Journal of
Electronics Technology Exploration

Homepage: htttps://shmpublisher.com/index.php/joetex
p-ISSN 3025-3470 | e-ISSN 3026-1066

Milkfish Freshness Detection Based On Eye Images Using Convolutional Neural Network (CNN) With Mobilenetv3 Architecture On A Mobile Application

Khalimah Musaadah*[1], Lasmedi Afuan[2], Ipung Permadi[3]
[1,2,3]Informatics, Engineering Faculty, Jenderal Soedirman University, Indonesia

Article Info

Article history:

Received Jan 22, 2026
Revised Jan 25, 2026
Accepted Jan 25, 2026

Keywords:

Hyperparameter tuning
Milkfish
Fish freshness
Mobilenetv3
Random

ABSTRACT

Indonesia has abundant fishery resources, making it one of the world's largest producers and consumers of fish. One of the most commonly consumed types is milkfish (Chanos chanos). Before consumption, it is important to determine the freshness level of the fish. This freshness can be identified using a Convolutional Neural Network (CNN) model with the MobileNetV3 architecture, which is efficient and suitable for mobile application implementation. This study aims to detect the freshness level of milkfish based on eye images using the MobileNetV3 CNN architecture implemented in a mobile application. The dataset used consists of 500 images, divided into training, validation, and testing sets with proportions of 70%, 20%, and 10%, respectively. The data underwent preprocessing, including resizing and image augmentation, to increase data variation. The model was developed using hyperparameter tuning with both random search and grid search methods. The results show that random search achieved better performance with a training accuracy of 92.88%, validation accuracy of 89.90%, and an overall test accuracy of 91%. The trained model was successfully implemented into a mobile application named ScanBang, which can classify the freshness level of milkfish and display its confidence score in a practical and user-friendly manner.

Corresponding Author:

Khalimah Musaadah,
Department of Informatics, Engineering Faculty,
Jenderal Soedirman University,
Purwokerto, Indonesia.
Email: khalimah.musaadah@mhs.unsoed.ac.id
https://doi.org/10.52465/joetex.v3i2.649

1. INTRODUCTION

Indonesia, as a maritime country, has abundant fishery resources, making it a significant producer and consumer of fish. One type of fish that is widely consumed in Indonesia is milkfish (Chanos chanos). This is evidenced by average fish consumption data, which shows that in 2022, fish consumption reached 12.268 kg

per capita [1]. Milkfish are highly adaptable to aquatic environments, making them easy to cultivate in various regions of Indonesia, especially in brackish water ponds [2]. Milkfish are also known to be beneficial to health because they are high in protein, low in fat, and contain omega-3 fatty acids that are good for the body [3].

Milkfish has advantages in terms of price and availability, making it widely consumed by the public [4]. However, its freshness must be considered because fish easily deteriorates due to bacterial growth. The freshness of fish can be identified by observing its physical characteristics. One simple way to check the freshness of fish is by looking at its eyes. Fresh fish generally have clear, convex, and shiny eyes. Conversely, fish that are starting to spoil will have cloudy and dull eyes [5].

Checking the freshness of fish, especially milkfish, is still largely done manually by visually observing the physical characteristics of the fish, such as the condition of the eyes, gills, and meat texture. This method relies on individual subjectivity, which can lead to bias in assessing fish freshness. Checking the freshness of milkfish is also closely related to Sustainable Development Goal (SDG) number 3, because consuming fish that is not fresh can hurt health, such as food poisoning due to contamination by bacteria or harmful substances. Therefore, technology is needed that can detect fish freshness automatically and objectively to ensure that the fish consumed is of high quality.

One solution that can be applied is the use of machine learning technology to detect fish freshness more accurately and objectively. Machine learning is a computational method that allows systems to learn from experience to improve their performance or make more accurate predictions [6]. In image processing to detect fish freshness, deep learning is one of the most effective approaches. Deep learning is a machine learning method that consists of an input layer, a hidden layer, and an output layer [7], [8]. One of the capabilities of deep learning is the use of CNNs, which can detect important elements in images, such as edges, textures, and object shapes, through a convolution-based learning process [9]. This model is highly flexible because it can be retrained to recognize new objects and allows for the development of more sophisticated models by utilizing existing networks [10]. CNNs have many architectures, such as VGGNet, AlexNet, GoogleNet, ResNet, MobileNet, EfficientNet, etc.

The above architectures have their own advantages. VGGNet, AlexNet, GoogleNet, and ResNet are known to have high performance in various image recognition tasks. However, these architectures also have quite high complexity, both in terms of the number of parameters and network depth [11]. On the other hand, MobileNet offers high efficiency with the depthwise separable convolution technique, which drastically reduces the number of parameters and computational requirements without significantly compromising accuracy [12]. MobileNet is suitable for mobile applications because it balances performance and efficiency, allowing it to run smoothly on devices with limited power and computation.

Although MobileNet is efficient for mobile devices, most previous studies on fish freshness detection use conventional CNNs or early-generation MobileNet variants. They rarely discuss these models' limitations in real-world, especially mobile app, implementations. Most studies also focus on general freshness analysis or lab-based conditions. In contrast, using fish eye images as a practical indicator of freshness remains limited [13], [14].

