98.72%, and an F1-score of 98.71%. Meanwhile, MobileNetV2 achieved a competitive accuracy of 98.37% with good computational efficiency, while ResNet50 demonstrated the lowest performance (66% accuracy) due to network complexity on the limited dataset. This research indicates that lightweight architectures exhibit good stability for limited-size visual datasets; however, external validation using larger datasets remains necessary to test the models' generalization capabilities more broadly.

Keywords: Deep Learning; Image Classification; MobileNetV2; ResNet50; YOLOv8n-cls; Swiftlet's Nest

1. INTRODUCTION

Indonesia is one of the world's largest producers of swiftlet nests, with export demand increasing every year [1]. Swiftlet nests have high economic value because they are believed to offer health benefits and are widely used as a premium consumer product in international markets [2]. The quality of swiftlet nests is a key factor determining the selling price of the product in both local and global markets [3]. Quality assessment is generally based on the physical shape, color, texture, and cleanliness of the nests [4]. These visual parameters also play a crucial role in the process of determining the quality of swiftlet nests, which is based on SNI 8998:2021 as the quality standard for commercial and export purposes [5] .

The process of classifying the quality of swiftlet nests is currently still carried out manually through visual inspection by human workers. This method has drawbacks, including subjective assessment, inconsistent classification results, and reliance on the assessor's experience [6]. Classification errors can result in economic losses for farmers and distributors because the product quality does not meet market standards. Therefore, an automated system based on artificial intelligence is needed that can classify the quality of swiftlet nests objectively and consistently. In addition to improving the consistency of evaluations, automated systems also have the potential to assist in standardizing the quality of swiftlet nests based on the visual characteristics used in industrial sorting processes [7].

Advances in computer vision and deep learning technologies have made significant contributions to the field of image classification. Convolutional Neural Networks (CNNs) have become one of the most widely used methods because they are capable of automatically extracting visual features [8]. Previous studies have shown that CNNs can be successfully applied to classify the quality of agricultural and biological products with a high degree of accuracy [9].

MobileNetV2 is a lightweight CNN architecture designed for computational efficiency and implementation on devices with limited resources [10]. This model uses inverted residuals and depthwise separable convolutions to reduce the number of parameters without significantly compromising classification performance. MobileNetV2 is widely used in real-time applications due to its small model size and fast inference processt [11].

ResNet50 is a deep learning architecture that uses residual learning to address the vanishing gradient problem in deep networks [12]. This architecture has more sophisticated feature extraction capabilities than lightweight CNNs. ResNet50 has been used in various image classification studies because it improves training stability and model accuracy on complex visual datasets.

YOLOv8n-cls is a lightweight variant of the YOLOv8 family designed specifically for image classification tasks, offering real-time inference capabilities and high computational efficiency. This architecture has a relatively small number of parameters compared to other YOLO variants, making it suitable for use on devices with limited computational resources [13]. YOLOv8n-cls supports a transfer learning approach through the use of pretrained weights from the ImageNet dataset, which allows the model to acquire initial visual feature representations more effectively and accelerates the training convergence process. Additionally, this model has been widely used in various image classification studies because it delivers strong performance on visual datasets of limited size and offers high inference speed for real-time system implementations [13].

Several studies on the quality testing of swiftlet nests have been conducted previously. Ahmad and Supatman [6] applied a Convolutional Neural Network (CNN) to classify the cleanliness of swiftlet nests. However, that study was limited to surface cleanliness parameters and did not evaluate the nest's geometric shape. Meanwhile, conventional approaches using the Fuzzy Tsukamoto method and Simple Additive Weighting (SAW) [14]. have proven capable of analyzing physical parameters, but still require manual feature extraction that is prone to evaluator subjectivity. In general, the majority of previous studies have a significant research gap, namely a tendency to focus on testing a single model in isolation without a comparative multi-architecture evaluation, and they have not aligned their visual parameters with formal industry standards.

To address these limitations, this study offers the advantage of an automated classification approach for swiftlet nest quality that strictly adheres to Indonesian National Standard (SNI) 8998:2021. The proposed methodological novelty is the implementation of a strategic comparison across a broad spectrum of deep learning architectures, including MobileNetV2 (lightweight architecture), ResNet50 (deep and complex architecture), and YOLOv8n-cls (state-of-the-art real-time classification architecture). Through this comparative approach, the research not only pursues the highest accuracy values but also maps the optimal trade-offs between model depth, training stability with limited augmented datasets, and inference efficiency. The scientific contribution of this research is aimed at providing a valid and practical blueprint for model recommendations for industry practitioners. By integrating visual assessment parameters into nationally standardized computer-based models, this research is expected to reduce reliance on subjective manual sorting, while also serving as a technical foundation for building consistent automated sorting technology.

2. RESEARCH METHODOLOGY

2.1 Research Steps

This research was conducted through a series of systematic steps, ranging from dataset acquisition to a comparative evaluation of deep learning model performance, with the entire experimental workflow integrated using the Google Colab environment to ensure process efficiency. In the initial stage, a primary dataset consisting of swiftlet nest images was collected and classified into three quality categories-good, moderate, and poor-through direct documentation with varying lighting conditions and shooting angles to enhance the diversity of the research data. Next, the preprocessing stage included image resizing, normalization, and the application of augmentation techniques such as rotation, flipping, zooming, and brightness adjustment to mitigate class imbalance and strengthen the model's robustness against variations in real-world objects [15]. To maintain experimental consistency, all architectures in this study were implemented in the same three-class image classification scenario: good, moderate, and poor. The YOLOv8n used in this study is the classification variant (YOLOv8n-cls), so the evaluation process remains within the image classification domain and can be directly compared with MobileNetV2 and ResNet50. The overall research stages are illustrated in Figure 1.

