

# Classification Freshness of Red Snapper (*Lutjanus Campechanus*) Based on Eye Image Using Convolutional Neural Network

Muh Subhan<sup>a,\*</sup>, Nursakinah<sup>b</sup>

<sup>a</sup>Informatic Management Studi Program, Politeknik Negeri Fakfak, 98011, Papua Barat, Indonesia. Email: subhan@polinef.id

<sup>b</sup>Informatic Management Studi Program, Politeknik Negeri Fakfak, 98011, Papua Barat, Indonesia. Email: nursakinah@polinef.ac.id

---

## Abstract

Indonesia is a maritime country where fish is the most widely extracted and consumed marine natural resource, one of which is snapper. Snapper contains high protein. Therefore, it is suitable for health. Red snapper or *Lutjanus campechanus* is one economical fish with a broad market share. Red snapper is a demersal fish group that ranks third with the most exported commodities after tuna and shrimp. In addition, snapper is one of the most common consumption fish in Indonesia. Therefore, the community needs to be able to identify the freshness of the fish. Fish freshness detection is done manually by touching the fish's body, eyes, and gills. However, this can cause accidental damage to the fish parts, which will be very detrimental. Several studies on identifying fish freshness explain that the VGGNet-16 Architecture on the Convolutional Neural Network algorithm is superior in its modeling performance. This research uses a different fish object, a red snapper object, with two different architectures from several previous studies, namely the Le-Net15 and VGGNet-16 architecture. This research focuses on the eye image carried out through the pre-processing data stage by cutting the fish body, followed by augmentation to reproduce the image data without losing its essence before training the dataset. The model will be trained using the Adam optimization method with very fresh and not fresh predictions. The experimental results of the classification of two classes of red snapper freshness using 600 fish images show that VGGNet-16 achieves the best performance compared to the LeNet-5 architecture, where the classification accuracy reaches 98.40%.

*Keywords: Red Snapper; CNN; neural network; deep learning; VGGNet-16*

---

## 1. Introduction

Indonesia is a maritime country where the marine natural resources most often taken and consumed are fish, one of which is red snapper. Red snapper, or *Lutjanus campechanus*, is a demersal fish that can live in shallow to deep seas. According to the Central Statistics Agency (BPS), national *Lutjanus campechanus* production was recorded at 1.95 thousand tons in 2021. *Lutjanus campechanus* is an economically important type of fish that belongs to the demersal fish group and ranks third in terms of the largest export commodity after tuna and shrimp. In addition, *Lutjanus campechanus* is one of the most common consumption fish found in Indonesia, so the public needs to be able to identify the freshness of the fish.

The quality of fresh fish is characterized by clear eyes, clear corneas, black pupils, convex eyes, and fresh red gills. If the quality decreases, the gills are gray, slimy, and smelly; the scales are strongly attached, shiny, and covered with clear mucus. The smell is typical of fish [1]. The level of freshness of fish is generally identified

manually using eye observation, so it is challenging for the community to distinguish the fish's freshness level. In addition, the freshness of the fish can be identified by touching the fish's body, eyes, and gills, but this can cause accidental damage to the fish, which will be very detrimental.

Many studies on the classification of fish freshness have been carried out, one of which uses non-destructive image processing techniques using fish skin as a focused network. The skin tissue was segmented using the saturation channel of the HSV color space model. Feature statistics were extracted in the HSV color space that provided the fish freshness degradation pattern, which was used to design a framework for fish freshness identification. The result of the maximum classification accuracy of this method is 96.66% [2]. Identification of the freshness of the gill fish tissue is also carried out with an automatic image processing approach by performing. Features have been extracted from the automatically segmented gill focal tissues using Wavelet Transform. The gill tissue of fresh fish is reddish brown. The changing color of the fish tissue indicates fish damage. In the proposed methodology, the gill focus network is taken by region of interest (ROI), the image segment that carries the

---

\*Corresponding author. Tel.: +62-823-112-66360

Imam Bonjol Atas Street  
Fakfak-West Papua  
Indonesia, 98612

complete information about feature extraction. From the input RGB fish image, the gills are segmented as ROI because they have complete information because of their reddish-brown color. A Non-Destructive Technique evaluates material properties without causing damage. These discriminatory features from the experiment establish a relationship between the statistical wavelet coefficients and the freshness of stored fish [3]. In addition, the classification of fish freshness using several fish samples, namely Giant Gourami, Red Snapper Fish, and Nile Tilapia, was carried out using digital images with the K-Nearest Neighbor approach producing an average accuracy of 91.36% [4].

