

Classification of Diseases in Oyster Mushroom Baglogs Using the MobileNetV3-Small Architecture

Muhammad Vihir Mujahid, Ery Hartati*

Faculty of Computer Science and Engineering, Universitas Multi Data, Palembang, South Sumatra, Indonesia

*Correspondence: ery_hartati@mdp.ac.id

SUBMITTED: 30 September 2026; REVISED: 12 October 2026; ACCEPTED: 16 October 2026

ABSTRACT: Oyster mushroom baglogs are highly susceptible to contamination by pathogens such as *Trichoderma*, *Mucor*, *Neurospora*, and *Penicillium*, while disease identification still relies primarily on subjective manual observation by farmers. This study evaluated the effect of data augmentation on the classification performance of the MobileNetV3-Small architecture for detecting diseases in oyster mushroom baglogs. A primary dataset comprising 900 independently collected images was organized into five classes: Healthy, *Trichoderma*, *Mucor*, *Neurospora*, and *Penicillium*. The dataset was divided at the physical baglog level into 600 training images (66.67%), 150 validation images (16.67%), and 150 testing images (16.67%) to prevent data leakage between subsets. Two experimental scenarios were compared under identical conditions: a baseline model without augmentation and an augmented model in which data augmentation was applied exclusively to the training set. Model training employed transfer learning using ImageNet-pretrained weights, beginning with feature extraction and followed by progressive fine-tuning. Classification performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrices. The results demonstrated that the baseline model achieved a test accuracy of 91.33% and an F1-score of 0.9127, whereas the augmented model achieved a higher test accuracy of 94.67% and an F1-score of 0.9466. All classification errors occurred between the visually similar *Penicillium* and *Trichoderma* classes, with data augmentation reducing the number of misclassified images from 13 to 8. Furthermore, both models correctly classified all test baglogs using majority voting across multiple image views. These findings demonstrate that data augmentation improves the classification performance of MobileNetV3-Small, particularly in distinguishing visually similar disease classes. The study provides a foundation for developing lightweight, augmentation-enhanced disease classification systems for oyster mushroom cultivation.

KEYWORDS: Image classification, MobileNetV3-small, oyster mushroom, transfer learning, data augmentation

1. Introduction

Oyster mushroom (*Pleurotus ostreatus*) is a promising agricultural commodity due to its relatively simple cultivation requirements, suitability for limited growing spaces, strong market demand, and short production cycle of approximately 3–4 months per cultivation bag (baglog). Biologically, *P. ostreatus* belongs to the phylum Basidiomycota, a group of fungi that

reproduce sexually through specialized microscopic structures called basidia, which produce basidiospores. Although its reproductive structures are microscopic, the fungus develops a macroscopic, shell-shaped fruiting body that is visible to the naked eye. As a saprophytic organism, *P. ostreatus* obtains nutrients by decomposing organic matter, primarily the sawdust-based substrate contained within the baglog. Oyster mushrooms are available in several varieties, including white, brown, and gray, and can be consumed fresh or processed into various food products [1]. Oyster mushroom cultivation also offers considerable economic potential because of increasing consumer demand and its nutritional value. However, production capacity remains relatively limited in some cultivation areas. Oyster mushrooms contain approximately 19–35% protein, 72% unsaturated fat as a proportion of total fat, and 7–24% dietary fiber, although these values may vary depending on the variety, cultivation conditions, and analytical basis [2].

Oyster mushroom cultivation has gained considerable attention in the agricultural sector because of its potential contribution to food production and agricultural diversification. The cultivation substrate typically consists of organic materials such as straw, sawdust, wood powder, rice bran, and lime, which are prepared and packed into cultivation bags known as baglogs. Although oyster mushrooms are relatively easy to cultivate, contamination remains a persistent challenge throughout the production cycle and can significantly reduce crop quality and yield [3]. Contamination may result from unfavorable environmental conditions, inadequate cultivation practices, improper substrate preparation, and contaminated mushroom spawn. Therefore, maintaining appropriate temperature and moisture conditions, together with proper sanitation and cultivation procedures, is essential for supporting healthy mycelial development and minimizing contamination risks [4].

Fungal contamination can inhibit mycelial growth and disrupt the development of oyster mushroom fruiting bodies. Several fungal genera are commonly associated with contamination in oyster mushroom baglogs, including *Trichoderma*, *Mucor*, *Neurospora*, and *Penicillium*. *Trichoderma*, commonly referred to as green mold, is characterized by the appearance and spread of green patches on the substrate. *Mucor*, often associated with black mold contamination, typically develops white, cotton-like mycelial growth that may gradually become gray or dark as spores mature. *Neurospora* can spread rapidly through airborne spores, potentially contaminating neighboring baglogs and causing substantial production losses. Meanwhile, *Penicillium* contamination commonly appears as blue-green or greenish patches and may interfere with mushroom growth and substrate colonization [1]. The economic consequences of contamination were highlighted by Mona et al. [3], who reported approximately 5,000 contaminated baglogs across mushroom cultivation sites in Semarang, Ungaran, and Salatiga. Difficulties in identifying contamination during its early stages contributed to abnormal fungal growth and crop failure, resulting in estimated financial losses ranging from IDR 10 million to IDR 15 million [3].

The challenge of contamination management is further complicated by the continued reliance on manual disease identification through direct visual observation by farmers. Although visual inspection is widely practiced, its effectiveness depends heavily on individual experience, technical knowledge, and the ability to distinguish among visually similar contamination symptoms. This approach is inherently subjective and may result in inconsistent identification, particularly during the early stages of fungal contamination when visible characteristics are not yet fully developed. Misidentification or delayed detection can prevent

timely intervention, allowing contamination to spread and potentially causing substantial crop losses [5]. Consequently, there is a growing need for technology-based diagnostic approaches capable of providing accurate, rapid, consistent, and objective identification of contamination directly within mushroom cultivation environments [6].

Digital image processing based on Convolutional Neural Networks (CNNs) has emerged as a promising approach for automated visual classification. CNNs are deep learning architectures capable of extracting hierarchical spatial features from images through convolutional operations, enabling the recognition of patterns associated with different object categories or disease symptoms [7]. Their ability to learn discriminative visual features has supported applications in agricultural disease detection and fungal classification. Darius et al. [7] applied a CNN-based approach to identify contamination in oyster mushroom baglogs and reported a validation accuracy of 97.14% across four contamination classes: *Trichoderma*, *Mucor*, *Neurospora*, and *Penicillium*. These findings demonstrate the potential of deep learning for automated contamination identification. However, further investigation is needed to determine whether computationally efficient architectures can achieve reliable classification performance using images collected from different physical baglogs while minimizing the risk of data leakage between training and evaluation subsets.

Among lightweight CNN architectures, MobileNetV3-Small offers a promising alternative for image classification in environments with limited computational resources. The architecture incorporates inverted residual bottleneck blocks, Squeeze-and-Excitation (SE) modules, and hard-swish activation functions to improve feature extraction efficiency while reducing computational complexity [16]. Compared with larger CNN architectures, MobileNetV3-Small is designed to balance classification performance with computational efficiency, making it potentially suitable for applications on mobile and embedded devices. The architecture also incorporates improvements derived from hardware-aware neural architecture search and NetAdapt optimization, which contribute to efficient network design. These characteristics are particularly relevant to agricultural applications where computational resources, power availability, and hardware capacity may be limited.