Furthermore, studies that compare the performance of lightweight models such as MobileNetV2 and EfficientNet-Lite in terms of computational efficiency and accuracy on mobile devices are rarely discussed comprehensively [15], [16]. Therefore, this study proposes using the MobileNetV3 architecture to detect the freshness of milkfish from fish eye images. MobileNetV3 integrates neural architecture search and attention mechanisms to improve accuracy while maintaining computational efficiency. This makes it more suitable for implementation on mobile devices than previous generations [17].

Additionally, this study evaluates the effect of hyperparameter optimization using random search and grid search methods to enhance model performance. The resulting model is then implemented as a mobile application to test its effectiveness in real-world conditions. Thus, this study not only contributes to improving the accuracy of fish freshness detection but also provides a practical and ready-to-use solution to support accurate fish quality assessment.

2. METHOD

This study uses the CNN method with MobileNetV3 architecture for the classification process in detecting the freshness of milkfish. The research stages consist of several steps shown in Figure 1.

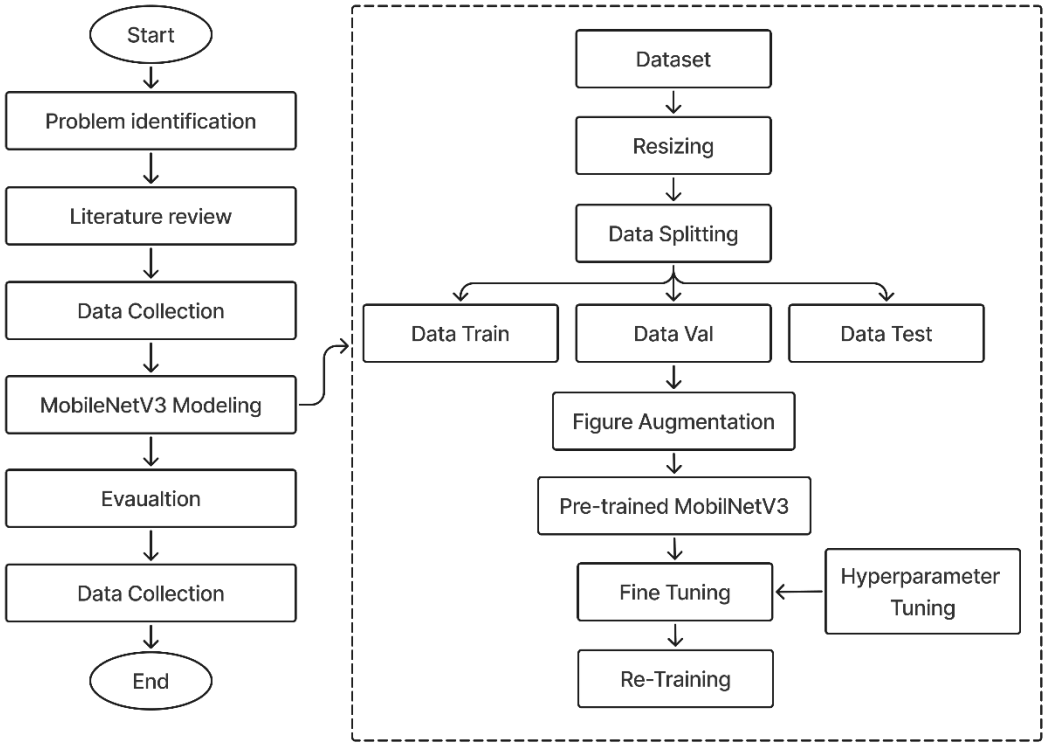

Figure 1. Research stage flow

2.1. Problem Identification

Problem identification is the initial stage of research that aims to determine the problems to be solved. In this study, the problem raised is the assessment of fish freshness, which is still done manually by relying on visual observation, resulting in subjective and inconsistent assessments. Therefore, this study developed an application that can automatically detect fish freshness using a CNN with MobileNetV3 architecture.

2.2. Literature Study

Literature study is a research stage that aims to collect and analyze various relevant references to gain a deeper understanding of the topic being studied. Literature study is conducted by reading and collecting reliable sources such as books, materials, journals, and previous research results that are similar to this study.

2.3 Data Collection

Data collection is a stage in research that aims to obtain a dataset that will be used in the model training and testing process. Data collection in this study used secondary data taken from previous studies uploaded to the Mendeley data platform [13]. The dataset consists of 4392 images covering 8 types of fish, namely milkfish, tilapia, nila fish, samge fish, kuniran fish, senangin fish, mackerel, and gulamah fish. This study will only take the milkfish dataset with 500 images categorized into three levels of freshness based on the length of storage of the fish, as shown in Table 1.

Table 1. Fish freshness levels

Storage Duration	Freshness Category
1 - 2 days	Very Fresh
3 – 4 days	Fresh
5 – 6 days	Less Fresh

2.4 MobileNetV3 Modeling

MobileNetV3 modeling in this study includes several key mechanisms that must be carried out systematically. These mechanisms include the following steps.

