

Determining Tuna Grade Quality Based on Color Using Convolutional Neural Network and k-Nearest Neighbors

I Gede Sujana Eka Putra^{a1}, Ahmad Catur Widyatmoko^{b1}, I K G Darma Putra^{c1}, Made Sudarma^{c2}, Oka Sudana^{c3}

^aDepartment of Computer Informatics, Faculty of Informatics, Institute of Business and Technology Indonesia

^bInstitute for Marine and Antarctic Studies, University of Tasmania, Hobart, 7000, Tasmania, Australia

^cDepartment of Information Technology, Faculty of Engineering, Udayana University, Indonesia

¹sujanaekaputra@instiki.ac.id, ²ahmadcatur.widyatmoko@utas.edu.au

³ikgdarmaputra@unud.ac.id, ⁴sudarmaee@unud.ac.id, ⁵agungkakas@unud.ac.id

Abstract

One of the main commodities that Indonesia exports is tuna. Indonesia's inadequate handling of food safety is demonstrated by a number of instances when the United States has rejected Indonesian fishery goods and food poisoning incidences. Fish quality grade is currently determined by manual inspection, which has a risk of human mistakes. According to Robert DiGregorio, four tuna grade classifications exist: grade 1, 2+, 2, and 3. The purpose of this study is to assess the quality of tuna meat according to its color. The procedure involves pre-processing images, training datasets, and classifying them using the Convolutional Neural Network (CNN) and k-Nearest Neighbors algorithms. CNN pre-processing involves converting the image into HSV color space and training the CNN model using 240 training datasets and 74 testing datasets. CNN's accuracy was 84% higher than k-Nearest Neighbors', which was 54%. Additionally, a comparison of the classification accuracy of CNN, VGG (Visual Geometry Group) 16, and AlexNet revealed that CNN outperformed the others with an accuracy of 84%, followed by VGG16 with 70% and AlexNet with 66%.

Keywords: Grade, Pre-Processing, Convolutional Neural Network, k-Nearest Neighbors, Classification

1. Introduction

One of the most popular exports from the US, Europe, Japan, and Vietnam is tuna goods[1]. Among the 32 nations that make up the Indian Ocean Tuna Commission (IOTC), The world's largest tuna producer is Indonesia, contributing more than sixteen percent of global production, and skipjack[2][3]. To guarantee product quality and food safety for the global market, it is critical to identify fish quality[4].

The United States' repeated rejections of Indonesian fisheries exports are proof of the country's inadequate food security management.[4]. The United States has been fighting illegal, unrecorded, and unregulated (IUU) fishing in order to defend the country's economy, global food security, and fisheries sustainability since 2019 by implementing and monitoring scheme for seafood imports to ensure traceability of farm products[5][6].

The manual tuna grade determination technique is susceptible to human error[7]. According to Robert DiGregorio's book Tuna Grading and Evaluation[8] states there are 5 characteristics to determine tuna grade, including freshness, size and shape of fish, meat color, texture, and fat content. Four categories are used to classify tuna meat: grade 1, 2+, 2, and 3[8]. Previous studies from [9] had been carried out to determine the grade quality of tuna meat based on color space using k-NN algorithm which their pre-processing using the Symlet wavelet and Haar wavelet. Using 95 training datasets to classify 65 test datasets, it is obtained that k-NN classification using Symlet as feature extraction has a better accuracy of 81.8% compared to using Haar with an

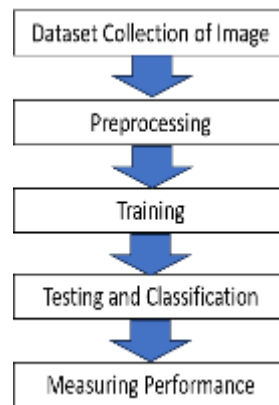


Figure 1. Classification Step

accuracy of 80.3%. This research is an extension of a previous study to compare how accurate the k-NN and CNN algorithms are at predicting the grade of tuna according to its color. This study also discusses comparing the performance of customized CNN, CNN using VGG16, and CNN using the AlexNet architecture model.

2. Research Methods

Classifying tuna meat involves a number of procedures, including collecting datasets, pre-processing images, and training and classifying datasets. This study uses a Convolutional Neural Network and k-Nearest Neighbors to compare two methods for classifying tuna's grade quality. We compare the performance of other CNN types, such as VGG16, AlexNet, and customized CNN, in the discussion section. As shown in Figure 1, the study method entails collecting image datasets, pre-processing images, training, testing, and classification, as well as performance evaluation. Image pre-processing by converting an image from the RGB to the HSV color space is a common practice due to better color-based segmentation, making it easier to isolate specific colors for segmentation, to illuminate invariance, which is the value (V) component that represents brightness, and then to work with Hue and Saturation while ignoring Value, making the processing less sensitive to shadows, lighting changes, and brightness variations. HSV is more aligned with how humans perceive and describe colors, and thresholding in HSV is often more straightforward.

2.1. Image Acquisition, Image Preprocessing

The CAMTECH CT50 webcam and C#-based programs are used for image acquisition[10][11][12], afterward, Python is used for pre-processing[13][14]. After dividing the dataset into four categories: Grades 1, 2, 2+, and 3, the datasets were pre-processed and center-cropped, as illustrated in Figures 2, 3, 4, and 5.