The selection of MobileNetV3-Small in this study was motivated by its relatively low computational requirements and potential suitability for lightweight image classification. Compared with MobileNetV3-Large, the Small variant generally requires fewer parameters and computational operations, making it more appropriate for resource-constrained applications [16]. Previous research by Rakhmat and Rizkiawarman [10] examined different MobileNetV3 variants for mushroom image classification, demonstrating the relevance of this architecture family to fungal image recognition. The MobileNetV3-Small implementation used in the present study comprises approximately 1.01 million parameters, with a stored model size of approximately 12 MB. Furthermore, transfer learning using ImageNet-pretrained weights enables the model to utilize previously learned visual representations, potentially improving feature extraction and classification performance when training data are limited. Nevertheless, computational suitability for deployment cannot be established solely from model size and parameter count. Actual inference speed, memory consumption, and hardware compatibility require further evaluation under practical operating conditions [11].

Despite advances in CNN-based agricultural image classification, several research gaps remain in the detection of fungal contamination in oyster mushroom baglogs. Previous studies have primarily focused on plant disease identification using leaf and fruit images or on the

general classification of fungal species. Many investigations have also relied on publicly available datasets, including Kaggle and PlantVillage, which may not adequately represent the visual variability encountered in actual mushroom cultivation environments. Furthermore, existing research has provided limited evidence regarding the application of MobileNetV3-Small to the classification of multiple contamination conditions in oyster mushroom substrates using independently collected primary image datasets. The effects of data augmentation on classification performance, particularly when visually similar contamination classes are involved, also require further systematic evaluation. In addition, image-level dataset partitioning may introduce data leakage when multiple images of the same physical cultivation unit are distributed across training, validation, and testing subsets, potentially producing overly optimistic performance estimates.

To address these gaps, this study evaluates the performance of MobileNetV3-Small in classifying five oyster mushroom baglog conditions: Healthy, *Trichoderma*, *Mucor*, *Neurospora*, and *Penicillium*. A primary dataset comprising 900 images collected directly from oyster mushroom cultivation baglogs was used to compare two experimental scenarios: a baseline model without data augmentation and a model trained with augmentation applied exclusively to the training dataset. To minimize data leakage, the dataset was partitioned at the physical baglog level, ensuring that images originating from the same baglog were not distributed across different subsets. The study also examines class-specific classification errors, particularly confusion between the visually similar *Trichoderma* and *Penicillium* classes, and evaluates prediction consistency at the individual baglog level using majority voting across multiple image views.

The main contribution of this study is the systematic evaluation of data augmentation within a lightweight MobileNetV3-Small framework using a primary dataset of oyster mushroom baglog images and a baglog-level data partitioning strategy. Unlike approaches that emphasize overall classification accuracy alone, this study examines the influence of augmentation on class-specific errors and evaluates whether predictions remain consistent across multiple views of the same cultivation unit. Therefore, this study aims to (1) develop a MobileNetV3-Small classification model for five oyster mushroom baglog conditions, (2) evaluate the effect of data augmentation on classification performance, and (3) analyze classification errors and baglog-level prediction consistency. The findings are expected to provide empirical evidence supporting the future development of lightweight image-based contamination detection systems for oyster mushroom cultivation.

2. Materials and Methods

2.1. Oyster Mushroom.

White oyster mushroom (*Pleurotus ostreatus*) is an economically valuable agricultural commodity due to its relatively simple cultivation requirements, limited land requirements, short production cycle, and consistently high market demand [2]. Taxonomically, *P. ostreatus* belongs to the phylum Basidiomycota and is classified as a saprophytic fungus that obtains nutrients by decomposing organic materials, particularly decaying wood. It produces large, macroscopic fruiting bodies characterized by broad, rounded caps resembling oyster shells. Oyster mushrooms occur in several commercially recognized types, including white, brown,

and gray varieties, with white oyster mushrooms being widely consumed as fresh produce and processed food products.

The cultivation of white oyster mushrooms offers considerable economic and nutritional potential. Increasing consumer demand, combined with relatively accessible cultivation techniques and low land requirements, has encouraged the expansion of oyster mushroom production. However, production capacity may remain insufficient to satisfy growing market demand. Nutritionally, oyster mushrooms contain approximately 19–35% protein, which is higher than that of several staple foods, including rice, when compared on a dry-weight basis. In addition, approximately 72% of their total fatty acid content consists of unsaturated fatty acids, while dietary fiber accounts for approximately 7.4–24.6%, depending on the analytical basis and cultivation conditions [1]. These nutritional characteristics contribute to the increasing popularity of oyster mushrooms in domestic and international markets.

Another advantage of oyster mushroom cultivation is its potential for year-round production under suitable environmental conditions without requiring extensive agricultural land. Consequently, mushroom farming can provide opportunities for both primary and supplementary income generation [12]. Nevertheless, cultivation activities remain vulnerable to several biological and environmental risks that can adversely affect productivity and profitability, particularly when cultivation management is inadequate. One of the major challenges is the early identification of fungal contamination in cultivation substrates, commonly referred to as baglogs. Delayed or inaccurate identification of contamination symptoms may lead to ineffective preventive measures, progressive substrate deterioration, and substantial crop losses [5]. Therefore, reliable and timely contamination detection is essential for maintaining cultivation quality and improving production efficiency.

2.2. Oyster Mushroom Disease.

Diseases and contamination in oyster mushroom cultivation can disrupt normal mycelial growth, substrate colonization, and fruiting body development. These disturbances may be associated with various biological agents, including fungi and bacteria, which compete with oyster mushroom mycelia for nutrients and growing space. Fungal contamination is particularly problematic because it can spread rapidly within cultivation substrates and significantly reduce production quality. Several fungal genera commonly associated with oyster mushroom baglog contamination include *Trichoderma*, *Neurospora*, *Mucor*, and *Penicillium* [13].

Trichoderma spp., commonly known as green mold, are among the most frequently encountered fungal contaminants in mushroom cultivation. Contamination may occur during substrate preparation, inoculation, or incubation, particularly when sterilization procedures or cultivation hygiene are inadequate. The presence of *Trichoderma* is generally indicated by green patches that develop on the substrate as spores mature. Severe contamination can inhibit oyster mushroom mycelial growth and interfere with substrate colonization, ultimately reducing mushroom production [14].

Neurospora spp., commonly associated with orange bread mold, can be identified by the development of orange-colored fungal growth on contaminated substrates. This contaminant may spread rapidly through airborne spores, increasing the likelihood of transmission between adjacent baglogs. When contamination becomes extensive, it can interfere with substrate

utilization and prevent normal oyster mushroom development, potentially resulting in substantial production losses [1].

Mucor spp., often associated with black mold contamination, may develop during the incubation or propagation stages following inoculation. Contamination is generally characterized by cotton-like fungal growth that may become gray or black as sporangia mature. In severely affected baglogs, changes in substrate appearance and texture may also occur. The proliferation of *Mucor* can compete with oyster mushroom mycelia for nutrients and space, thereby restricting normal growth and reducing cultivation performance.

Penicillium spp. represent another group of fungal contaminants that may affect oyster mushroom baglogs. Contamination is commonly indicated by light-green, blue-green, or greenish powdery patches distributed across the substrate surface. These patches may expand as fungal growth and sporulation progress. Although the severity of contamination varies depending on environmental conditions and the extent of fungal development, prolonged contamination can inhibit oyster mushroom mycelial growth and reduce substrate productivity [7].

Because these contaminants exhibit differences in color, texture, and growth patterns, visual characteristics can provide useful preliminary indicators for disease identification. However, similarities in the appearance of certain fungal contaminants, particularly *Trichoderma* and *Penicillium*, may complicate accurate identification through manual observation. These challenges highlight the importance of developing image-based classification methods capable of distinguishing different contamination conditions consistently and objectively.