2.4.1 Resizing

Resizing is the process of changing the image size to suit the model's requirements before it is used in training. In this study, resizing was performed to adjust the size of the fish eye image to the MobileNetV3

input standard of 224 x 224 pixels. Resizing was performed using the bilinear interpolation method, which is a resampling method that uses the four nearest neighboring pixels to calculate the new pixel value [14].

$$f(x,y) = W_1 f_1 + W_2 f_2 + W_3 f_3 + W_4 f_4 \quad (1)$$

In equation (1), f(x,y) is the interpolated pixel value, W is the weight of each neighboring pixel, f_1 is the pixel value at the upper left point (x_1,y_1), f_2 is the pixel value at the upper right point (x_2,y_1), f_3 is the pixel value at the lower left point (x_2,y_1), f_4 is the pixel value at the lower right point (x_2,y_2).

2.4.2 Split Data

After the resizing process is complete, the image data will be divided into three main parts, namely 70% train data to train the model, 20% val data to evaluate the model's performance, and 10% test data to test the final performance of the model with data that has never been used before.

2.4.3 Image Augmentation

Image augmentation is a technique used to expand and enrich the dataset to improve the model's ability to recognize image variations and prevent overfitting during the training process. Image augmentation is performed using 6 techniques such as image rotation, flipping (mirroring), brightness adjustment, contrast adjustment, zoom, and translation (image position shifting).

2.4.4 Pre-trained MobileNetV3

Pre-trained MobileNetV3 is a model that has been trained using a large dataset called ImageNet, which contains around 1000 different object classes [18]. This model has the ability to recognize various patterns and features from many types of images, so it can be used to detect fish freshness without the need to train the model from scratch.

2.4.5. Fine Tuning

Fine-tuning is the process of refining a model by opening the last few layers of a pre-trained model and adjusting them to the dataset used [19]. Fine-tuning aims to improve model performance by allowing certain parts of the model to relearn specific data used in the research.

2.5. Evaluation

Evaluation is the final stage in this study, which aims to measure the performance of the developed model. In this study, evaluation was carried out using several metrics, namely accuracy, precision, recall, and F1-score. The use of several evaluation metrics is intended so that the assessment of model performance does not depend on one measure, but can provide a more comprehensive picture of the model's ability to perform classification.

Accuracy is used to indicate the model's accuracy in classifying all test data as a whole. However, accuracy alone is not sufficient to represent the model's performance as a whole, especially in cases of classification with an unbalanced data distribution or when classification errors have a significant impact. Therefore, the precision and recall metrics were also used in this study.

Precision describes the model's ability to produce correct predictions for a particular class, thus reflecting the reliability of the classification results. Meanwhile, recall shows the model's ability to detect all data that should belong to that class. In the context of fish freshness detection, the recall value is important to minimize errors in classifying fish that are no longer fresh as fresh fish, which can have an impact on the quality and safety of consumption [20].

Furthermore, the F1-score is used as the harmonic mean between precision and recall. This metric provides a more balanced evaluation, especially when there are differences between precision and recall values. The use of F1-score is widely recommended in classification research because it can represent model performance more objectively than using accuracy alone [21]. Thus, the combination of these four metrics allows for a more comprehensive and reliable evaluation of the model.

2.6. Mobile Application Implementation

After the CNN model with MobileNetV3 architecture was trained and evaluated, it was then implemented into a mobile application using TensorFlow Lite (TFLite). TensorFlow Lite was chosen because it is capable of running deep learning models efficiently on devices with limited resources, making it suitable for implementation on smartphones.

The application was tested using a Samsung Galaxy A25 5G device with the following specifications: Samsung Exynos 1280 processor, 8 GB RAM, and 256 GB storage. This device is equipped with a 50 MP (wide), 8 MP (ultra-wide), and 2 MP (macro) rear camera, as well as a 13 MP front camera, and runs on the

Android 15 operating system. Fish eye images were captured using the rear camera and then processed by the CNN model that had been converted to TFLite format.

3. RESULTS AND DISCUSSIONS

This section presents the research results and analysis in a structured manner based on the stages of the research methodology.

3.1. Dataset Collection

In this study, the milkfish image dataset was obtained from previous research [13], which had eight types of fish with three levels of freshness each. However, this study only used milkfish images, with a total of 500 images consisting of 3 freshness levels, namely very fresh, fresh, and less fresh. There were 168 images in the very fresh class, 162 images in the fresh class, and 170 images in the less fresh class.

3.2. Preprocessing

The preprocessing stage has several sub-stages as follows.

3.2.1. Resizing

Resizing was done to standardize the size of all images, because the milkfish eye image dataset had varying sizes. The following are the descriptive statistics of the image size before resizing.

Table 2. Descriptive statistics of the original image size

Class	Width			Height		
	Mean	Min	Max	Mean	Min	Max
Fresh	162	191	441	307	199	298
Highly Fresh	331	199	583	328	203	566
Not Fresh	289	166	398	286	165	379

Table 2 shows that the image sizes in the milkfish eye image dataset vary. Therefore, the images need to be resized or normalized before being used in model training so that the model is not biased towards image size. The images will be resized to 224x224 according to the size in MobileNetV3.

3.2.2. Data Splitting

The data splitting process is carried out to divide the milkfish eye image dataset into 3 parts, namely train data, validation data, and test data. This division aims to allow the model to be trained, validated, and tested separately to avoid overfitting. Each image in each class will be separated with 70% for training data, 20% for validation data, and 10% for test data. This section describes the results of the research and testing that have been carried out. The results of data splitting are shown in Table 3.