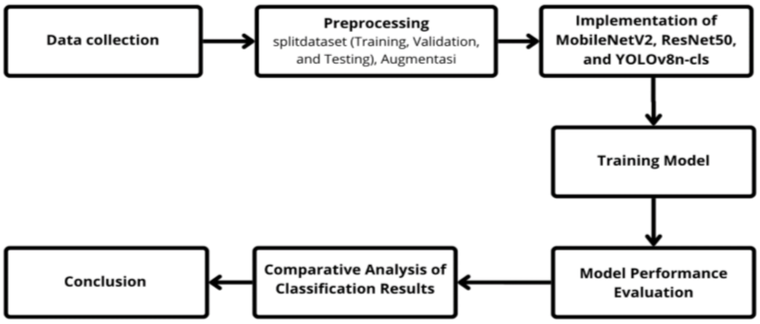

Figure 1. Research Flowchart

Based on the flowchart in Figure 1, the dataset that has undergone preprocessing is then objectively split into training, validation, and test sets so that the model's generalization ability can be accurately measured. The models were implemented on three main architectures: MobileNetV2 and ResNet50 for CNN-based image classification tasks, and YOLOv8n-cls, which is used as an image classification model based on the Ultralytics framework. Training was performed using NVIDIA Tesla T4 GPU acceleration, where MobileNetV2 and ResNet50 were trained using the TensorFlow/Keras framework with the Adam optimizer, a batch size of 16, a learning rate of 0.001, and 30 epochs. Meanwhile, YOLOv8n-cls was implemented via the Ultralytics framework by utilizing the pretrained weights from yolov8n.pt. Model performance was comprehensively evaluated using accuracy, precision, recall, F1-score, confusion matrix, metrics to determine the effectiveness of each architecture in identifying nest quality. Finally, a comparative analysis was conducted to determine the best model based on accuracy, computational efficiency, classification stability, and real-time detection capabilities to support the standardization of swiftlet nest quality at the national level.

2.2 Data Collection

The dataset used in this study consists of primary data collected through direct documentation of swiftlet nests using a digital camera under controlled lighting conditions. The dataset was labeled based on the visual characteristics of the swiftlet nests, in accordance with the quality standards of SNI 8998:2021 [5]. In this study, the visual parameters used include nest shape, nest color, and the level of physical cleanliness from contaminants such as feathers and droppings. These three parameters were selected because they are the primary characteristics commonly used in the quality sorting process of swiftlet nests in the processing and trading industries.

The three quality categories Good, Fair, and Poor, were determined manually based on the visual characteristics of the images. The "Good" category represents nests with an intact shape, an ivory-white color, and a high level of cleanliness with minimal contamination. The "Fair" category indicates nests with slight color changes or light feather contamination, while the "Poor" category includes nests with dense feather coverage, duller colors, or significant physical defects in the nest's shape. The determination of these categories refers to the visual quality parameters described in the SNI standard regarding clean swiftlet nests [5]. Details regarding the distribution of the number of images for each quality category are presented in Table 1.

Table 1. Initial Dataset

Class	Total
Good	435
Average	191
Poor	136
Total	762

2.3 Preprocessing

The preprocessing stage was performed to improve data quality prior to model training. All images were resized to 224 × 224 pixels for MobileNetV2 and ResNet50 [16]. Meanwhile, YOLOv8n used an input size of 224 × 224 pixels in accordance with the YOLO architecture standard [17]. Data augmentation techniques were used to increase data variety and reduce the risk of overfitting. Some of the augmentation techniques applied include horizontal flipping, rotation, zooming, and brightness adjustment. After augmentation, the dataset size increased to 1,227 images [18]. The detailed data distribution for each category is presented in Table 2.

Table 2. Dataset After Augmentation

Class	Amount
Good	435
Average	392
Poor	400
Total	1227

In this study, the dataset was divided into three parts: 70% for training, 20% for validation, and 10% for testing. The data was divided to ensure that the deep learning model could effectively learn the data patterns and be objectively evaluated against previously unseen data [19]. In this study, data augmentation was performed to increase image variation and reduce data imbalance between classes [20]. Augmentation aims to help the deep learning model learn the visual characteristics of swiftlet nests more effectively and improve the model's generalization ability on new data [21].

Several augmentation techniques applied include rotation, horizontal flip, zoom, and brightness adjustment. Rotation is used to generate variations in the angle of the image capture so that the model can recognize objects from various orientations. Horizontal flip is used to add variation to the position of objects in the image. The zoom technique helps the model recognize objects at various scales, while brightness adjustment is used to improve the model's ability to handle variations in lighting during the image capture process [20].

2.5 Deep Learning Models

2.5.1 MobileNetV2

MobileNetV2 is a deep learning architecture specifically designed to optimize performance on devices with limited computational resources without significantly compromising accuracy [21]. With the rapid advancement of information technology, the development of efficient computational methods has become crucial for facilitating the automatic and accurate resolution of various complex problems [21]. This architecture was chosen for this study because it has an efficient number of parameters and low computational load, enabling the classification process to be performed more quickly.

Technically, MobileNetV2 introduces an innovation in the form of an inverted residual structure with a linear bottleneck that serves to maintain data representation capacity at a low dimension [22]. The use of this inverted residual block allows the network to expand the data to a higher dimension before performing feature filtering, enabling the model to capture more complex visual patterns in swiftlet nest images, such as fine textures and color gradations.