2.3. Digital Image Processing.

Digital image processing refers to the use of computational techniques to manipulate, enhance, analyze, and extract meaningful information from digital images. A digital image consists of a collection of discrete picture elements, commonly known as pixels, each containing numerical values representing image intensity or color information. Digital image processing techniques enable the identification of visual characteristics such as color distribution, texture, shape, and spatial patterns. In agricultural applications, these techniques have been widely employed for plant disease detection, crop monitoring, and automated visual inspection. Compared with conventional manual observation, image-based analysis offers opportunities to improve the speed, consistency, and objectivity of disease identification [15].

Digital images may originate from still photographs or individual frames extracted from video recordings, including images captured using digital cameras or webcams. Mathematically, a grayscale image can be represented as a two-dimensional discrete function, $f(x,y)$, where (x) and (y) denote spatial coordinates and $f(x,y)$ represents the intensity value of the pixel at the corresponding location. In a digital image, these intensity values are arranged in a numerical matrix containing a finite number of rows and columns.

For an image with (N) rows and (M) columns, the digital image matrix can be expressed as follows:

$$f(x,y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & f(1,1) & \dots & f(1,M-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

An image $f(x, y)$, in mathematical function form, can be written as follows:

$$0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

$$0 \leq f(x, y) \leq G-1$$

Where M is the number of row pixels in the image array, N is the number of column pixels in the image array, and G is the gray level value

The magnitudes of M , N , and G are generally powers of two: $M = 2^m$; $N = 2^n$; $G = 2^k$, where the values m , n , and k are positive integers. The interval $(0, G)$ is called the grayscale. The magnitude of G depends on the digitization process. Typically, a gray value of 0 (zero) represents black intensity, and 1 (one) represents white intensity. For an 8-bit image, the value of G equals $2^8 = 256$ colors (gray levels).

2.4. Deep Learning.

Deep learning is a type of machine learning that uses deep neural networks to learn and identify complicated patterns in data. Deep neural networks are artificial neural networks with multiple layers. This approach has the ability to automatically learn features from the raw data. Thus, the important data features are identified without relying on manual feature engineering and the network learns the features during the training phase.

2.5. Convolutional Neural Network (CNN).

A CNN is a type of artificial neural network widely used for processing image data. The strength of a CNN lies in its ability to accurately detect and recognize objects, as its architecture is specifically designed to process data with a grid-like structure. Most image data processing uses convolutional neural networks (CNNs). CNNs can accurately find and identify objects because they are designed to handle grid-like data. CNNs were invented in the 1960s and have shown great promise in the area of computer vision. The CNN, from a technical point of view, is a trainable architecture consisting of a number of processing stages. It is a combination of two main processes, the image convolution for feature extraction and an artificial neural network for classification. A CNN basically has four main layers which are operated in a sequence, which are Convolution Layer, Activation Layer, Pooling Layer and Fully Connected Layer. These four layers work in tandem to extract key features from an image while simultaneously classifying it. An illustration of the CNN architecture is shown in Figure 1.

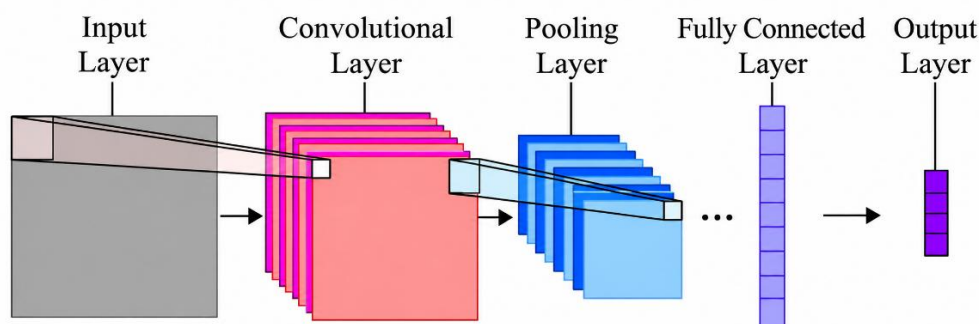


Figure 1. CNN Layer Structure.

2.6. MobileNetV3.

MobileNetV3 is the third generation of the MobileNet family of Convolutional Neural Network (CNN) architectures developed by Google, specifically designed to achieve high computational efficiency on mobile devices and resource-constrained systems. Its development incorporates two principal optimization approaches: Platform-Aware Neural Architecture Search (NAS), which identifies an efficient overall network structure by balancing classification accuracy and inference latency, and the NetAdapt algorithm, which further optimizes individual network layers to improve computational performance.

MobileNetV3 also introduces several architectural improvements, including the hard-swish (h-swish) nonlinear activation function, Squeeze-and-Excitation (SE) modules, and optimization of the initial and final network layers. These enhancements improve feature representation while reducing computational complexity. As illustrated in Figure 2, the MobileNetV3 block integrates pointwise and depthwise convolutional operations, nonlinear activation functions, and channel-wise attention mechanisms to support efficient feature extraction. Residual connections are also employed in compatible blocks to facilitate information flow and improve network efficiency. Together, these architectural features enable MobileNetV3 to achieve a favorable balance between classification accuracy, model complexity, and computational requirements. Consequently, MobileNetV3 is particularly suitable for image classification applications on devices with limited processing capabilities, including mobile and embedded systems.

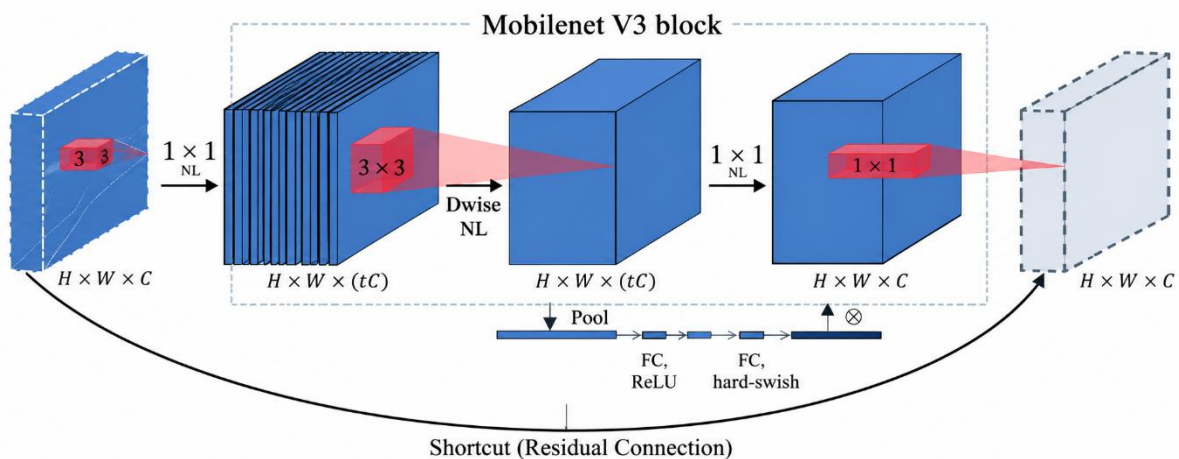


Figure 2. MobileNetV3 Architecture.

2.6.1. MobileNetV3-Small

In 2019, Google Brain and Google AI presented MobileNetV3-Small as an enhancement of the MobileNetV1 and MobileNetV2 architectures. This variant has a simpler and lightweight network structure, which is suitable for devices with limited computational ability such as Internet of Things (IoT) devices and embedded systems. Its architectural development began with a search for a network structure from scratch including adjustment of weighting parameters to establish a baseline model. The model was improved further with the NetAdapt method and other optimizations applied to particular parts of the network. The MobileNetV3-Small variant with this method requires less parameters and floating-point operations than the MobileNetV3-Large variant [16], thus reducing the computational requirements.