In addition, research using the Convolutional Neural Network (CNN) algorithm approach is currently a widely developed research topic, including identifying or classifying fish freshness. Research related to the classification of the freshness level of milkfish was carried out by comparing several architectures, namely Xception, MobileNet V1, Resnet50, and VGG16. The experimental results of the classification of two classes of milkfish freshness using 154 images show that VGG16 achieves the best performance compared to other architectures, where the classification accuracy reaches 97% [5]. The study used a Deep Convolution Neural Network (DCNN) approach to detect the freshness of sardine samples and classify fish samples as fresh fish or rotten fish. The automatic detection system was implemented, evaluated, and obtained results of 99.5% accuracy, 96.2% sensitivity, 92.3% specificity, 92.6% PPV, 96% NPV, and 94% f1-score. Using several stages, including pre-processing data, namely Image Rescaling Color Transformation, then the distribution of testing data and training data, then classification using the Deep CNN approach [6]. Another study that implemented fish freshness detection using a convolutional neural network (CNN) approach was carried out to detect goldfish freshness. A VGG-16 architecture was applied to extract features from FSH images automatically. Then, the developed classifier block is constructed by dropout, and a solid layer is used to classify the FSH image. The results indicate a classification accuracy of 98.21%, and the conclusion is that the CNN-based proposal has lower complexity with higher accuracy than traditional classification methods [7]. In another study on freshness detection using fish samples, Nile Tilapia employs an automated method for classifying fish freshness based on a combined deep learning model and image processing. The process extracts features using VGG-16 neural network architecture, and bi-directional long-short-term memory is used to build a machine learning model. The proposed model has achieved 98% accuracy in testing [8].

This study aims to develop software to read and analyze fish images and then automatically predict whether the image is fresh fish or not fresh fish, using two different architectures from previous studies, namely the LeNet-5 and VGGNet-16 architectures. This experiment uses the red snapper object, which consists of image acquisition, pre-processing image, augmentation, and utilizing the holdout method.

## 2. Material and Method

### 2.1. System design

Figure 1 shows this study's system design, which consists of Image Acquisition, Pre-processing data, Augmentation classification using Convolutional Neural Network Algorithm Performance analysis method of classification, and Algorithm Performance Result. Implementation of fish freshness classification uses Python programming language to create models and is assisted by the Tensorflow library, which is one of the most famous Python libraries for creating Deep Learning models.

### 2.2. Image acquisition

Image data of *Lutjanus campechanus* was obtained from the Dulan Pokpok Fisheries Port, Jl. Yos Sudarso, Dulan Pok-Pok Village, Wagom Village, Kec. Fak-Fak, Fak-Fak Regency. The image of the fish is taken from various angles using the camera Hp Iphone 7+ Dual 12 MegaPixel full HD camera specifications with a screen resolution of 1920 x 1080 pixels. A sample of fish image data was taken during April–August 2021. The data obtained were 300 images of very fresh fish and 150 images of Not Fresh Fish. A sample of image data for fresh *Lutjanus campechanus* can be seen in Fig. 2, while the sample data for *Lutjanus campechanus* that is not fresh can be seen in Fig. 3.

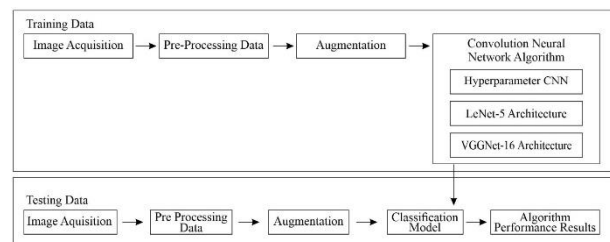


Figure 1. System design



Figure 2. Fresh fish of *Lutjanus campechanus*



Figure 3. Not fresh fish of *Lutjanus campechanus*



Figure 4. Eyes of fresh fish and not fresh fish

### 2.3. Pre-processing image

Pre-processing an image is a step to get input data of a *Lutjanus campechanus* image for the classification process by cutting the image. The process is done after getting some *Lutjanus campechanus* image data at the acquisition stage, then doing the cutting process to remove unnecessary objects, namely the fish's body. The research focuses on the red snapper eye object. The image-cutting process gives different image resolution results. Sample image data of a *Lutjanus campechanus* that has been pre-processed image can be seen in Fig. 4