2.5.2 ResNet50

ResNet50 (a Residual Network with 50 layers) is a Convolutional Neural Network (CNN) architecture widely recognized for its ability to perform deep feature extraction through the use of residual blocks [21]. his architecture is specifically designed to address the vanishing gradient phenomenon that frequently occurs in very deep neural networks, where gradient values tend to decrease exponentially during the backpropagation process, thereby hindering weight updates in the early layers. Through the mechanism of identity shortcut connections or skip connections, ResNet50 allows information flow from one layer to be passed directly to the next layer without undergoing non-linear transformations, which effectively maintains gradient stability and facilitates model convergence within complex network structures. In practical implementation, ResNet50 has proven to deliver superior performance in various image classification tasks due to its ability to capture high-level feature representations hierarchically [23]. The depth of this architecture provides the model with the flexibility to distinguish highly specific visual details, which is particularly relevant for identifying characteristics of biological products with high variability in texture and color. The use of ResNet50 in this study aims to provide a stronger and more accurate foundation for feature extraction, complementing the efficiency offered by previous models, thereby enabling the creation of a classification system that is more robust against various variations in input data.

2.5.3 YOLOv8n-cls

YOLOv8n-cls (You Only Look Once version 8 nano classification) is a lightweight variant of the YOLOv8 family designed specifically for image classification tasks, offering real-time inference capabilities and high computational efficiency. This model was selected because it has a relatively small number of parameters compared to other YOLO variants, enabling it to deliver good classification performance with lower computational resource requirements. This approach makes YOLOv8n-cls suitable for use in automated classification systems as well as for implementation on devices with limited specifications [24]. In addition, YOLOv8n-cls supports the use of transfer learning via pretrained weights from the ImageNet dataset, enabling the model to learn visual feature representations more effectively and accelerate the training convergence process [13]. The use of the YOLOv8n-cls architecture has also demonstrated good performance in various image classification studies based on visual characteristics, particularly for biological objects and agricultural products. With its optimized backbone structure, the model is able to efficiently extract key features from swiftlet nest images to support quality classification based on visual characteristics [25].

2.6 Hyperparameters

The swiftlet nest quality classification experiments in this study utilized three deep learning architectures: MobileNetV2, ResNet50, and YOLOv8n-cls. All models were trained using an NVIDIA T4 GPU with an image input size of 224 × 224 pixels, a batch size of 16, and 30 epochs. For the MobileNetV2 and ResNet50 architectures, the study utilized pretrained ImageNet weights with the parameter `include_top=False` at the input size (224, 224, 3) to support the transfer learning process. Meanwhile, the YOLOv8n-cls model was initialized using the pretrained weights `yolov8n-cls.pt` from the Ultralytics framework. The training process for MobileNetV2 and ResNet50 uses the Adam optimizer with a learning rate of 0.001, the categorical cross-entropy loss function, and the application of dropout at 0.5 and 0.3, respectively, on the final layers to help reduce the risk of overfitting. As for YOLOv8n-cls, it uses the default training configuration from Ultralytics, including learning rate settings and a loss function that have been optimized for image classification tasks. The data augmentation process was specifically applied to the CNN model and included normalization (rescale=1/255), random rotation of up to 10°, 10% zoom, and horizontal flipping, without specifying a brightness range. Through the application of these augmentation techniques, the number of images per class was adjusted from the pre-augmentation to the post-augmentation state, specifically: Good (435→435), Moderate (191→392), and Poor (136→400). Although there is a slight discrepancy in the Average class-where the initial target was 400 images but the final result was 392 images this condition is still sufficient for data balancing.

The decision to set the number of epochs to 30 was based on initial observations of model convergence during the training process. Within that range, the training loss and validation loss graphs showed a stable trend with no significant improvement in performance after a certain number of epochs. In addition, the training process was monitored using validation accuracy and validation loss to avoid excessive overfitting during the experiment. Details of the training configurations are presented in Table 3.

Table 3. Hyperparameters used

Hyperparameter	Value		
	MobileNetV2	ResNet50	YOLOv8n-cls
Input Shape	224, 224, 3	224, 224, 3	-
Pretrained Weights	ImageNet	ImageNet	yolov8n-cls.pt
Freeze Base Model	True	True	-

Hyperparameter	Value		
	MobileNetV2	ResNet50	YOLOv8n-cls
Optimizer	Adam (learning_rate=0.001)	Adam (learning_rate=0.001)	(unspecified, default Ultralytics)
Loss Function	categorical_crossentropy	categorical_crossentropy	(unspecified, implisit klasifikasi)
Metrics	Accuracy	Accuracy	Precision, Recall, F1-Score
Batch Size	16	16	16
Epochs	30	30	30
Image Size	224	224	224
Dropout	0.5 (sebelum Dense 128), 0.3 (sebelum output)	0.5 (sebelum Dense 128), 0.3 (sebelum output)	-
Device	GPU NVIDIA Tesla T4 (implisit)	GPU NVIDIA Tesla T4 (implisit)	GPU (device=0, Tesla T4)

Based on Table 3, the experimental results show that the performance of deep learning models is significantly influenced by hyperparameter configurations during the training process. Using a fixed learning rate of 0.001 and the Adam optimizer across all architectures does not necessarily provide the optimal configuration for every model. More complex architectures, such as ResNet50, generally require more in-depth fine-tuning strategies, such as an adaptive learning rate scheduler, gradual layer unfreezing, or additional regularization to ensure a more stable learning process. Therefore, future research should systematically explore hyperparameter optimization to achieve more representative performance.