MobileNetV3-Small was designed to ease data processing on resource-constrained devices by using a smaller architecture. This model is less computation intensive than the Large model. The maximum expansion size is 576 and the number of output channels before the classification stage is 1024. Most of the layers employ Squeeze-and-Excitation (SE) modules, which help the network to learn and represent features better. h-swish activation function is used in the beginning part of the network. MobileNetV3-Small, which achieves 67.5% accuracy with ~2.5M parameters. Benchmark testing results demonstrate that this architecture yields 6.6% accuracy gain over MobileNetV2 at the same latency levels. Thus, this architecture has the potential to maintain classification performance at relatively low computational requirements. The structural parameters and complete specifications of the MobileNetV3-Small architecture are detailed in Table 1.

Table 1. MobileNetV3-Small Architecture Specifications.

Input Size	Operator	Exp Size	Out	SE	NL	Stride
224×224×3	Conv2d 3×3	–	16	–	HS	2
112×112×16	Bottleneck 3×3	16	16	✓	RE	2
56×56×16	Bottleneck 3×3	72	24	–	RE	2
28×28×24	Bottleneck 3×3	88	24	–	RE	1
28×28×24	Bottleneck 5×5	96	40	✓	HS	2
14×14×40	Bottleneck 5×5	240	40	✓	HS	1
14×14×40	Bottleneck 5×5	240	40	✓	HS	1
14×14×40	Bottleneck 5×5	120	48	✓	HS	1
14×14×48	Bottleneck 5×5	144	48	✓	HS	1
14×14×48	Bottleneck 5×5	288	96	✓	HS	2
7×7×96	Bottleneck 5×5	576	96	✓	HS	1
7×7×96	Conv2d 1×1	–	576	–	HS	1
7×7×576	AvgPool 7×7	–	–	–	–	–
1×1×576	Conv2d 1×1	–	1024	–	HS	1

2.7. Transfer Learning.

Transfer learning is a deep learning technique that utilizes a model previously trained on a large dataset like ImageNet, to be reused in another problem with similar data characteristics. The transfer learning method makes the model training more efficient by leveraging the knowledge acquired in previous training sessions. This method often requires less time and data to train a model compared to building and training a network from scratch. The working principle of transfer learning is to keep the convolutional and pooling layers that have already learnt many visual features and adapt the last layers of the network to the number of classes in the target dataset. The last layers give values that are predictions for each class in the form of a one dimensional array that is used to determine the classification result [17].

2.8. Fine-Tuning.

Fine-tuning is used in the later stage of transfer learning to adjust the model capability to the dataset. This is achieved by unfreezing some layers of the pre-trained model, and allowing the parameters in these layers to be fine-tuned by further training in using the research data. By adjusting the weights of the network, the model can learn more specific patterns and features

of the images being used. Hence, it is hypothesized that the fine-tuning method would increase the capacity of a model to differentiate between classes leading to more accurate classifications.

2.9. Data Augmentation.

Data augmentation is a technique to increase the diversity of the training data by applying different transformations to the images without changing their class labels. Typical transformations are rotation, flipping, scaling or zooming. By this technique the model learns the object's features from different position and viewpoints, which helps to avoid the dependence on specific visual patterns that exist in the initial training data. Moreover, Data augmentation methods are employed in this study to enhance the model's ability to generalize to new images and to reduce the risk of overfitting during the training process.

2.10. Dataset collection.

The dataset used in this study was a primary dataset collected independently through image acquisition of oyster mushroom (*Pleurotus ostreatus*) baglog media obtained from several oyster mushroom farmer in Palembang, South Sumatra. Images were captured individually using an normal iPhone 13 under natural daylight conditions with an estimated illumination of 300 lux and a plain white background to minimize visual noise. The dataset consisted of 900 images grouped into five classes: four contamination classes (*Trichoderma*, *Mucor*, *Neurospora*, and *Penicillium*) and one healthy class, with 180 images per class to ensure a balanced distribution Figure (3).



Figure 3. Example of oyster mushroom baglog condition: (A) *Trichoderma*, (B) *Mucor*, (C) *Neurospora*, (D) *Penicillium*, (E) Healthy.

To maintain the quality and consistency of the captured images, each baglog was placed on a wooden table 70 cm high, while the height of each baglog was approximately 25 cm. Image acquisition took place inside a box lined with plain white cardboard, with the front and top left open to allow controlled natural lighting. Additionally, an 80 cm tall tripod was used to keep the camera stable and aligned with eye level. Images were captured from four mutually perpendicular sides of each baglog, supplemented by two additional angles from the upper-left and upper-right at a 20° forward tilt to capture contamination hidden near the baglog opening. Image compositions included variations with and without the baglog ring, as well as close-up shots, letting the model learn both the overall structure of the baglog and fine-grained spore textures. The natural variations in lighting as a consequence of the transit of the sun during the day were deliberately maintained as a natural addition to increase the robustness of the model against real-world changes in illumination.

All images were saved in JPEG format with 1152x1536 pixel resolution, and converted to PNG format through Google Colab for background removal.

2.11. Research Design.

The objective of this research is to classify five types of oyster mushroom baglog conditions using MobileNetV3-Small architecture with the transfer learning method. The dataset contains 900 images with equal distribution among the classes namely *Trichoderma*, *Mucor*, *Neurospora*, *Penicillium* and Healthy with 180 images each. Such even data distribution guarantees an equal number of samples for each category during the model development and evaluation stages. The research was carried out in several methodological stages as follows.

2.11.1. Preprocessing.

The preprocessing will be launch in two steps. In the first step, all the images were reduced in size from their original resolution 1152×1536 pixels to 224×224 pixels in order to satisfy the input requirements of the MobileNetV3-Small architecture. In the second step, the pixel values were normalized to the $[0, 1]$ range in order to speed up convergence and ensure numerical stability during training.

2.11.2 Data splitting.

The dataset in this study consists of 900 images evenly distributed across five classes, with 180 images per class. This dataset is divided into three subsets comprising 600 training images (66.67%), 150 validation images (16.67%), and 150 testing images (16.67%). The data partitioning was performed based on the physical baglog sources to prevent images originating from the same baglog from appearing in different subsets. This group-based splitting procedure was intended to reduce data leakage caused by visually correlated images captured from multiple viewpoints of the same physical baglog. The resulting dataset distribution was used consistently for both experimental scenarios, with and without data augmentation, to ensure a fair comparison of classification performance.

2.11.3 Data augmentation.

To assess the effect of data augmentation on the classification performance of MobileNetV3-Small, two experimental situations were carried out. For this study, the model training process was conducted under two different circumstances. In the first case, the original training images were used without applying any data augmentation techniques, whereas in the second case geometric transformations were employed in order to increase the variety of images in the training set. The augmentation techniques employed in this study include horizontal and vertical flipping, 45° and 90° rotations, and zooming. These techniques were applied to generate diverse object orientations, viewpoints, and sizes, enabling the model to learn contamination characteristics from various visual perspectives. The augmentation process was applied solely to the training dataset, while images in the validation and testing datasets remained unaltered to ensure consistency in evaluation conditions. Moreover, this study did not make use of the MixUp augmentation technique or any other transformations such as changes in brightness and contrast during the training process.