### 2.4. Augmentation

Data augmentation is the process of reproducing an image without losing its essence [9]. Artificially augmentation is a technique to create new training data from existing training data. Data augmentation aims to expand the training data set to improve CNN performance and prevent over-fitting problems [10]. Augmentation is carried out only on not fresh fish data because the data obtained is less than fresh fish data, so the available data is not balanced. In the data augmentation process, traditional transformations are used, namely reflection and color transformation. These techniques are some of the most popular augmentation techniques because the method is easy to understand and has proven to be fast, reproducible, and reliable. Besides that, the implementation code is relatively easy and available for download with most deep learning frameworks [11]. Augmentation implementation is carried out using the Keras Library deep learning through the Image Data Generator class. Three techniques are used in this study, namely random brightness, one type of augmentation; Color transformation produces 50 new data types; horizontal flip and vertical flip, which is a type of reflection in the traditional transformation technique, each producing 50 new data. Image data non-fresh red snapper was 150 fish. After augmentation, the data obtained was 300 non-fresh red snapper image data, so the training data is 300 fresh *Lutjanus campechanus* and 300 Not fresh *Lutjanus campechanus*.

### 2.5. Augmentation architectures of Convolution Neural Network (CNN)

CNN is a supervised deep-learning tool. This algorithm is acceptable for multi-class classification and binary classification. CNN is often used to solve various pattern and image recognition problems. Deep learning approaches are effective and suitable for visuals [12]. The

CNN model is a combination of the following types: convolutional layers, pooling layers, fully connected layers, and fully connected layers that extract features from the input, minimize the size for computational performance and classify an image respectively [10]. This study uses CNN architecture. There are LeNet-5 and VGG-16.

The two architectures used are described below:

#### 1. Hyperparameter CNN

Hyperparameter is a variable that determines how a model is trained. In this experiment, the researcher also set the CNN hyperparameter, as presented in Table 1 We made adjustments to hyperparameters during the experiment as follows: The number of neurons in the fully connected layer is 1024, the dropout is 0.1, the optimizer is ADAM, the learning rate is 1e-5, the loss function is binary cross-entropy, the epoch is 100 times, and the batch size is set to 35.

#### 2. LeNet-5 Architecture

Neural Network Architecture was designed by Yann Lecun, Leon Bottou, Yosuha Bengio, and Patrick Haffner for handwriting and printing machine character recognition in 1998 called Lenet-5 [13]. LeNet-5 has eight layers, which are five convolution layers and three fully connected layers. Each unit has 25 inputs. The unit in the first hidden layer receives input from the 5×5 area. The input image is passed to the first hidden layer. This local area of the input image is called the unit receptive field. The unit's output is stored in the same location on the feature map. Various feature maps are generated from different weight vectors applied to the same input image. The features can be extracted from the obtained feature map. Sub-sampling has been described in the second layer. The number of map features obtained after sub-sampling is the same as that obtained after convolution. Here in the 2×2 sub-sampling layer, the area is taken as input and calculated as the average of the four inputs, multiplied by the trainable coefficient and adding trainable bias, giving it to the sigmoid function. An increase in the number of feature maps can be observed as the spatial resolution decreases layer by layer. Learning is carried out using the backpropagation method [14].

Table 2 shows the CNN LeNet-5 Architecture Network Layer used to implement fish freshness classification. There is a difference from the architecture in the output layer with a size of 2 classes because the output classification in this study only uses two classes, namely, Fresh Fish and Not fresh Fish.