2.7 Evaluation

A confusion matrix is used to compare actual labels with the model's predicted labels and yields four main components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [26]. Accuracy is used to measure the proportion of correct predictions overall; precision indicates the accuracy of positive predictions; recall indicates the model's ability to identify all positive instances; and the F1-score is used to assess the balance between precision and recall. The metric is calculated as follows:

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

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

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

3. RESULT AND DISCUSSION

3.1 Comparative Analysis

A comparative analysis was conducted to evaluate the performance of MobileNetV2, ResNet50, and YOLOv8n-cls in classifying the quality of swiftlet nests based on visual characteristics of the objects, such as shape, color, texture, and cleanliness. The evaluation of the three models was performed using several evaluation metrics, namely accuracy, precision, recall, and F1-score. The use of these metrics is important because each model has different architectural characteristics, so their performance needs to be assessed comprehensively to identify the strengths and limitations of each model in handling swiftlet nest image data. The results of this test are then used to determine the most optimal deep learning architecture to support the process of classifying swiftlet nest quality automatically, objectively, and efficiently. As presented in Table 4.

Table 4. Comparative Analysis

Model	Accuracy	Precision	Recall	F1-Score
MobileNetV2	98,37%	98,37%	98,37%	98,37%
ResNet50	66%	66%	65%	65%
YOLOv8n-cls	99,5%	98,78%	98,72%	98,71%

Based on the experimental scenarios in this dataset, YOLOv8n-cls demonstrated higher evaluation performance compared to other models. Although the evaluation results indicate high performance, this study has limitations regarding dataset size and has not yet utilized external validation or cross-validation. Therefore, the study's results cannot yet be broadly generalized to all variations of swiftlet nest conditions in real-world environments. Further research is needed with a larger dataset and more diverse testing scenarios to ensure the model's performance stability more comprehensively.

The fact that ResNet50 performs worse than MobileNetV2 and YOLOv8n-cls indicates that architectures with greater network depth do not always yield optimal results on limited-size datasets. Theoretically, ResNet50 possesses excellent feature extraction capabilities due to its use of residual learning and a deeper network. However, the performance of deep learning models is significantly influenced by dataset quality, hyperparameter configuration, and the fine-tuning strategies employed during training. In this study, the use of a fixed learning rate of 0.001, the Adam optimizer without further adjustments, and the process of freezing the base layers likely prevented the model from achieving optimal convergence. Additionally, the relatively limited dataset size also affected ResNet50's ability to learn discriminative features between classes in greater depth, resulting in poor model generalization to the test data. These findings indicate that high-complexity models require hyperparameter optimization and a larger dataset to achieve optimal performance.

3.2 Analysis of Training Graphs

Training plots are used to analyze the model's performance over the course of the training process. The analysis is based on changes in accuracy and loss values for both the training and validation data at each epoch. These plots help identify the model's ability to learn data patterns, the stability of the training process, and the potential for overfitting or underfitting.

3.2.1 MobileNetV2 Performance

In the MobileNetV2 model, the training accuracy and validation accuracy plots in Figure 2 show a steady increase as the number of epochs increases. The training accuracy reached 94.76%, while the validation accuracy reached 95.53%. These results indicate that the model was able to effectively learn the visual characteristics of swiftlet nests without experiencing a significant drop in performance on the validation data. Additionally, the training loss and validation loss graphs show a consistent decrease throughout the training process, indicating that the model optimization process is proceeding well and the learning parameters have converged stably. The relatively balanced pattern between the training and validation data also suggests that MobileNetV2 possesses sufficient generalization ability on this research dataset. Changes in accuracy and loss values for both training and validation data can be observed in Figure 2.

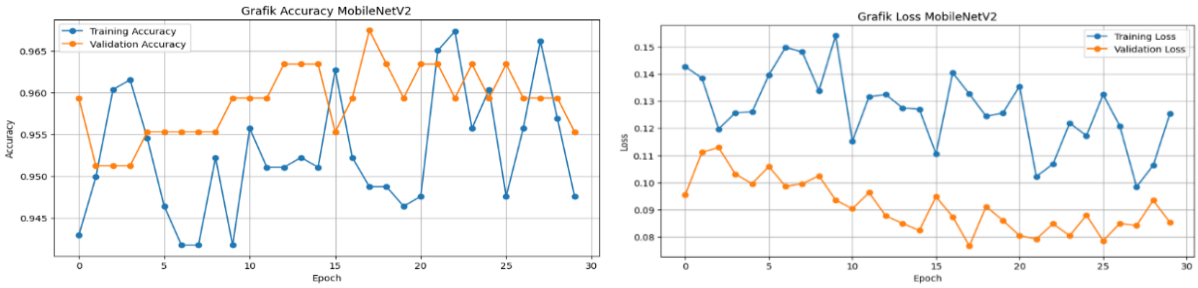

Figure 2. Training Graph

3.2.2 ResNet50 Performance

The ResNet50 model exhibits less stable learning patterns compared to MobileNetV2. Although training accuracy improved during the training process, validation accuracy tended to fluctuate and did not show consistent improvement. This suggests that the model still struggles to generalize to the test data. Theoretically, ResNet50 has strong feature extraction capabilities because it uses a residual learning architecture with a deeper network. However, the complexity of this architecture also results in a greater need for training data and a more optimized fine-tuning process so that the model can effectively learn discriminative features. Additionally, using the same hyperparameter configuration as other models such as a fixed learning rate and a limited number of epochs may not yet be sufficient to achieve optimal convergence in ResNet50. The validation loss graph in Figure 3, which tends to stagnate in the final few epochs, also indicates overfitting, where the model begins to over-adapt to the training data, resulting in no significant improvement in performance on the validation data. The patterns of changes in accuracy and loss values during the training process can be seen in Figure 3.