2.11.4 *Overfitting mitigation strategy.*

Because the main dataset used in the study was relatively small, consisting of 900 images, several measures were put in place in order to reduce the risk of overfitting when the model was being trained. Along with using data augmentation, a Dropout rate of 0.5 was introduced in the classification layer to decrease the model's dependence on particular neurons. Moreover, an Early Stopping technique was used which included a patience value of 7 epochs so that training would stop if there was no improvement in validation performance over a period of seven consecutive epochs. The study also made use of ReduceLROnPlateau in order to deal with the learning rate, causing it to be lowered automatically whenever the validation loss failed to improve. This combination of techniques is intended to stabilise the process by which the network updates its weights, This will help to decrease the model's tendency to memorise the training data and improve its ability to generalise to images which it has not seen before.

2.11.5 *Model development.*

The MobileNetV3-Small architecture (which had been pre-trained on the ImageNet dataset) is used as the main element for extracting image features. It was chosen because of its high parameter efficiency (with about 2.5 million parameters), its appropriateness for training when only limited datasets are available, and since it makes use of inverted bottleneck and Squeeze-and-Excitation (SE) mechanisms, allowing the network to learn visual features efficiently. In this study, the last part of the architecture is altered by incorporating a Global Average Pooling layer in order to reduce the dimensionality of the features produced during extraction. A Dense layer with 128 neurons and a ReLU activation function is then employed, after which a Dropout layer (with a rate of 0.5) is used to reduce the likelihood of overfitting, a Softmax activation function with five neurons is applied at the output stage, and each neuron corresponding to one of the oyster mushroom baglog condition categories that are being classified.

2.11.6 *Training.*

The model's training makes use of a transfer learning approach that is carried out in two stages and is intended to take 50 epochs with a batch size of 32. During training, the categorical cross-entropy loss function is used to measure the prediction error, and the Adam algorithm is applied as the method for optimizing the updates to the network weights. In the first stage, which involves feature extraction and covers epochs 1 to 35, all of the base layers of the MobileNetV3-Small model remain frozen so that their parameters do not change. Training is carried out solely on the new classification layers at a learning rate of 0.001. After that stage, the process moves on to fine-tuning from epoch 36 to 50; in this phase, some of the deeper layers of MobileNetV3-Small are unfrozen so that their weights can be adjusted to suit the particular features of the oyster mushroom baglog images. The learning rate is then lowered to 0.0001 in order to ensure that the updates to the weights are more controlled. Through these two stages, the model was expected to leverage visual features learned from ImageNet while simultaneously adapting high-level feature representations to the contamination characteristics present in the research dataset.

Table 2. MobileNetV3-Small training parameters.

Parameter	Configuration
Architecture	MobileNetV3-Small
Input Image Size	224x224
Batch Size	32
Epoch	50
Initial Stage (Freeze)	1-35
Fine-Tuning Stage	35-50
Unfrozen Layers	Selected Final Layers of MobileNetV3-Small
Dense Layer	128 Neuron (ReLU)
Dropout	0.5
Optimizer	Adam
Learning Rate	0,001
Learning Rate Fine-Tuning	0,0001
Loss Function	Categorical Crossentropy

3. Results and Discussion

The performance of MobileNetV3-Small was evaluated on two scenarios, without augmentation (Scenario 1) and with augmentation (Scenario 2), on a test dataset of 150 images from five physical baglogs that were not seen during training or validation. Model will be performance was assessed using the confusion matrix, including accuracy, precision, recall, and F1-score. Validation loss was reported for the selected checkpoint. Previous research has demonstrated the application of a CNN-based approach to classify diseases in oyster mushroom baglogs [7], while the lightweight MobileNetV3 architecture has also been investigated for mushroom image classification but not for oyster mushroom baglogs [10].

3.1. Model Performance Without Augmentation.

The baseline model achieved a test accuracy of 91.33%, with a precision of 0.9166, a recall of 0.9133, and an F1-score of 0.9127. In stage-wise validation, feature extraction reached 91.33% validation accuracy, fine-tuning raised it to 94.67%, and full fine-tuning kept 94.67% with the lowest validation loss (0.0917), so full fine-tuning was selected. Early stopping ended feature extraction at epoch 31 (best epoch 24). The confusion matrix is presented in Figure 4, and 13 of the 150 test images were misclassified. Using transfer learning with MobileNetV3-Small is consistent with previous research investigating lightweight architectures for image classification [9, 16]. However, the performance values report in this study were obtained from the present baglog dataset and are not directly comparable with those of studies using different evaluation datasets. The confusion matrix in Figure 4 shows that the classification errors in the baseline model were concentrated in the *Trichoderma* and *Penicillium* classes. Of the 30 *Trichoderma* test images, nine were incorrectly classified as *Penicillium*, resulting in a recall of 0.70. In the meantime, four cases of *Penicillium* were wrongly classified because the model recognised them as *Trichoderma*. This indicates that the baseline model still has trouble in distinguishing between these two types of contamination, even though it has been able to correctly classify all the images in the *Healthy*, *Mucor*, and *Neurospora* groups. One possible explanation for this error is the visual similarity between *Trichoderma* and *Penicillium*, particularly since both can exhibit a greenish discoloration on the surface of the baglog substrate [1], [7]. In addition to colour, other factors such as the density of the contamination, the texture, the fungal growth patterns, and the angles from which the images were taken might also influence the visual features that the model has learned. It therefore seems that relying solely on colour information is probably not sufficient to consistently tell the two categories

apart. The relatively low recall rate for Trichoderma also suggests that incorporating a greater variety of features into the training data could help improve the model's ability to identify this class. However, the precise reasons for these misclassifications cannot be definitely established since the study did not perform any specific feature visualization analysis or model interpretability tests. Hence, this explanation remains a hypothesis based on the misclassification patterns that were observed during testing.

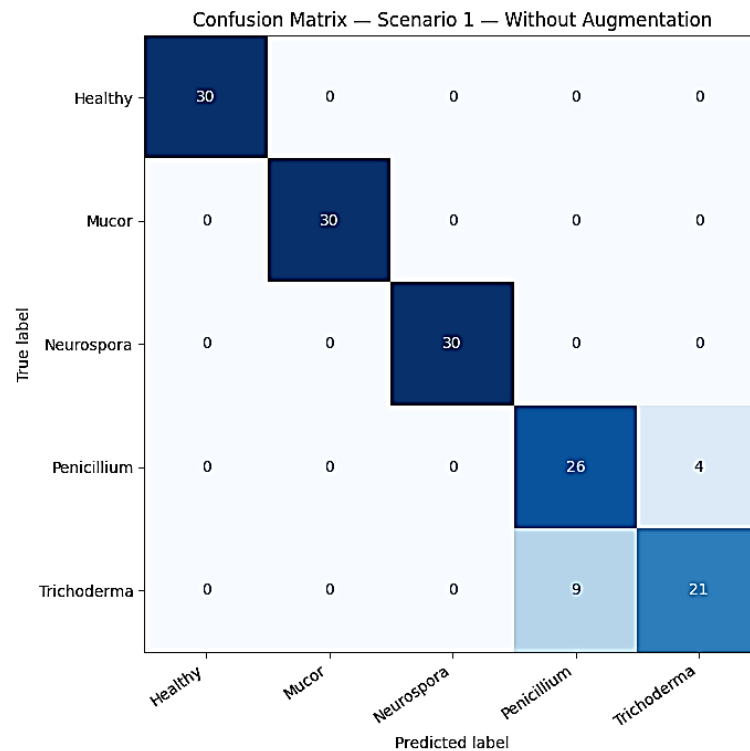


Figure 4. Confusion Matrix of the Model Without Augmentation.