Table 1. Hyperparameter CNN

No.	Hyperparameter	Value
1	Neurons in the hidden, fully connected layer	1024
2	Dropout	0.1
3	Optimizer	ADAM
4	Learning rate	1.00E-05
5	Loss function	Binary cross entropy
6	Epoch	100
7	Batch size	35
8	LR 2	0.0005

Table 2. Network layer architecture CNN LeNet-5

Layer	Layer Type	Feature Maps	Kernel Size	Size	Activation Function
Input	Image		-	32x32	
C1	Convolution	6	5x5	28x28	Tanh
S2	Sub Sampling	6	5x5	14x14	Sigmoid
C3	Convolution	16	5x5	10x10	Tanh
S4	Sub Sampling	16	5x5	5x5	Sigmoid
C5	Convolution	120	5x5	1x1	Tanh
F6	Fully Connected	-	5x5	84	Sigmoid
Output	Fully Connected	-	-	2	Softmax

Table 3. Network layer architecture CNN modified VGGNet-16

Layer	Layer Type	Feature maps	Kernel size	Size	Activation Function
Input	Image	3	-	50 x 50	-
	Convolution	64	3x3	50 x 50	Relu
	Convolution	64	3x3	50 x 50	Relu
	Max Pooling	64	2x2	25 x 25	Relu
	Convolution	128	3x3	25 x 25	Relu
	Convolution	128	3x3	25 x 25	Relu
	Max Pooling	128	2x2	12 x 12	Relu
	Convolution	256	3x3	12 x 12	Relu
	Convolution	256	3x3	12 x 12	Relu
	Convolution	256	3x3	12 x 12	Relu
	Max Pooling	256	2x2	6 x 6	Relu
	Convolution	512	3x3	6 x 6	Relu
	Convolution	512	3x3	6 x 6	Relu
	Convolution	512	3x3	6 x 6	Relu
	Max Pooling	512	2x2	3 x 3	Relu
	Convolution	512	3x3	3 x 3	Relu
	Convolution	512	3x3	3 x 3	Relu
	Max Pooling	512	2x2	1 x 1	Relu
	Fully Connected	-	-	4096	Relu
	Fully Connected	-	-	4096	Relu
	Fully Connected	-	-	1000	Relu
Output	Fully Connected	-	-	2	SoftMax

### 3. VGGNet-16 Architecture

Convolution input layer 1 using a standard image size of 224 x 224 RGB VGGNet-16 has 16 layers, namely 13 convolution layers and three fully connected layers. VGGNet-16 uses the block concept to form a convolution layer, each of which has a size of 3 x 3 and a stride layer of 1. At the end of the block, a max pooling layer of size 2 x 2 and stride 2 of 16 is used. The first convolution input layer is modified to 50 x 50 because of the large amount of processed data, so it requires a heavy training process.

The solution is to reduce the resolution of the input image in the training and testing process. The researcher modified it using the VGGNet-16 concept and produced a convolution neural network model with a modified VGGNet-16 architecture. Table 3 shows Network layer Arsitektur CNN Modified VGGNet-16.

### 4. Validation Holdout

The validation process is fundamental to do. The goal is that every piece of data can be used as training and experimental data. There are several model validations, one of which is Holdout validation [15]. Holdout validation is a dataset distribution where the data will be divided into testing data and training data. For example, if 0.2, then 20% of the data is used for testing and the rest for training data, which is 80%.

In this study, the holdout validation method is used, the simplest method that takes the original dataset and randomly divides it into two sets: the dataset into "training" and "testing" sets. The holdout method was applied to all trials conducted using deep learning (CNN), which used 80% of the 480 data for training and the remaining 20% of the 120 data for testing.

## 3. Results and Discussion

### 3.1. LeNet-5 architecture training performance

The performance of the model in the LeNet-5 architecture training process is based on hyperparameters and the Network Layer Architecture of CNN. The results showed that the highest training data accuracy reached 95.78% in the 100th epoch, while the lowest training data accuracy resulted in the 20th epoch was 87.77%. From 20 to 100 epoch, there is a rapid change in accuracy. The results of the LeNet-5 architectural training performance can be seen in Table 4. The graphics of Train-Tess Accuracy and Train-test Loss can be seen in Fig. 5.