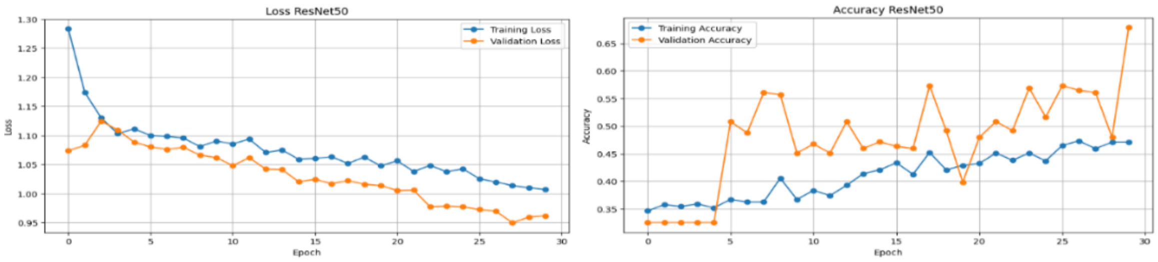

Figure 3. Training Graph

3.2.3 Yolov8n-cls Performance

The YOLOv8n-cls model demonstrated high evaluation performance on this dataset. Based on the test results, the model achieved an accuracy of 99.5%, a precision of 98.78%, a recall of 98.72%, and an F1-score of 98.71%. These values indicate that the model is capable of effectively recognizing the visual characteristics of swiftlet nests during the classification process. Additionally, the training graph in Figure 4 shows a relatively stable learning pattern throughout the training process. The gradual decrease in the loss value indicates that the model optimization process is proceeding well and the model is able to achieve stable convergence. YOLOv8n-cls's ability to achieve high evaluation performance is also influenced by its lightweight architecture, which is efficient in extracting visual features from images. However, the results of this study are still limited to the dataset and experimental scenarios used; therefore, further testing with larger datasets and additional validation is required to ensure the model's broader generalization capabilities. Changes in the accuracy, precision, recall, and F1-score values during the training process can be seen in Figure 4.

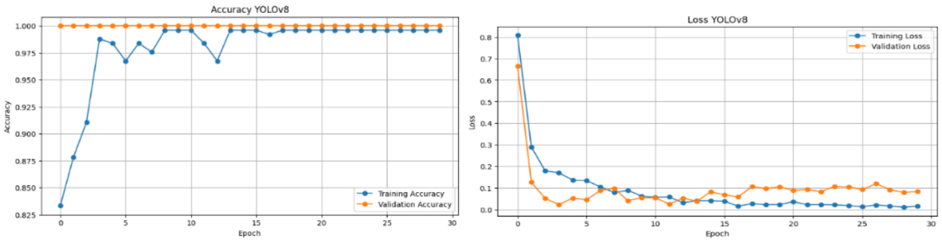

Figure 4. Training Graph

Based on the analysis of the training graphs in Figures 2, 3, and 4, YOLOv8n-cls demonstrates higher classification performance compared to other models. The model's ability to efficiently extract visual features contributes to improvements in accuracy, precision, recall, and F1-score. However, the results of this study are still limited to the dataset and experimental scenarios used; therefore, additional testing with larger datasets and external validation are needed to ensure the model's performance consistency under real-world conditions.

3.3 Confusion Matrix Analysis

The confusion matrix is an important evaluation tool in machine learning for thoroughly assessing the performance of classification models. This matrix provides a visual representation that compares the model's predicted values with the actual values in the test data, thereby revealing how accurately the model predicts each specific class. In this study, the confusion matrix was analyzed to assess the ability of the MobileNetV2, ResNet50, and YOLOv8n-cls models to distinguish between good, moderate, and poor classes based on the visual characteristics of swiftlet nests. Through this analysis, cross-class classification errors can be identified to evaluate the strengths and weaknesses of each architecture in recognizing the unique features of each quality level.

3.3.1 Analysis of the MobileNetV2 Model

Based on the results in Figure 5, the MobileNetV2 model demonstrated good classification performance with a relatively low prediction error rate. Most of the test data was successfully classified into the class corresponding to the actual label. The model achieved a test accuracy of 98.37%, indicating that MobileNetV2 is capable of consistently recognizing the visual characteristics of each quality category of swiftlet nests. Classification errors occurred only in a few samples with similar visual characteristics, particularly between the medium and poor classes. This indicates that the model still struggles to perfectly distinguish the visual features between certain classes. Despite its lightweight architecture, MobileNetV2 still demonstrates strong generalization capabilities on this dataset with relatively efficient computational processing. The evaluation results of the MobileNetV2 model on the test data can be seen in the confusion matrix presented in Figure 5.

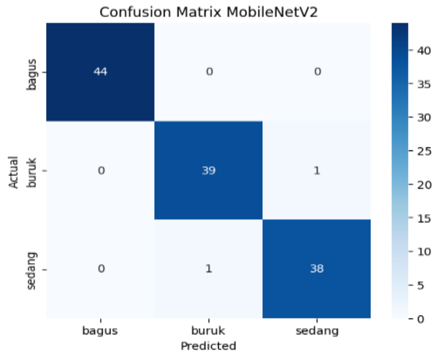

Figure 5. Confusion Matrix

3.3.2 Analysis of the ResNet50 Model

Based on the results in Figure 6, the ResNet50 model still exhibits some fairly significant classification errors, particularly in the “moderate” and “poor” classes. The model often predicts the “moderate” class as “poor” and vice versa because these two categories share similar visual characteristics, particularly in terms of texture, color, and the cleanliness of the nests. The classification report results show that the “good” class achieves higher precision and recall scores compared to other classes, while the “moderate” class has the lowest precision and F1-score. This indicates that the model struggles to distinguish more subtle visual features within categories that share nearly identical characteristics. Theoretically, ResNet50 possesses strong feature extraction capabilities due to its use of a residual learning architecture with a deeper network. However, the complexity of this architecture also means the model requires a larger dataset and a more optimized fine-tuning process to effectively learn discriminative features. Additionally, using the same hyperparameter configuration as other models likely did not result in an optimal learning process for ResNet50, thereby affecting the model’s low generalization ability on the test data. The evaluation results of the ResNet50 model on the test data are presented in the form of a confusion matrix and a classification report in Figure 6.