3.2. Model Performance With Augmentation.

In the augmentation scenario, the model achieved a test accuracy of 94.67%, with a precision of 0.9473, a recall of 0.9467, and an F1-score of 0.9466. The selected checkpoint (full fine-tuning) obtained a validation accuracy of 94.67% with a validation loss of 0.0707. Feature extraction alone reached only 88.67% validation accuracy and ended at epoch 34 (best epoch 27), which was slower than the baseline. The slower convergence is due to the greater variety of images in the training data after data augmentation techniques have been applied. The transformations used produce a broader range of visual features, so the model has to go through a more complicated learning process in order to detect patterns in each image. Using data augmentation may result in a slightly slower convergence, but it can greatly improve the model's ability to generalize to new datasets [18]. As seen in the confusion matrix in Figure 5, the model misclassified only 8 among a total of 150 test images examined. From the results of the confusion matrix as shown in Figure 5, The accuracy of the classification obtained with the MobileNetV3-Small model was increased as a result of applying data augmentation, especially in distinguishing Trichoderma from Penicillium. Where previously nine cases of Trichoderma had been incorrectly identified as Penicillium in the baseline case without augmentation, this number was reduced to five. Likewise, the number of instances in which Penicillium was misidentified as Trichoderma decreased from four to three, the total number of errors in the

test set thus falling from 13 to 8. The Healthy, Mucor, and Neurospora categories, on the other hand, showed the same classification results in both testing situations. This improvement can be explained by the greater spatial variation in the training images that results from transformations such as horizontal flipping, vertical flipping, rotation, and zooming. These transformations allowed the model to learn contamination patterns in different orientations and sizes, possibly decreasing its dependence on the particular positioning or arrangement of disease features present in the original images. This conclusion is in agreement with earlier research which shows that data augmentation increases the diversity of the training data and improves the model's ability to generalize when recognizing new images [18]. The improvement in test accuracy from 91.33% to 94.67%, together with the reduction in misclassification, suggests that the augmented model performed more consistently on the evaluated test images. Nevertheless, the contribution of each augmentation technique was not examined individually, and further evaluation using additional independent baglogs is required to confirm whether these improvements remain consistent across different cultivation environments.

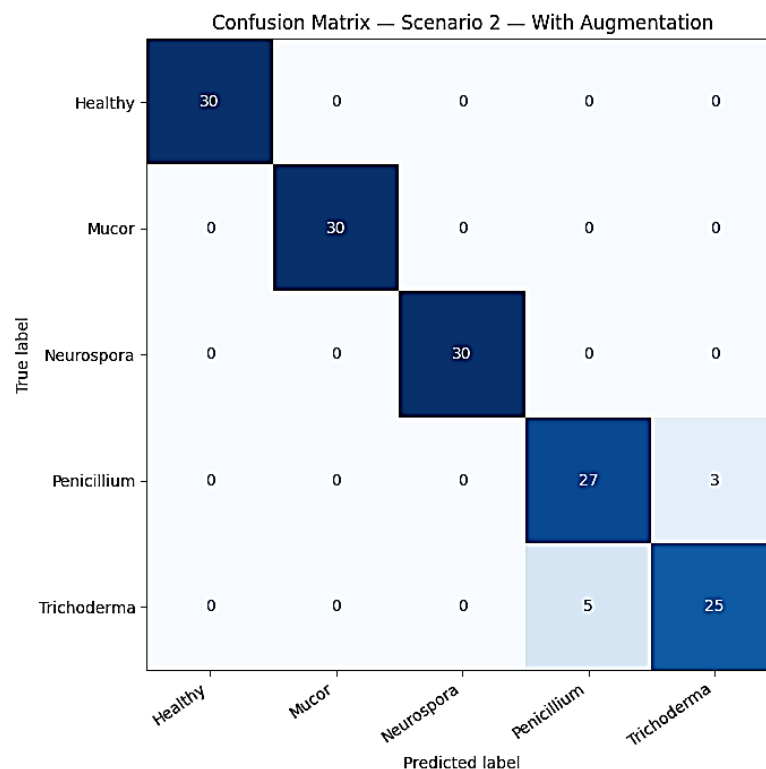


Figure 5. Confusion Matrix of the Model With Augmentation.

3.3. Comparison of Performance With And Without Augmentation.

Table 3 compares per-class precision, recall, and F1-score. The model with augmentation reached a test accuracy of 94.67% and F1-score of 0.9466, and without augmentation reach 91.33% and 0.9127. Augmentation therefore improved accuracy by 3.33 percentage points and the F1-score by 3.39 percentage points, which corresponds to five fewer misclassified images. This improvement is consistent with previous findings that image augmentation can improve the generalization ability of deep learning models by increasing the variation of training samples [18].

Table 3. Comparison of Precision, Recall, and F1-Score with and Without Augmentation.

Class	Without aug. Precision	Recall	F1-score	With aug. Precision	Recall	F1-score
Healty	1.00	1.00	1.00	1.00	1.00	1.00
Mucor	1.00	1.00	1.00	1.00	1.00	1.00
Neurospora	1.00	1.00	1.00	1.00	1.00	1.00
Penicillium	0.74	0.87	0.80	0.84	0.90	0.87
Trichoderma	0.84	0.70	0.76	0.89	0.83	0.86

From the evaluation results shown in Table 3, the use of data augmentation techniques led to an improvement in the classification performance of the MobileNetV3-Small model, especially with regard to the Penicillium and Trichoderma categories. For the Penicillium class, the precision increased from 0.74 to 0.84, the recall went up from 0.87 to 0.90 and the F1-score rose from 0.80 to 0.87. This shows that the model trained using augmentation reduced the number of prediction errors for Penicillium while at the same time increasing the proportion of Penicillium images that were correctly identified. Likewise, similar gains were seen in the Trichoderma class, with precision increasing from 0.84 to 0.89, recall rising from 0.70 to 0.83 and the F1-score going from 0.76 to 0.86. The increase in recall for the Trichoderma class is a notable result since this category had the highest number of classification errors in the case without augmentation. These findings are also supported by the fact that the number of Trichoderma images misclassified as Penicillium decreased from nine in the non-augmented situation to five after augmentation was applied. On the other hand, the Healthy, Mucor, and Neurospora categories kept their precision, recall, and F1-score at 1.00 in both testing situations. This means that the application of augmentation had no effect on the classification performance for these three categories as measured by the test dataset in the study. In general, the improvements were mainly seen in the two classes that have similar visual characteristics, which suggests that augmentation helped the model to classify the previously ambiguous contamination patterns more consistently. Nevertheless, further testing using additional independent baglogs is needed to establish whether these improvements remain consistent across different cultivation conditions. The observed improvement is consistent with previous research discussing the role of data augmentation in enhancing training-data diversity and reducing overfitting in deep learning models [18].

The patterns shown in Figures 6 and 7 are the changes in accuracy and loss during the training and validation of the MobileNetV3-Small model in both testing situations. In each case, there is a trend for accuracy to increase while loss decreases as the number of epochs grows. This suggests that the model is gradually learning the visual features that are used to differentiate the conditions of the oyster mushroom baglogs. The vertical line on the graphs marks the beginning of the fine-tuning phase, which involves reactivating the layers of the previously trained model in order to update the network weights and make them better match the characteristics of the research dataset. According to the learning curves obtained, both models show a tendency to converge by the end of the training process. Yet the accuracy and loss values on the validation data still show certain fluctuations, meaning that the model's performance has not completely stabilised during the learning process. In the augmentation scenario, the validation loss at the selected checkpoint was lower than that of the baseline model (0.0707 compared with 0.0917). These observations suggest more favorable validation

performance with augmentation, although they do not establish the complete absence of overfitting.