### 3.2. Training performance of the VGGNet-16 architecture

The VGGNet-16 architecture training process model is based on the parameters and the Network Layer Architecture of CNN. The results showed that the highest training data accuracy reached 98.40% in the 100th epoch, while the lowest training data accuracy in the 20th epoch was 94.10%. From epoch 20 to epoch 100, there is a rapid change in accuracy. Table 5 Show The results of the VGGNet architectural training performance, Figure 6 shows the train-tess accuracy and train-test loss.

Table 4. LeNet-5 architecture training performance

Epoch	Batch size	Learning Rate	Optimizer	Val_Loss	Val_Acc	Loss	Accuracy
20	35	0.0001	Adam	25.25%	89.17%	25.99%	87.77%
40	35	0.0001	Adam	27.76%	89.17%	17.70%	94.73%
60	35	0.0001	Adam	17.05%	95.83%	21.84%	90.80%
80	35	0.0001	Adam	15.89%	94.17%	14.57%	94.95%
100	35	0.0001	Adam	16.50%	95.83%	14.44%	95.78%



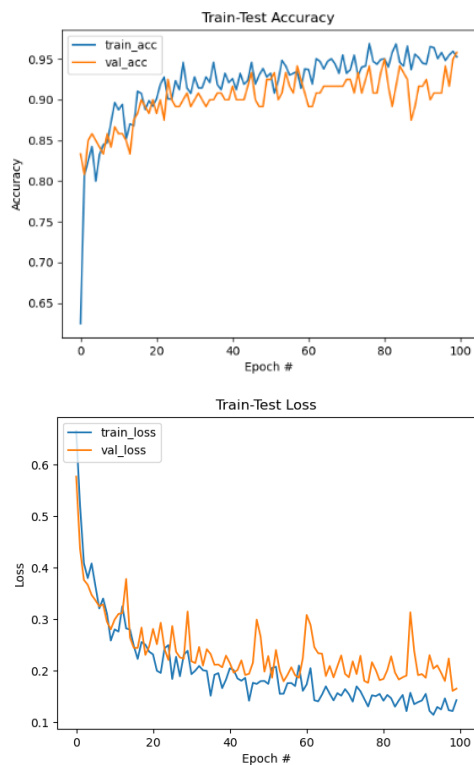


Figure 5. Graphics train-test accuracy and loss LeNet-5 Epoch 100

Table 5. VGGNet-5 architecture training

Epoch	Batch size	Learning Rate	Optimizer	Val_Loss	Val_Acc	Loss	Accur acy
20	35	0.0001	Adam	308.70%	80.00%	207.90%	94.10%
40	35	0.0001	Adam	234.75%	94.17%	206.18%	95.16%
60	35	0.0001	Adam	241.30%	92.50%	212.00%	95.70%
80	35	0.0001	Adam	202.01%	97.50%	196.43%	98.24%
100	35	0.0001	Adam	226.00%	97.00%	193.00%	98.40%

### 3.3. Testing performance of the LeNet-5 architecture

After training using LeNet-5 and VGGNet-16, the best model was obtained based on predetermined hyperparameters. The modeling was tested on new data

(that had not been previously trained) to determine the model's performance. The amount of tested data is 40 fish images, consisting of 20 images of fresh fish and 20 photos of not fresh fish. The test is to see how much accuracy is obtained from the model generated using LeNet-5 and VGGNet-16. The results of the LeNet-5 test of Lutjanus campechanus with a fresh label were 20, with the detection results of 14 fresh Lutjanus campechanus and six non-fresh Lutjanus campechanus. The LeNet-5 test of Lutjanus campechanus with the label not fresh amounted to 20, with the detection of not fresh 13 and 7 fresh Lutjanus campechanus. The following table shows the LeNet-5 red snapper test results, fresh and not fresh.