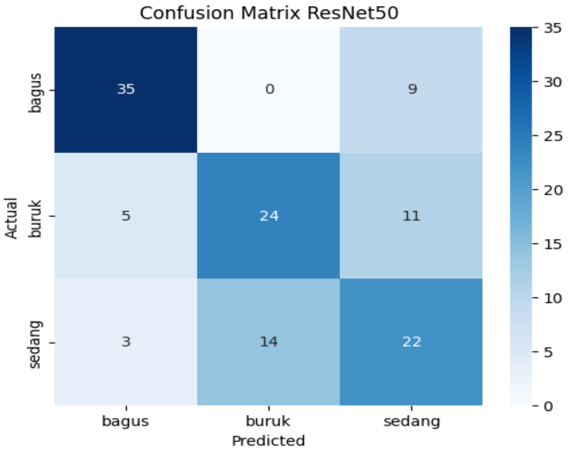

Figure 6. Confusion Matrix

3.3.3 Analysis of the YOLOv8n-cls Model

Based on the results in Figure 7, the YOLOv8n-cls model demonstrated excellent classification performance with a relatively low prediction error rate across all categories of swiftlet nest quality. Most of the test data was successfully classified according to the true labels in each class. The model achieved a precision of 98.78%, a recall of 98.72%, and an F1-score of 98.71%, indicating that the model was able to consistently recognize the visual characteristics of the images during the classification process. Prediction errors occurred only in a small number of samples that shared similarities in shape, color, and cleanliness levels across classes. This performance is attributed to the YOLOv8n-cls architecture’s ability to efficiently extract visual features through a lightweight backbone structure optimized for image classification tasks. Additionally, the use of pretrained weights from ImageNet also helps the model learn visual feature representations more effectively on this research dataset. However, these evaluation results are still limited to the experimental scenarios and datasets used; therefore, additional testing using larger datasets and external validation is required to ensure the model’s broader generalization capabilities. The evaluation results of the YOLOv8n-cls model on the test data are presented via a confusion matrix and classification report in Figure 7.

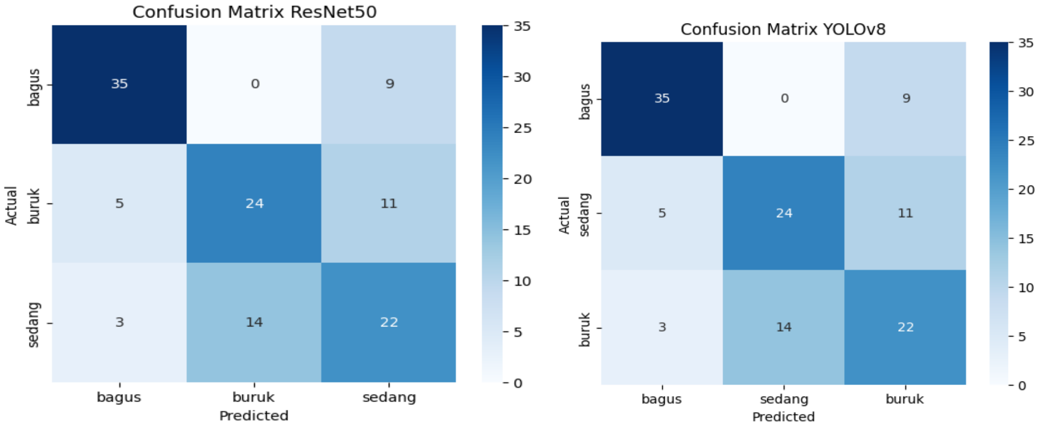

Figure 7. Confusion Matrix

3.4 Discussion of Findings and Literature Review

The experimental results show that the choice of model architecture affects classification performance, particularly on datasets of limited size. Based on the testing scenarios conducted, YOLOv8n-cls and MobileNetV2 demonstrated more stable evaluation performance compared to ResNet50. MobileNetV2 achieves high accuracy with good computational efficiency, while YOLOv8n-cls achieves the highest evaluation performance on this dataset because it effectively extracts visual features during the image classification process. Meanwhile, ResNet50's lower performance indicates that architectures with deeper network complexity require larger datasets and more thorough hyperparameter optimization for optimal generalization.

The performance characteristics demonstrated by YOLOv8n-cls (99.5% accuracy) and MobileNetV2 (98.37% accuracy) provide insight into the potential of deep learning to support objective quality assessment processes. When compared to a conventional decision support system-based approach using the Simple Additive Weighting (SAW) method [4], which recorded an accuracy of 85%, the use of automatic classification models in this study offers more measurable consistency. This is because the previous rule-based method was highly dependent on the accuracy of manual human input regarding the physical criteria of the nests (such as size and moisture content), so the accuracy level remained fluctuating due to operator subjectivity.

Nevertheless, the high accuracy achieved by YOLOv8n-cls and MobileNetV2 in this test should be viewed in context and cannot be generalized absolutely to all conditions. In a previous study conducted by Ahmad and Supatman [6] using a conventional CNN architecture, the accuracy achieved was 84%, with a focus limited to surface cleanliness parameters only. The improvement in metric values achieved by the YOLOv8n-cls model in this study indicates that its backbone structure works effectively when combined with multi-condition augmentation techniques. However, this performance is still dependent on computational environment parameters and the specific characteristics of the augmented dataset used in this experiment.