Table 1. Data splitting result

Class	Train	Val	Test
Highly Fresh	117	33	18
Fresh	113	32	17
Not Fresh	118	34	18

After the splitting process is complete, the next step is the data loading process, which aims to enable the images to be used directly in the TensorFlow training pipeline.

3.2.3. Image Augmentation

After the images are successfully loaded, the next step is to perform image augmentation to increase the variety in the milkfish eye image dataset. Image augmentation aims to enrich the training data so that the model is more robust against image variations. In this study, there are six types of augmentation, namely image rotation, flip (reflection), brightness adjustment, contrast adjustment, zoom, and translation (image position shift). Image augmentation is performed using Tensorflow Keras.

a. Image Rotation

Image rotation is performed by rotating the image up to a maximum of 20% of 360%. The results of image rotation are shown in Figure 2.

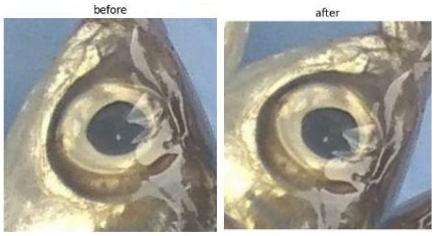

Figure 1. Rotation image results

b. Image Flipping

Image flipping is performed using Tensorflow Keras, which by default will flip the image horizontally or vertically. The results of image flipping are shown in Figure 3.

Figure 3. Image flipping results

c. Brightness Adjustment

Brightness adjustment is done by changing the image to be 20% brighter or 20% darker than the original image. The results of the image brightness adjustment are shown in Figure 4.

Figure 4. Brightness adjustment results

d. Image Contrast Change

Image contrast adjustment is done by filling in the parameter factor = 0.2, which means that there will be a 20% change in contrast from the original image. The results of the contrast change can be seen in Figure 5.

Figure 5. Image contrast change results

e. Image Zoom

Image zoom uses height_factor = -0.2 and 0.2, then width factor = -0.2 and 0.2. The value 0.2 means that the image will be enlarged and reduced in height and width by 20%. The results of image zoom can be seen in Figure 6.

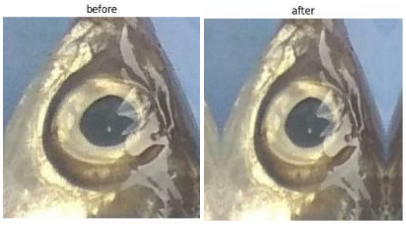

Figure 6. Image zoom results

f. Image translation

Translation augmentation is performed by shifting the image 10% from its original size to the right, left, top, and bottom. The results of image translation can be seen in Figure 7.

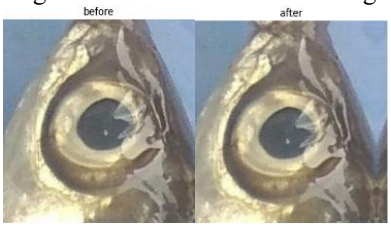

Figure 2. Translation image results

3.3. MobileNetV3 Modeling

After going through the preprocessing stage, the next step is to perform modeling using the MobilenetV3 architecture. The purpose of this modeling is so that the model can learn and recognize the freshness level of milkfish based on images of their eyes. The modeling process begins with a pretrained MobileNetV3, output layer adjustment according to the number of classes, fine-tuning, hyperparameter tuning, and validation. Model training is carried out using training data, while validation data is used for the validation process to monitor model performance during training.

3.3.1. Pretrained MobileNetV3

At this stage, the modeling process was carried out using the MobileNetV3 architecture that had been pre-trained with weights from ImageNet. However, the weights from the pre-trained model were not used entirely. The model underwent a fine-tuning process and architectural modifications by adding two fully connected layers before the final classification layer.

The two additional layers consist of a combination of dense layers, Dropout, L2_regularization, and batch normalization, where the values will be determined through a hyperparameter tuning process. The complete structure of the modified MobileNetV3 architecture can be seen in Figure 8.

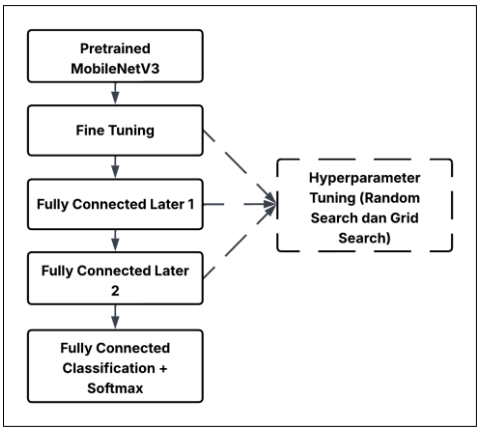

Figure 3. Pretrained mobilenetv3 modification structure

3.3.2. Fine Tuning

After the model compilation process is complete, the next step is to fine-tune the MobileNetV3 architecture. Fine-tuning is the process of retraining some layers of the pre-trained model to adapt it to the characteristics of the data used. This process begins by reactivating all layers of MobileNetV3 so that they can be retrained. However, only some of the upper layers (close to the output) will remain active for training, while the initial layers (close to the input) will be locked again so that they do not change. The starting point of the fine-tuning process will be determined through hyperparameter tuning, which involves finding the layer where retraining will begin. The search is carried out starting from layer 150 to the last layer before the fully connected layer, with an increment (step) of 10 layers.