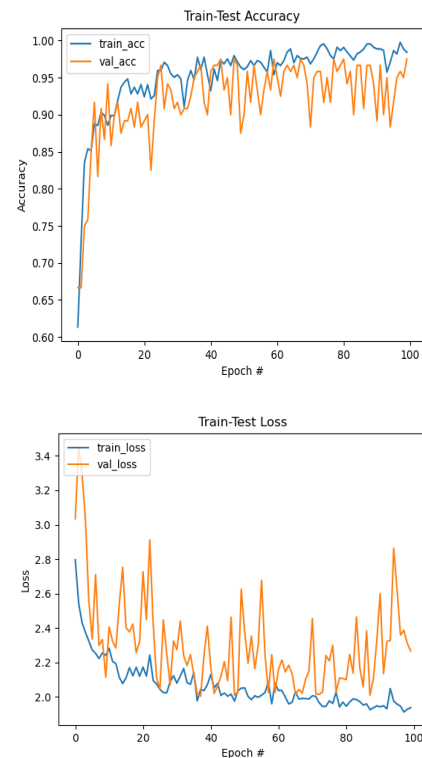


Figure 6. Train-test accuracy dan loss VGGNet-16 Epoch 100

Table 6. Testing result performance of the LeNet-5 architecture

Result testing fresh Lutjanus campechanus.				Result testing Not fresh Lutjanus campechanus.			
Image Fish	Class	classify	Probability	Image Fish	Class	classify	Probability
1 jpg	Fresh	Fresh	99.94%	21 jpg	Not Fresh	Not Fresh	100%
2 jpg	Fresh	Fresh	96.01%	22 jpg	Not Fresh	Not Fresh	100%
3 jpg	Fresh	Fresh	99.98%	23 jpg	Not Fresh	Not Fresh	65.32%
4 jpg	Fresh	Fresh	99.99%	24 jpg	Not Fresh	Not Fresh	99.95%
5 jpg	Fresh	Fresh	99.92%	25 jpg	Not Fresh	Not Fresh	99.95%
6 jpg	Fresh	Fresh	96.45%	26 jpg	Not Fresh	Not Fresh	99.93%
7 jpg	Fresh	Fresh	95.04%	27 jpg	Not Fresh	Fresh	98.55%
8 jpg	Fresh	Fresh	98.68%	28 jpg	Not Fresh	Fresh	100%
9 jpg	Fresh	Fresh	99.58%	29 jpg	Not Fresh	Fresh	99.89%
10 jpg	Fresh	Not Fresh	94.40%	30 jpg	Not Fresh	Fresh	100%
11 jpg	Fresh	Not Fresh	99.58%	31 jpg	Not Fresh	Fresh	99.58%
12 jpg	Fresh	Not Fresh	98.68%	32 jpg	Not Fresh	Not Fresh	99.70%
13 jpg	Fresh	Not Fresh	96.62%	33 jpg	Not Fresh	Not Fresh	100%
14 jpg	Fresh	Not Fresh	99.99%	34 jpg	Not Fresh	Not Fresh	99.76%
15 jpg	Fresh	Not Fresh	99.22%	35 jpg	Not Fresh	Not Fresh	97.21%
16 jpg	Fresh	Fresh	98.62%	36 jpg	Not Fresh	Not Fresh	55.78%
17 jpg	Fresh	Fresh	89.31%	37 jpg	Not Fresh	Not Fresh	100%
18 jpg	Fresh	Fresh	98.68%	38 jpg	Not Fresh	Not Fresh	99.64%
19 jpg	Fresh	Fresh	99.98%	39 jpg	Not Fresh	Fresh	100%
20 jpg	Fresh	Fresh	99.99%	40 jpg	Not Fresh	Fresh	99.98%