Confirmation of this model's limitations is reinforced by the low accuracy of ResNet50, which reaches only 66%, empirically demonstrating that deeper architectures do not always guarantee superior results on relatively sparse local datasets. Therefore, the advantages of YOLOv8n-cls and MobileNetV2 in this study are more accurately interpreted as the models' efficiency in recognizing subtle visual features (shape, color, and texture) specific to the scope of the tested dataset. The alignment of quality categories with the formal SNI 8998:2021 standard ultimately serves as an initial bridge to provide a reference for a more standardized automated approach for the swiftlet nest sorting industry before testing on a larger scale of real-world data.

4. CONCLUSION

Based on the results of the experiments and the discussion, it can be concluded that the deep learning-based approach to classifying the quality of swiftlet nests exhibits varying performance characteristics depending on the complexity of the model architecture, with lightweight architectures tending to be more effective and stable for exploring subtle visual features in limited-size local datasets. In this testing scenario, YOLOv8n-cls achieved the highest evaluation metrics with an accuracy of 99.5%, precision of 98.78%, recall of 98.72%, and an F1-score of 98.71%, followed by MobileNetV2 with a competitive accuracy of 98.37%, while ResNet50 recorded the lowest score of 66% due to convergence issues in the sparse dataset. The practical contribution of this research lies in the adoption of SNI 8998:2021 regulatory parameters as a reference for image labeling, thereby providing a development direction aligned with the need for objective industrial sorting standardization. Although optimal results were achieved on the test data, the generalization ability of these models is limited because they remain constrained by the scope of the experimental dataset's characteristics and the computational environment parameters used. Therefore, future research requires external validation testing using a larger dataset, as well as exploration of variations in lighting and object background parameters to comprehensively test the model's stability before considering practical implementation.

REFERENCES

[1] W. Adelina and R. S. Munawaroh, "Analisis Kualitas Produk dan Strategi Pemasaran Sarang Burung Walet di Kecamatan Pelaihari Kabupaten Tanah Laut," *Din. Ekon. J. Ekon. dan Bisnis*, vol. 16, no. 2, pp. 393–402, 2023, doi: 10.53651/jdeb.v16i2.453.

[2] R. A. Sari, R. A. Setiawan, and E. Setiawan, "Analisis Potensi Pasar Ekspor Sarang Burung Walet Menuju Pasar Internasional dalam Perspektif Ekonomi Islam (Studi Kasus di Desa Rena Panjang)," *J. Ilm. Mhs. Perbank. Syariah*, vol. 6, no. 2, pp. 691–706, 2025, doi: 10.36908/jimpa.

[3] M. Irfan, A. Patimbangi, and Jumriani, "Analisis Budidaya Sarang Burung Walet terhadap Peningkatan Pendapatan Rumah Tangga di Kelurahan Lonrae Kabupaten Bone," *Arus J. Sos. dan Hum.*, vol. 5, no. 2, 2025, doi: 10.57250/ajsh.v5i2.1174.

[4] P. R. Prayogo and P. H. Susilo, "Sistem Pendukung Keputusan dalam Menentukan Kualitas Sarang Burung Walet Terbaik Menggunakan Metode Simple Additive Weighting (SAW)," *Insearch (Information Syst. Res. J.*, vol. 2, no. 2, pp. 83–89, 2022, doi: 10.15548/isrj.v2i02.4363.

[5] BSN, "Penetapan Standar Nasional Indonesia 8998:2021, Sarang Burung Walet Bersih," SCRBD. Accessed: Sep. 05, 2026. [Online]. Available: https://www.scribd.com/document/623102005/SNI-Sarang-Burung-Walet

[6] S. Ahmad and Supatman, "Analisis CNN untuk Klasifikasi Kebersihan Sarang Burung Walet," *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3, 2025, doi: 10.23960/jitet.v13i3.6685.

[7] Fitriany, M. Zainul, and A. Santi, “Kebijaksanaan Inovasi Guna Meningkatkan Produktivitas dan Kualitas Sarang Burung Walet (Studi Usaha Burung Walet PT. Adipurna Mranata Jaya Kapuas),” *Pros. Semin. Nas. Ekon.*, vol. 1, pp. 133–138, 2023, doi: 10.31602/piuk.v0i0.12465.

[8] I. A. Prasetya, F. Sukandiarsyah, N. A. Fitri, and S. Adam, “Klasifikasi Kualitas Buah Jeruk menggunakan Computer Vision dengan Arsitektur YOLOv8,” *J. Pendidik. Inform. dan Sains*, vol. 13, no. 2, pp. 187–201, 2024, doi: 10.31571/saintek.v13i2.8346.

[9] M. A. Syaharani, T. Aurelly, C. Budianto, and R. I. Adam, “Klasifikasi Buah Segar dan Busuk Menggunakan Algoritma Convolutional Neural Network (CNN),” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 5, pp. 10823–10827, 2024, doi: 10.36040/jati.v8i5.11132.

[10] S. A. Maulana, S. H. Batubara, Y. Permata, P. Pasaribu, H. Syahputra, and F. Ramadhani, “Deteksi Burung menggunakan Convolutional Neural Network (CNN) dengan Model Arsitektur MobileNetV2,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 4, pp. 6108–6114, 2024, doi: 10.36040/jati.v8i4.10126.