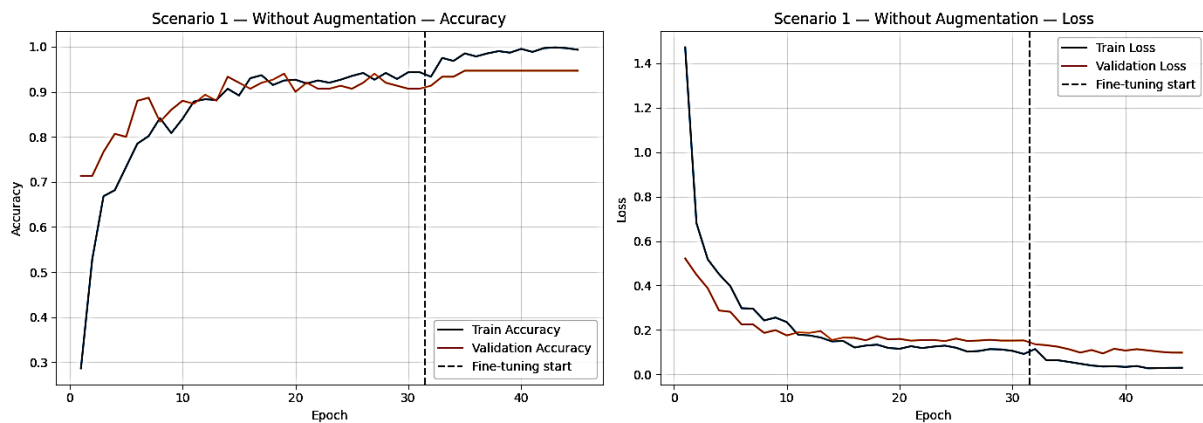


Figure 6. Accuracy and Loss Per Epoch without Augmentation.

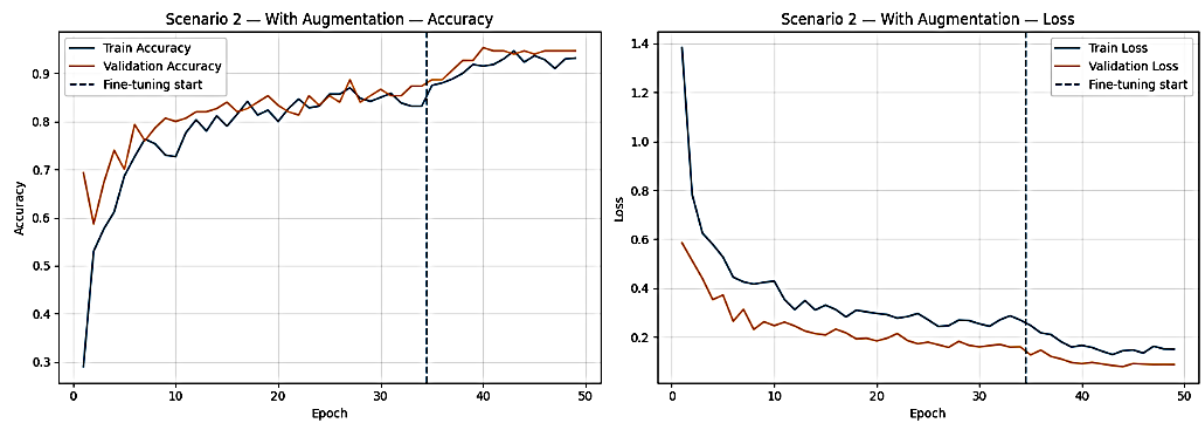


Figure 7. Accuracy and Loss Per Epoch with Augmentation.

The relatively limited validation dataset and the use of a single data partition mean that the models' generalization behavior should be interpreted cautiously. The improvement was concentrated in the two visually closest classes. *Trichoderma* recall increased from 0.70 to 0.83, and the F1-scores of *Penicillium* and *Trichoderma* increased from 0.80 to 0.87 and from 0.76 to 0.86, respectively. The classes with distinct colors (white *Healthy*, black *Mucor*, and orange *Neurospora*) were already separated perfectly, so augmentation had no room to improve them. Both diseases in the confusing pair appear as green-toned contamination, so color alone is a weak separator and the model must rely on finer texture cues. The application of geometric augmentation techniques in this study including horizontal and vertical flipping, rotation, and zooming generated additional variations in object orientation and size within the training images. These variations probably enabled the model to adapt to the different positions of the baglogs and the various angles from which the images were taken, thus improving its ability to recognise contamination patterns in images that it had not seen before during training. This is in agreement with previous research which shows that data augmentation increases data diversity and helps to improve the generalisation abilities of deep learning models [18]. Nevertheless, this study did not assess the individual effect of each augmentation technique. Moreover, a feature interpretability analysis such as Gradient-weighted Class Activation Mapping (Grad-CAM) was not carried out in order to find the visual features that had the

greatest influence on the model's predictions. A comparison of the validation and testing accuracies also showed better performance in the case with augmentation. In the case without augmentation (baseline), accuracy fell from 94.67% during validation to 91.33% during testing, a difference of 3.34 percentage points. On the other hand, the model that used augmentation achieved the same accuracy of 94.67% in both the validation and testing phases, meaning that there was no difference between the validation and testing accuracy. These results indicate that the use of augmentation can result in more consistent generalisation capabilities when the model is employed to classify images of baglogs that have not been seen before. However, the validation set was also used to select the checkpoint and consisted of only five baglogs, so this gap should be interpreted with caution. At the baglog level, the majority voting approach resulted in correct predictions for all five independent test baglogs in both experimental scenarios, corresponding to a baglog-level accuracy of 100% within the evaluated dataset. However, this result should be interpreted cautiously because the number of independent test baglogs was limited. Although multiple images were captured from each baglog, these images originated from the same physical source and therefore cannot be considered independent biological samples. Therefore, the classification accuracy at the baglog level obtained in this study does not necessarily reflect the model's overall performance when applied to different cultivation environments or larger baglog populations. Furthermore, the evaluation process utilized only a single dataset-splitting scheme; consequently, the ability to assess the stability and consistency of the model's predictions across varying data remains limited. Future studies should include a larger number of independent baglogs collected from multiple cultivation locations and apply repeated group-based validation to evaluate the reliability and generalization performance of the proposed classification model. The augmented model also reached a lower validation loss at the selected checkpoint (0.0707 versus 0.0917). Augmentation increased only training difficulty and did not change the model: both models had 1,013,621 parameters and a saved size of about 12 MB.