Table 7. Testing result performance of the VGGNet-16 architecture

Result testing Not fresh Lutjanus campechanus				Result testing Not fresh Lutjanus campechanus			
Image Fish	Class	classify	Probability	Image Fish	Class	classify	Probability
1 Jpg	Fresh	Fresh	100%	21 jpg	Not Fresh	Fresh	97.19%
2 Jpg	Fresh	Fresh	85.26%	22 jpg	Not Fresh	Fresh	99.84%
3 Jpg	Fresh	Fresh	100%	23 jpg	Not Fresh	Not Fresh	93.90%
4 jpg	Fresh	Fresh	100%	24 jpg	Not Fresh	Not Fresh	99.24%
5 jpg	Fresh	Fresh	100%	25 jpg	Not Fresh	Not Fresh	100%
6 jpg	Fresh	Fresh	100%	26 jpg	Not Fresh	Not Fresh	100%
7 jpg	Fresh	Fresh	100%	27 jpg	Not Fresh	Not Fresh	100%
8 jpg	Fresh	Fresh	99.99%	28 jpg	Not Fresh	Not Fresh	100%
9 jpg	Fresh	Fresh	100%	29 jpg	Not Fresh	Not Fresh	100%
10 jpg	Fresh	Fresh	99.63%	30 jpg	Not Fresh	Fresh	100%
11 jpg	Fresh	Fresh	100%	31 jpg	Not Fresh	Fresh	100%
12 jpg	Fresh	Fresh	99.87%	32 jpg	Not Fresh	Fresh	100%
13 jpg	Fresh	Fresh	98.81%	33 jpg	Not Fresh	Not Fresh	100%
14 jpg	Fresh	Fresh	100%	34 jpg	Not Fresh	Not Fresh	99.99%
15 jpg	Fresh	Fresh	100%	35 jpg	Not Fresh	Not Fresh	99.22%
16 jpg	Fresh	Not Fresh	100%	36 jpg	Not Fresh	Not Fresh	99.93%
17 jpg	Fresh	Not Fresh	85.26%	37 jpg	Not Fresh	Not Fresh	99.65%
18 jpg	Fresh	Not Fresh	100%	38 jpg	Not Fresh	Not Fresh	99.84%
19 jpg	Fresh	Not Fresh	91.55%	39 jpg	Not Fresh	Not Fresh	99.90%
20 jpg	Fresh	Not Fresh	100%	40 jpg	Not Fresh	Not Fresh	100%

3.4. Testing performance of the VGGNet-16 architecture

After conducting training using LeNet-5 and VGGNet-16, the best model was obtained based on predetermined parameters, and it was tested on new data to determine the model's performance. The amount of data is 40 fish images, consisting of 20 images of fresh fish and 20 images of non-fresh fish, to see how much accuracy is obtained from the model generated using LeNet-5 and VGGNet-16. The results of the VGGNet-16 test of red snapper with a fresh label were 20, with the detection results of 15 fresh Lutjanus campechanus and five not fresh Lutjanus campechanus. The LeNet-5 test of Lutjanus campechanus with the label not fresh amounted to 20, with the detection of not being fresh amounted to 15 red snappers and five fresh Lutjanus campechanus. The following table shows the results of the VGGNet-16 red snapper testing, fresh and not fresh.

3.5. LeNet-5 and VGGNet-16 model testing results using new data

The following explains the classification performance used in this study by finding the value of performance measurement:

$$Accuracy = \frac{TP+TN}{TP+TN+FN+FP} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$F1 - measure = 2 \frac{Presisi*Recall}{Presisi+Recall} \tag{4}$$

Table 8 shows the prediction results of LeNet-5 and VGGNet-16. The results of the modeling test, the accuracy value obtained is the highest accuracy value of 75% Using the VGGNet-16 model. Figure 7 shows the new data image of testing data. The 40 images have not been used for training.

Table 8. Testing results using new data

Model	Accuracy	Precision	Recall	F1-Score
LeNet-5	73.00%	70.00%	66.70%	68.30%
VGGNet-16	75.00%	75.00%	75.00%	75.00%



Figure 7. Image data testing

4. Conclusion

Based on the analysis of the results of the identification of the freshness level of Lutjanus campechanus using the CNN LeNet-5 and VGGNet-16 methods, researchers can conclude from the research results:

The procedure for building this classification system involves several process stages, starting from Image acquisition, cutting, and data augmentation. The first stages are carried out to obtain input data for the classification system, and then the classification process is training and testing. The results of the training comparison of the 2 LeNet-5 and VGGNet-16 architectures with the highest accuracy value on LeNet-5 reached 95.78% using epoch 100, batch size 35, learning rate 0.0001, and the VGGNet-16 architecture with an accuracy value of 98.40% using epoch 100, batch size 35, learning rate 0.0001 So, the highest accuracy value is obtained on the VGGNet-16 architecture. Test results comparison of 2

LeNet-5 and VGGNet-16 architectures with the highest accuracy value on LeNet-5 reaching 73.0%, Precision 70.0%, Recall 66.7%, F1-Score 68.3%, and the highest VGGNet-16 architecture reaching 75.0 %, Precision 75.0%, Recall 75.0%, F1-Score 75.0%. It shows that VGGNet-16 is more appropriate to be implemented in classifying fresh and unfresh red snapper freshness.