[11] M. Adeva, F. Muttaqin, and B. M. Mulyo, “Analisis Efisiensi Arsitektur U-Net dengan Encoder MobileNetV2 pada Segmentasi Karat Daun Kopi,” *J. Comput. Sci. Inf. Technol.*, vol. 7, no. 1, pp. 84–90, 2026, doi: 10.37859/coscitech.v7i1.11221.

[12] M. I. Anshori, R. A. Yaqin, M. Ali, Z. Sidiq, and W. P. Agung, “Klasifikasi Jenis Jerawat secara Otomatis dengan Convolutional Neural Network Menggunakan Arsitektur Resnet-50,” *J. Manaj. Inform.*, vol. 15, no. April, pp. 73–84, 2025, doi: 10.34010/jamika.v15i1.13712.

[13] E. Tenda, S. C. W. Ngangi, C. A. J. Soewoeh, and E. Ketaren, “Sistem Deteksi Hama pada Tanaman Jagung (Zea Mays) Berbasis Kecerdasan Buatan dan Internet Of Things (IoT),” *J. TIMES*, vol. XIV, no. 2, pp. 225–233, 2025, doi: 10.51351/jtm.14.2.2025879.

[14] F. Agus, E. Sulfika, and G. Mahendra Putra, “Analisis Kualitas Sarang Burung Walet menggunakan Metode Fuzzy Tsukamoto,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 12, no. 2, pp. 391–398, 2025, doi: 10.25126/jtiik.2025129441.

[15] D. Husen, “Evaluasi Teknik Augmentasi Data untuk Klasifikasi Tumor Otak Menggunakan CNN pada Citra MRI,” *EKNIMEDIA Teknol. Inf. dan Multimed.*, vol. 5, no. 2, pp. 219–227, 2024, doi: 10.46764/teknimedia.v5i2.220.

[16] I. P. Agus, K. Hidjah, N. Sulistianingsih, G. Hendro, and Syahrir, “Implementasi Arsitektur Deep Convolutional Neural Network (CNN) dengan Transfer Learning untuk Klasifikasi Penyakit Kulit,” *J. Teknol. Inf. dan Multimed. (JTIM)*, vol. 7, no. 3, pp. 461–477, 2025, doi: 10.35746/jtim.v7i3.734.

[17] G. V. Agustin, M. Ayub, and S. L. Liliawati, “Deteksi dan Klasifikasi Tingkat Keparahan Jerawat : Perbandingan Metode You Only Look Once,” *J. Tek. Inform. dan Sist. Inf.*, vol. 10, no. 3, pp. 468–481, 2024, doi: 10.28932/jutisi.v10i3.9414.

[18] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop , Rotate , dan Mixup,” *J. Tek. Inform. dan Sist. Inf.*, vol. 6, no. 2, pp. 311–323, 2020, doi: 10.28932/jutisi.v6i2.2688.

[19] N. H. Muttaqin and A. M. Widodo, “Evaluation of Transfer Learning-Based Convolutional Neural Networks (InceptionV3 and MobileNetV2) for Facial Skin-Type Classification,” *J. Ilmu Komput. dan Inform.*, vol. 5, no. 1, pp. 11–32, 2025, doi: 10.54082/jiki.264.

[20] M. M. Nugraha, M. D. F. Saputra, D. A. Fauzan, and E. Y. Puspaningrum, “Augmentasi Data Pengenalan Citra Batik Yogyakarta menggunakan Pendekatan Random Crop , Rotate , dan MixUp,” *Semin. Nas. Inform. Bela Negara*, vol. 5, no. 1, pp. 152–158, 2025, doi: 10.33005/santika.v5i1.680.

[21] M. A. Al-fahrezi, “Pengaruh Augmentasi Data terhadap Akurasi Pelatihan Model CNN untuk Klasifikasi Jenis Ikan,” *J. Ilm. Teknol. Sist. Inf.*, vol. 6, no. 2, pp. 177–185, 2025, doi: 10.62527/jitsi.6.2.471.

[22] H. Firdaus, “Pengembangan Algoritma Numerik yang Efisien untuk Menyelesaikan Persamaan Diferensial Parsial dalam Simulasi Matematika Modern,” *Mandalika J. Ilmu Pendidik. dan Bhs.*, vol. 3, no. 1, 2025, doi: 10.59613/jipb.v3i1.289.

[23] H. Rumapea, “Evaluasi Kinerja CNN dan Vision Transformer pada Klasifikasi Citra Resolusi Tinggi Berbasis Deep Learning,” *METHOMIKA J. Manaj. Inform. Komputerisasi Akunt.*, vol. 9, no. 2, pp. 372–379, 2025, doi: 10.46880/jmika.Vol9No2.pp372-379.

[24] G. Jocher, A. Chaurasia, and J. Qiu, “Explore Ultralytics YOLOv8,” Ultralytics Docs. Accessed: May 22, 2026. [Online]. Available: https://docs.ultralytics.com/models/yolov8

[25] A. Rismayanti and R. Rahmadewi, “Deteksi dan Klasifikasi Tingkat Kematangan Buah Mangga Harum Manis Menggunakan You Only Look Once (YOLO) V8,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 9, no. 3, pp. 3645–3654, 2025, doi: 10.36040/jati.v9i3.13320.

[26] T. Puspita, E. R. Swedia, M. Cahyanti, and M. R. D. Septian, “A Real-Time Helmet Detection System Based on YOLOv8 to Support Traffic Law Enforcement Sistem Deteksi Penggunaan Helm Secara Real-Time Berbasis YOLOv8 untuk Mendukung Penegakan Hukum Lalu Lintas,” *Sebatik*, vol. 29, no. 1, pp. 1–10, 2025, doi: 10.46984/sebatik.v29i1.2585.