3.3.3. Hyperparameter Tuning

After the fine-tuning process is complete, the next step is to perform hyperparameter tuning to find the best combination of parameters that can improve model performance. In this study, two approaches were used, namely Random Search and Grid Search, to compare the best parameter combinations that can improve model

accuracy. The values sought through hyperparameter tuning are the fine-tuning layer, dropout, units (number of neurons in the fully connected layer), and L2_regularization.

a. Random Search

At this stage, hyperparameter tuning is used with the random search method, which is a random search for hyperparameters without having to try all possibilities, thus saving time and resources. The results of the random search hyperparameter combination can be seen in Table 4.

Table 2. Hyperparameter random search results

Best Hyperparameters	Value
Fine-tune	170
Dropout 1	0.5
Dropout 2	0.2
Dropout 3	0.1
Units 1	384
Units 2	192
L2 Regularization	0.001

b. Grid Search

The second method used in the hyperparameter tuning process is grid search, which involves searching for hyperparameters by trying all combinations. The results of the grid search hyperparameter combinations can be seen in Table 5.

Table 3. Hyperparameter grid search results

Best Hyperparameters	Value
Fine-tune	150
Dropout 1	0.3
Dropout 2	0.2
Dropout 3	0.1
Units 1	128
Units 2	96
L2 Regularization	0.000001

3.3.4. Re-Training

Re-training is the process of retraining the model using the best parameter value combinations found through the random search and grid search methods. These parameter values include hyperparameter configurations such as the number of units in the Dense layer, the dropout rate, and the magnitude of the penalty in L2 regularization. At this stage, the re-training process is carried out for 100 epochs for each combination of hyperparameter tuning results. The aim is to ensure that the model has enough time to learn from the data thoroughly with the optimal parameter configuration. During retraining, the evaluation metric used to determine model performance is validation accuracy (val_accuracy). The best val_accuracy value from each experiment will be recorded, and the model with the highest performance will be selected as the final model used for the next evaluation stage. The results of re-training each model are shown in Figures 9 and 10.

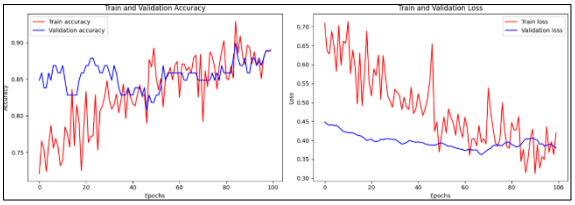

Figure 9. Random search hyperparameter tuning re-training graph

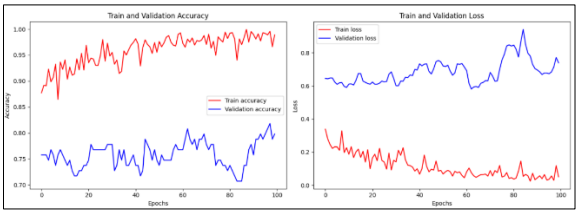

Figure 10. Grid search hyperparameter tuning re-training graph

The graph in Figure 9 shows that at epoch 85, the model achieved its best performance with a train_accuracy value of 0.9288, train_loss of 0.3445, val_accuracy of 0.89899, and val_loss of 0.3889. After

epoch 85, no further increase in val_accuracy was found until the end of training. Therefore, the best model used and saved is the model at epoch 85 because it has the highest validation performance throughout training.

Meanwhile, the graph in Figure 10 shows that at epoch 98, the model achieved its best performance with train_accuracy 0.9957, train_loss 0.0294, val_accuracy 0.8182, and val_loss 0.7146. After epoch 98, there was no further increase in val_accuracy until the end of training. Therefore, the best model used and saved is the model at epoch 98 because it has the highest validation performance throughout the training.

Based on the grid search model results graph, there are indications of overfitting, where the graph shows a significant gap between the very low train loss and the relatively high validation loss, which tends to increase. In addition, train accuracy continues to increase, almost reaching the perfect score, while validation accuracy is around 0.82 and does not show a significant upward trend. This difference in patterns indicates that the model is too adapted to the training data and is unable to generalize well to the validation data. This condition is caused by the grid search process, which, in principle, searches for hyperparameters by trying all possible combinations. However, in this study, only 10 hyperparameter combinations were searched to save computation time, so the best combination may not have been fully discovered, affecting the final performance of the model.

Therefore, the best model used in this study is the model generated from a combination of hyperparameters using the random search method. In addition to producing higher validation performance, this method is also very suitable for application in large and complex parameter spaces and performs the search for the optimal combination of parameters in a shorter time. A comparison of the performance between the two hyperparameter search methods is presented in more detail in Table 6.