4. Conclusions

This study evaluated the performance of the MobileNetV3-Small architecture in classifying five conditions of oyster mushroom baglogs, namely Healthy, Trichoderma, Mucor, Neurospora, and Penicillium, through two experimental scenarios: without data augmentation and with data augmentation. The baseline model obtained a test accuracy of 91.33%, a precision of 0.9166, a recall of 0.9133, and an F1-score of 0.9127. On the other hand, the model that was trained using data augmentation showed better performance in terms of testing results, reaching an accuracy of 94.67%, a precision of 0.9473, a recall of 0.9467, and an F1-score of 0.9466. These results show that the use of data augmentation enhanced the classification performance of MobileNetV3-Small on the dataset employed in the study. The most significant improvements were seen in the Trichoderma and Penicillium categories; these fungi have similar visual features, which makes it difficult for the model to tell them apart. After applying the augmentation, the number of misclassified images fell from 13 to 8, and the recall for the Trichoderma class increased from 0.70 to 0.83. In contrast, the Healthy, Mucor, and Neurospora categories showed no change in classification performance in either training scenario. These findings suggest that geometric augmentation techniques such as flipping, rotation, and zooming can aid the model in recognising contamination patterns regardless of the orientation and size of the objects within the images. As for practical use, the model

developed has the potential to help oyster mushroom growers identify contamination in baglog (mushroom growth substrate) media by means of digital image analysis. The relatively lightweight MobileNetV3-Small architecture, with about 1.01 million parameters and a model size of roughly 12 MB, provides a good basis for creating an image-based classification system that can be integrated with mobile devices or edge computing hardware. Such a system could enable more efficient monitoring of baglog conditions and decrease growers' dependence on subjective visual inspections. However, this study mainly concentrates on the development and evaluation of the classification model and therefore does not yet include implementation as an application suitable for direct use in cultivation environments. Even though the application of data augmentation led to improved classification performance, the study does have some limitations. The dataset consisted of 900 images taken under fairly controlled conditions and involved a small number of independent physical substrate logs (baglogs). Moreover, the model was assessed using only one dataset split scheme, so the consistency of its performance across different samples and in various cultivation environments has not been fully examined. The continued occurrence of classification errors between the *Trichoderma* and *Penicillium* categories indicates that these two types of contamination, which have similar visual characteristics, need further analysis. Hence, future research should make use of larger and more diverse datasets from a number of cultivation sites and involve repeated validation according to the source of the physical substrate logs in order to get a better understanding of the model's performance stability. The use of feature visualization methods, such as Gradient-weighted Class Activation Mapping (Grad-CAM), could also be considered to determine the image regions that have the greatest influence on the classification decisions. Additionally, it is necessary to test the inference time and computational efficiency on mobile devices or edge computing platforms in order to assess the model's practical applicability in oyster mushroom cultivation.

Acknowledgments

The authors gratefully acknowledge the supervision and constructive guidance of Mrs. Ery Hartati during the course of this research.

Competing Interest

The authors declare no competing interest.

References

- [1] Windriyati, R.D.H.; Kurniawan, A.; Farisi, H. (2025). Edukasi pengelolaan hama dan penyakit terpadu pada pelaku usaha jamur tiram di Desa Penolih, Kaligondang, Purbalingga. *Madaniya*, 6, 1447–1454. <https://doi.org/10.53696/27214834.1326>.
- [2] Faturachman, I.; Kusumawati, R. (2024). Usaha budidaya jamur tiram. *Ikraith-Ekonomika*, 7, 65–70. <https://doi.org/10.37817/ikraith-ekonomika.v7i3.4250>.
- [3] Mona, N.; Widyastuti, D.A.; Nurwahyunani, A.; Hayat, M.S. (2022). Analisis permasalahan umur baglog dan hama penyerang pada budidaya jamur tiram (*Pleurotus ostreatus*) di Kabupaten Semarang. *Biodidaktika: Jurnal Biologi dan Pembelajarannya*, 17. <https://doi.org/10.30870/biodidaktika.v17i2.16497>.

- [4] Anggraeni; Salmi; Helmi, H.; Robika. (2023). Optimalisasi usaha budidaya jamur tiram di Desa Teru melalui kemandirian pengadaan media tumbuh dan penanaman bibit jamur. *Jurnal Pengabdian Masyarakat Borneo*, 7, 69–74. <https://doi.org/10.35334/jpmb.v7i1.2998>.
- [5] Ade, N.; Suranti, D.; Suryana, E. (2024). Implementasi metode certainty factor dalam mendeteksi hama dan penyakit pada jamur tiram putih. *Jurnal Media Infotama*, 20, 638–645. <https://doi.org/10.37676/jmi.v20i2.6570>.
- [6] Naveed, M.U.; Jabbar, S.; Iqbal, M.M.; Awais, A.; Alkhazi, I.S.; Alghamdi, M. (2026). A hybrid lung and colon histopathological image classification framework using MobileNetV3-Small deep features and differential evolution optimization. *Diagnostics*, 16, 1256. <https://doi.org/10.3390/diagnostics16091256>.
- [7] Christnatis; Lase, C.D.S.; Sitompul, T.H.; Hondro, A.P. (2024). Klasifikasi penyakit pada baglog jamur tiram menggunakan metode convolutional neural network. *Jurnal Pendidikan dan Teknologi Indonesia*, 4, 359–367. <https://doi.org/10.52436/1.jpti.480>.
- [8] Shi, G.; Wang, Y.; Yang, Z.; Guo, Y. (2024). Wildfire identification based on an improved MobileNetV3-Small model. *Forests*, 15, 1975. <https://doi.org/10.3390/f15111975>.
- [9] Athallah, D.F.; Budiman, T.; Sianipar, A.Z. (2025). Optimizer evaluation for maize leaf disease using transfer learning with MobileNetV3-Small. *Journal of Information Systems and Informatics*, 7, 1939–1954. <https://doi.org/10.51519/journalisi.v7i2.1144>.
- [10] Rakhmat, G.A.; Rizkiawarman, F. (2023). Implementasi arsitektur MobileNetV3 (studi kasus klasifikasi jamur beracun). *Prosiding Diseminasi FTI Itenas*, 3, 1–14. Available online: <https://eproceeding.itenas.ac.id/index.php/fti/article/view/3299>.
- [11] Zhao, L.; Wang, L. (2022). A new lightweight network based on MobileNetV3. *KSII Transactions on Internet and Information Systems*, 16, 1–15. <https://doi.org/10.3837/tiis.2022.01.001>.
- [12] Chin, P.-W.; Ng, K.-W.; Palanichamy, N. (2024). Plant disease detection and classification using deep learning methods: A comparison study. *Journal of Informatics and Web Engineering*, 3, 155–168. <https://doi.org/10.33093/jiwe.2024.3.1.10>.
- [13] Pratama, R.; Harvianti, Y.; Rahayu, R.I.; Safitri, A.W. (2025). Cultivation of white oyster mushroom (*Pleurotus ostreatus*) in Indralaya. *Sainmatika: Jurnal Ilmiah Matematika dan Ilmu Pengetahuan Alam*, 22, 164–175. <https://doi.org/10.31851/sainmatika.v22i2.20190>.
- [14] Imanulloh, S.B.; Muslikh, A.R.; Setiadi, D.R.I.M. (2023). Plant diseases classification based leaves image using convolutional neural network. *Journal of Computing Theories and Applications*, 1, 1–10. <https://doi.org/10.33633/jcta.v1i1.8877>.
- [15] Roseno, M.T.; Oktarina, S.; Nearti, Y.; Syaputra, H.; Jayanti, N. (2024). Comparing CNN models for rice disease detection: ResNet50, VGG16, and MobileNetV3-Small. *Journal of Information Systems and Informatics*, 6, 2099–2109. <https://doi.org/10.51519/journalisi.v6i3.865>.
- [16] Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; Le, Q.V.; Adam, H. (2019). Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324. <https://doi.org/10.1109/ICCV.2019.00140>.
- [17] Salka, T.D.; Hanafi, M.; Rahman, S.M.S.A.A.; Zulperi, D.M.; Omar, Z. (2025). Plant leaf disease detection and classification using convolution neural networks model: A review. *Artificial Intelligence Review*, 58, 322. <https://doi.org/10.1007/s10462-025-11234-6>.
- [18] Shorten, C.; Khoshgoftaar, T.M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 60. <https://doi.org/10.1186/s40537-019-0197-0>.