### Acknowledgements

LPPPM Politeknik Negeri Fakfak funded this research. The researchers really appreciate this help. In addition, we would like to convey our gratitude to all of our direct and indirect supporters.

### References

- [1] Kementerian Kelautan dan Perikanan (KKP), "Pencarian Produksi Ikan Dengan Perbandingan Tahun," 2020.
- [2] N. Sengar, V. Gupta, M. K. Dutta, and C. M. Travieso, "Image Processing Based Method For Identification Of Fish Freshness Using Skin Tissue," in *4th International Conference on Computational Intelligence & Communication Technology (CICT)*, 2018, pp. 1–4, doi: 10.1109/CICT.2018.8480265.
- [3] M. Arora, P. Mangipudi, M. K. Dutta, and R. Burget, "Image Processing Based Automatic Identification of Freshness in Fish Gill Tissues," doi: 10.1109/ICACCCN.2018.8748778.
- [4] N. M. S. Iswari, Wella, and Ranny, "Fish freshness classification method based on fish image using k-Nearest Neighbor," in *2017 4th International Conference on New Media Studies (CONMEDIA)*, 2017, pp. 87–91, doi: 10.1109/CONMEDIA.2017.8266036.
- [5] K. A. Priya, A. C. Kaladevi, and R. Perumal, "Detection of Sardine Fish Freshness Using Deep Convolution Neural Network," *Ann. Rom. Soc. Cell Biol.*, vol. 25, no. 4, pp. 16063–16070, 2021.
- [6] E. Prasetyo, R. Purbaningtyas, R. D. Adityo, E. T. Prabowo, and A. I. Ferdiansyah, "A Comparison Of Convolution Neural Network For Classifying Milkfish's Freshness On Eye Images," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, pp. 601–608, 2021, doi: 10.25126/jtiik.202184369.
- [7] A. Nasiri, A. Banan, Y.-D. Zhang, and A. T. Garavand, "Smart deep learning-based approach for non-destructive freshness," *J. Food Eng.*, vol. 278, no. 3, 2020.
- [8] M. A. Rayan, A. Rahim, M. A. Rahman, M. A. Marjan, and U. A. M. E. Ali, "Fish Freshness Classification Using Combined Deep Learning Model," in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, 2021, pp. 1–5, doi: 10.1109/ACMI53878.2021.9528138.
- [9] T. Badriyah, N. Sakinah, I. Syarif, and D. R. Syarif, "Machine Learning Algorithm for Stroke Disease Classification," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, 2020, pp. 1–5, doi: 10.1109/ICECCE49384.2020.9179307.
- [10] M. M. A. Monshi, J. Poon, V. Chung, and F. M. Monshi, "CovidXrayNet: Optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR," *Comput. Biol. Med.*, vol. 133, no. 104375, 2021.
- [11] Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in an image classification problem," in *2018 International Interdisciplinary PhD Workshop (IIPHDW)*, 2018, pp. 117–122, doi: 10.1109/IIPHDW.2018.8388338.
- [12] D. M. Abdullah, A. M. Abdulazeez, and A. B. Sallow, "Lung cancer Prediction and Classification based on Correlation Selection method Using Machine Learning Techniques," *Qubahan Acad. J.*, vol. 1, no. 2, pp. 141–149, 2021, [Online]. Available: 10.48161/qaj.v1n2a58.
- [13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE 86 (11)*, 1998, pp. 2278–2324.
- [14] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," *Procedia Comput. Sci.*, vol. 132, pp. 679–688, 2018.
- [15] F. Tempola, R. Rosihan, and R. Adawiyah, "Holdout Validation for Comparison Classification Naïve Bayes and KNN of Recipient Kartu Indonesia Pintar," in *IOP Conference Series: Materials Science and Engineering*, 2021, vol. 1125, doi: 10.1088/1757-899X/1125/1/012041.