Table 4. Performance comparison between random search and grid search

Hyperparameter	Train_Acc	Val_Acc	Train_Loss	Val_Loss
Random Search	0.9288	0.8990	0.3445	0.3889
Grid Search	0.9957	0.8182	0.0294	00.7146

3.4. Evaluation

The next stage is evaluation, presented in the form of a confusion matrix that provides an overview of the number of correct and incorrect predictions in each class. In addition to the confusion matrix, other evaluation metrics such as accuracy, precision, recall, and F1-Score are also used to measure model performance more clearly. The confusion matrix is shown in Figure 11.

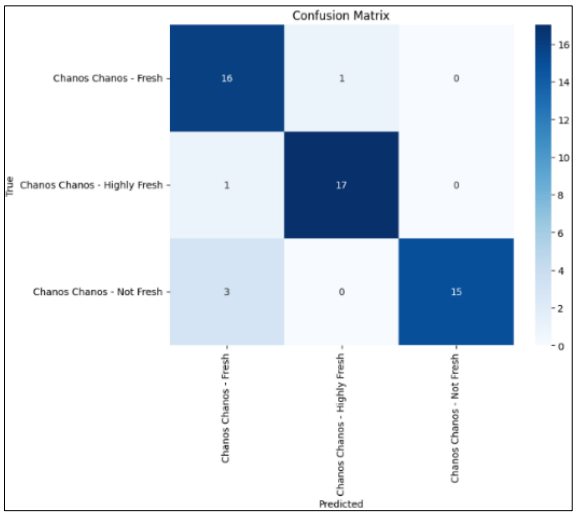

Figure 11. Confusion matrix

From the confusion matrix in Figure 25, the values shown in Table 7 are obtained.

Table 5. TP, FP, TN, FN statistics per class

Class	TP	FP	TN	FN
Fresh	16	4	32	1
Highly fresh	17	1	34	1
Not Fresh	15	0	35	3

After obtaining the TP, FP, TN, and FN values for each class, the next step is to calculate the evaluation metrics, including accuracy, precision, recall, and F1-score, which are presented in the form of a classification report in Table 8.

Table 6. Classification report

	Accuracy	Precision	Recall	F1-Score	Support
Fresh	0.90	0.80	0.94	0.86	17
Highly Fresh	0.96	0.94	0.94	0.94	18
Not Fresh	0.94	1.00	0.83	0.91	18
Accuracy				0.91	53
Macro avg		0.91	0.91	0.91	53
Weighted avg		0.92	0.91	0.91	53

Table 8 can be interpreted as follows:

a. Accuracy

The accuracy for each class is≥ 0.90, and the overall accuracy is 0.91. This indicates that the model performs well in classifying the freshness level of milkfish based on eye images.

b. Precision

From the classification report results, the highest precision is found in the Chanos Chanos – Not Fresh class with a value of 1.00, which means that all predictions for this class are correct. Meanwhile, the precision in the Chanos Chanos – Highly Fresh and Chanos Chanos – Fresh classes is 0.94 and 0.80, respectively. This shows that most of the model's predictions for these two classes are on target.

c. Recall

The Chanos Chanos – Highly Fresh and Chanos Chanos – Fresh classes both have a recall value of 0.94, indicating that the model is able to recognize most of the data from both classes. However, the recall for the Chanos Chanos – Not Fresh class is slightly lower at 0.83, meaning that there is still some data that is missed or misclassified.

d. F1-Score

The F1-score results for all classes show high values, namely 0.86 (Chanos Chanos - Fresh), 0.94 (Chanos Chanos - Highly Fresh), and 0.91 (Chanos Chanos - Not Fresh), indicating that the model generally has good and balanced classification performance in all classes.

3.5. Mobile Application Implementation

At this stage, the MobileNetV3 classification will be implemented into a mobile application using the Flutter framework. The mobile application implementation stages include several sub-stages, namely creating a use case diagram, designing the interface using Figma tools, mobile-based implementation, and testing the application using blackbox testing.

3.5.1 Use Case Diagram

A use case diagram is a type of diagram used to describe the interaction between actors (users) and the system being developed. The use case for the application for identifying the freshness of milkfish based on eye images is shown in Figure 12.

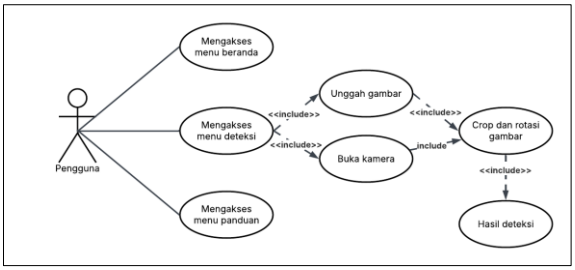

Figure 4. Use case diagram for milkfish freshness identification

The use case diagram, Figure 12, shows the interaction between actors and their various features. Users can access the home page, which displays information about the application and the freshness of the fish. Then, they can access the detection menu, where users are asked to upload images from the gallery or capture images from the camera to detect the freshness of milkfish based on the image of its eyes. In addition, users can access the guide menu, which contains steps for performing the detection.

3.5.2 Mobile-based application implementation

The next step is to implement the mobile application using the Flutter framework. At this stage, all the designs that have been made previously will be realized in the form of a runnable application. The application resulting from this process is named "ScanBang," which is a combination of the words 'scan' (detection) and "bang," short for milkfish. The following are the results of the mobile application implementation.

a. Home Page

The results of the mobile application implementation for the home page can be seen in Figure 13.

Figure 13. Home page

In Figure 13, the home page functions as the main display that presents general information about the application that has been developed. The purpose of this page is for users to understand the uses and benefits of this application, particularly in helping to detect the freshness level of fish. In addition, the home page also displays information about the freshness level of fish and its characteristics so that users can recognize the differences between very fresh, fresh, and not fresh fish. This page also provides information about the features available in the application to give users an idea of the advantages of the application.

b. Detection Page

The results of the implementation of the mobile application homepage can be seen in Figure 14.

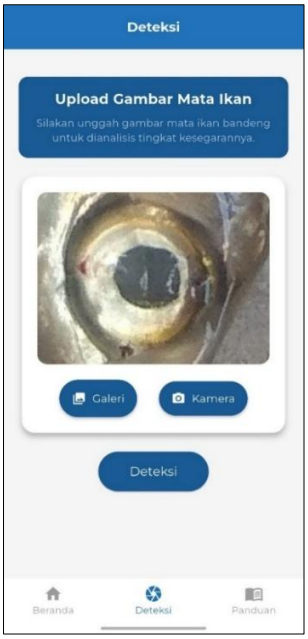

Figure 14. Detection page

In Figure 14, the detection page functions to analyze the freshness level of fish based on images of their eyes. On this page, users are asked to upload images from their gallery or capture images directly using the camera. Once the image is obtained, users will be directed to adjust the position of the image, especially in the area of the fish's head that includes the eye, using the zoom or rotation features. After the adjustment process is complete, the image will be displayed on the detection page before the classification process is carried out to determine the level of freshness of the fish.

c. Detection Results Page

The detection results page is used to display the output of the classification process performed by the model. The image previously uploaded by the user will be displayed again on this page. Next, the image is processed by the MobileNetV3 model, which then produces the classification results for the fish's freshness level along with its confidence level. In addition to displaying the main classification results, this page also presents probability values for each class, so that users can see how likely an image is to belong to a particular category. The results of the implementation of the mobile application for the detection results page can be seen in Figure 15.

Figure 15. Detection results page

d. Guide Page

The results of the implementation of the mobile application guide page can be seen in Figure 16.

Figure 16. Guide page

The guide page is designed to provide information and instructions to users on how to use the application correctly.

3.5.3 Application Testing

After the application has been developed, the next step is to conduct a testing process to ensure that the application can run properly according to the designed functionality. The testing method used is blackbox testing, which focuses on evaluating whether each feature in the application can function as expected. The results of the blackbox testing are presented in Table 9.

Table 9. Testing results

No	Test Description	Expected Result	Test Result	Conclusion
1	Pressing the gallery button to upload an image	The image can be uploaded and the application navigates to the image editing page	The application successfully uploads the image and navigates to the image editing page	Successful
2	Pressing the camera button to capture an image	The application can capture an image using the camera and navigate to the image editing page	The application successfully captures an image and navigates to the image editing page	Successful
3	Pressing the check button on the edit page to display the edited image	The application returns to the detection page and displays the edited image	The application successfully returns to the detection page and displays the edited image	Successful
4	Pressing the detection button to display detection results	The application navigates to the result page and displays the classification result and confidence value	The application successfully displays the detection result along with the confidence value	Successful

DISCUSSION

This study successfully developed and applied a Convolutional Neural Network (CNN) model with MobileNetV3 architecture to identify the freshness level of milkfish based on eye images, which was then

implemented into a mobile application. The developed model was able to achieve an accuracy of more than 90% without overfitting, demonstrating stable performance in classification. When compared to the research by Prasetyo et al. (2022), which used the MobileNetV1 architecture, the accuracy obtained was 63.21%, far below the ideal standard of 70% [14]. The low accuracy is thought to be due to the limited amount of data sets, so that the model could not learn optimally.

Meanwhile, another study by Nasrul et al. (2022) using the MobileNetV2 architecture produced an accuracy of 95% for the very fresh class, 70% for the fresh class, and 80% for the less fresh class [11]. This study adopted optimization using a momentum optimizer with learning rate settings based on Cosine Annealing and Warm Restarts (SGDR) for 5000 steps. Although the accuracy for the very fresh class is quite high, the model's performance in other classes is still lower than in this study. This is due to the model's limitations in recognizing specific parts of the fish's body, such as only the head or only the tail, which affects the accuracy in the fresh and less fresh classes.Thus, it can be concluded that the MobileNetV3 model used in this study has superior and stable performance in classifying the overall freshness of milkfish.

Although the research in this paper has produced high and stable accuracy, there are still several shortcomings that need to be considered for further development. One of the suggestions is to apply a comprehensive grid search method, which is to try every possible parameter combination one by one. This approach is important so that each configuration can be evaluated and compared systematically, thereby enabling the discovery of parameter combinations that produce the best performance.

In addition, to improve the system's capabilities, it is recommended that the model be further developed by combining the MobileNetV3 architecture with object detection methods such as YOLO (You Only Look Once) or SSD (Single Shot Multibox Detector). With this approach, the system will not only be able to classify one object in one image, but also detect and classify several objects simultaneously in real-time. The use of object detection methods also allows the display of bounding boxes that automatically identify the location of fish eyes in images. This will make the detection process more efficient, accurate, and suitable for application in real-world situations.

4. CONCLUSION

This study aims to classify the freshness level of milkfish through eye images. A total of 500 images were used and divided into training data (70%), validation data (20%), and test data (10%). All images were processed through preprocessing stages, such as resizing to 224×224 pixels and augmentation (rotation, flipping, brightness adjustment, contrast, zoom, and translation) to increase data variation. The classification model used was MobileNetV3 with the application of hyperparameter tuning techniques using the random search and grid search methods. Based on 10 trials over 100 epochs, random search showed better performance without any indication of overfitting, with a train accuracy of 92.88% and a validation accuracy of 89.90%. The test results showed that the model was able to accurately classify fish freshness, with the highest F1-score of 0.94 in the very fresh class. The fresh and not fresh classes obtained f1-scores of 0.86 and 0.91, respectively, and the overall accuracy of the model reached 91%. The model that was built was then successfully implemented into a mobile application called ScanBang, which provides features for uploading images from the gallery or capturing images from the camera, and provides complete classification results with prediction confidence values.

REFERENCES

[1] T. D. Astari, K. Sulandjari, and A. Bakar, “Pengaruh Input Terhadap Produksi Ikan Bandeng di Kecamatan Tirtajaya Kabupaten Karawang,” --, 2023.

[2] M. Ardini, C. Gustiana, and S. Anzitha, “Analisis Pengaruh Preferensi Konsumen terhadap Keputusan Pembelian Ikan Bandeng (Chanos Chanos),” *J. Inov. Penelit.*, vol. 2, no. 1, pp. 3765–3772, 2022.

[3] R. Pratama, “Rancang Bangun Alat Pendeteksi Kesegaran Ikan Berbasis Internet of Things (IoT),” *Jupiter*, vol. 2, no. 3, pp. 117–131, 2024.

[4] B. A. Magdalena, “Pengaruh Persepsi Harga, Kualitas Produk dan Lokasi terhadap Keputusan Pembelian Ikan Bandeng.” 2023.

[5] M. Muchtar, Y. P. Pasrun, R. Rasyid, N. Miftachurohmah, and M. Mardiawati, “Penerapan Metode Naive Bayes dalam Klasifikasi Kesegaran Ikan Berdasarkan Warna,” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 1, 2024.

[6] R. Nuraini and A. Amir, “Identifikasi Ikan Sebagai Protein Hewani Pencegah Stunting dengan Pendekatan Machine Learning,” *Joutica*, vol. 9, no. 2, 2024.

[7] G. A. Rakhmat and M. F. Haekal, “Peningkatan Performa MobileNetV3 dengan Squeeze and Excitation,” *MIND J.*, vol. 8, no. 1, pp. 27–41, 2023.

[8] A. M. F. M. Natsir, A. Achmad, and H. Hazriani, “Klasifikasi Ikan Tuna Layak Ekspor Menggunakan CNN,” *JISTI*, vol. 6, no. 2, pp. 172–183, 2023.

[9] R. A. Sutiani, N. Made, S. Ulandari, and R. A. Saputra, “Klasifikasi Kesegaran Ikan Menggunakan CNN Arsitektur VGG-16,” *JOINTER*, vol. 5, no. 2, pp. 30–35, 2024.

[10] A. Nasrul, E. Prasetyo, and R. Purbaningtyas, "Aplikasi Identifikasi Kesegaran Ikan Bandeng menggunakan CNN," *J. Electr. Eng. Comput. Sci.*, 2022.
[11] M. R. Alwanda, R. Putra, K. Ramadhan, and D. Alamsyah, "Implementasi CNN LeNet-5 untuk Pengenalan Doodle," 2020.
[12] M. Z. Basri, M. G. Somoal, and R. S. Aji, "Deteksi Tumor Otak pada Citra MRI Menggunakan MobileNet," *JEKIN*, vol. 5, no. 2, 2025.
[13] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, N. Suciati, and C. Fatichah, "Combining MobileNetV1 and Depthwise Separable Convolution," *Inf. Process. Agric.*, vol. 9, no. 4, pp. 485–496, 2022.
[14] M. S. A. Aziz and B. Chourasia, "Adaptive Image Resizing using Edge Contrasting," *Int. J. Trend Sci. Res. Dev.*, vol. 4, no. 6, pp. 1711–1716, 2020.
[15] M. Tan and Q. V Le, "EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks," 2019.
[16] A. Howard, W. Wang, G. Chu, L. Chen, B. Chen, and M. Tan, "Searching for MobileNetV3."
[17] A. G. Howard and W. Wang, "Applications," 2012.
[18] D. Lizard, S. Dimara, B. Rahmat, and H. Maulana, "Identifikasi Penyakit Daun Padi Menggunakan MobileNet-SVM," *J. Mhs. Tek. Inform.*, vol. 9, no. 3, 2025.
[19] S. Agustiani, R. Aryanti, S. K. Wildah, Y. T. Arifin, S. Marlina, and T. Misriati, "Optimisasi Model Deep Learning untuk Deteksi Penyakit Daun Tebu," *J. Informatics Manag. Inf. Technol.*, vol. 4, no. 4, pp. 150–157, 2024.
[20] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.
[21] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2012.