2.2 Convolutional Neural Network Algorithms

A convolutional neural network comprises three layers: input, hidden, and output[15]. The input layer entails taking characteristics out of the tested data, and the hidden layer is the learning coefficient that is created by learning the input data until the error rate is as low as possible. The output layer, on the other hand, is the outcome of the input data following convolution in the hidden layer. The layer that processes lowering the size of image data is called the pooling layer to increase the positional invariance of features, as well as speed up computation and control overfitting[16]. The one activation uses Rectified Linear Unit known as ReLU to implement non-linearity into neural networks[17], via the $\max(0, x)$ function. The Rectified Linear Unit (ReLU) activation function is applied after the convolution operation and before the pooling operation.

$$y = \begin{cases} x_i, & \text{if } x_i > 0 \\ 0, & \text{if } x_i < 0 \end{cases} \quad (1)$$

If the input $x > 0$ then ReLU returns x .
 If the input $x \leq 0$ then ReLU returns 0.



Figure 2. Pictures of Grade 1 Tuna



Figure 3. Pictures of Grade 2+ Tuna



Figure 4. Pictures of Grade 2 Tuna



Figure 5. Pictures of Grade 3 Tuna

ReLU return result straight line (slope = 1) for positive inputs and Flat at zero for negative inputs.

2.3 Confusion Matrix

Confusion Matrix is one of the methods used to evaluate classification methods[18]. The Confusion Matrix is shown in Table 1.

Table 1. Confusion Matrix

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

Data that are appropriately categorized as negative or false output are known as True Negative (TN) values. Data that is accurately categorized as a true or positive result is known as a True Positive (TP). False Positive (FP) data is data that is misclassified, even though the result is true or positive. Inaccurately classified data are known as False Negatives (FN)[18]. Precision is an evaluation metric of how often the model correctly predicts the positive class, among all positive predictions made by the model. Recall measures the proportion of actual positive cases that were correctly identified by a model, which describes how good a model is at correctly identifying the positive class. F1 Score gives an idea of how well our model is at accurately classifying both positive and negative class. Accuracy is an evaluation metric that measures how well a model makes correct predictions out of the total predictions it makes. In the context of classification, accuracy provides an idea of how often a model predicts the correct class, whether it is positive or negative.

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

while True Positives (TP): correctly predicted positive outcomes.

- $$\text{Recall} = \frac{TP}{TP+TN}$$
 (3)

while true negative (TN) represents the accurate prediction of a negative outcome.

- $$\text{F1-Score} = 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$
 (4)

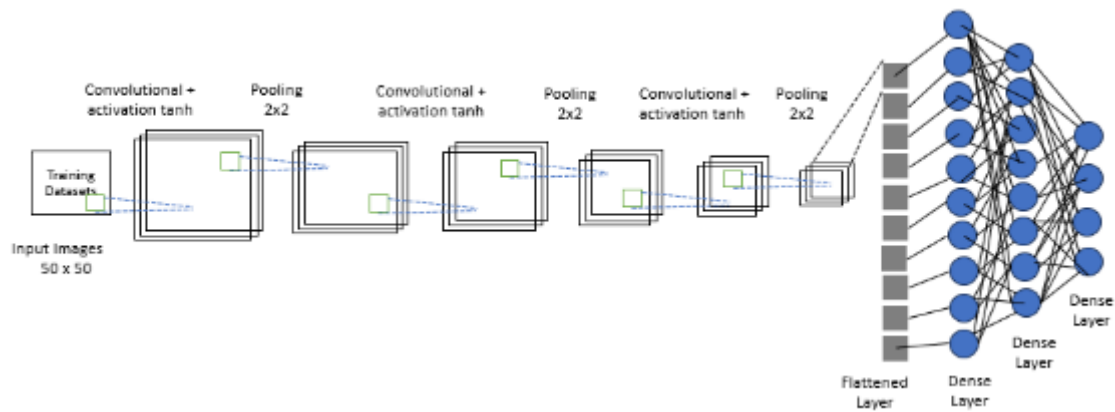


Figure 6. CNN Architecture Model[15]

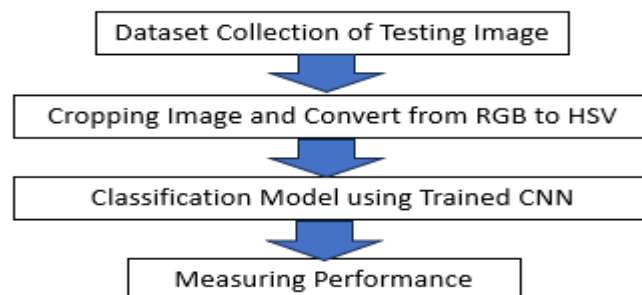


Figure 7. The Classification Step Using Convolutional Neural Network

- $$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (5)$$
 while False Positive (FP) occurs when the model incorrectly predicts a positive result, but the actual label is negative. A False Negative (FN) occurs when the model predicts a negative result, but the actual label is positive.

2.4 Training using Convolutional Neural Network

The training phase comes after the image acquisition and pre-processing stages. Training models using Convolutional Neural Network model as shown in Figure 6.

This study uses training datasets 240 images which is displayed in Table 2.

Table 2. Composition of Training Datasets of Images

No	Grade	Number of Training Datasets
1	1	60
2	2+	60
3	2	60
4	3	60

The training datasets using this Convolutional Neural Network model as seen in Figure 9 using 300 epochs and batch size = 10.

2.5 Classification using Convolutional Neural Network

The flow of the classification step using a Convolutional Neural Network includes collecting datasets for testing, as shown in Figure 7. The classification process starts with collecting test

dataset images, then cropping them to 50 x 50 pixels to eliminate unanticipated surrounding images, converting each test image from RGB to HSV, normalizing the images to obtain dataset features, and classifying the images using the CNN model. Lastly, the Confusion Matrix is used to evaluate each grade quality's performance. The composition of testing datasets is seen in Table 3.

Table 3. Composition of Testing Datasets of Images

No	Grade	Number of Testing Datasets
1	1	16
2	2+	20
3	2	15
4	3	23

2.6 Wavelet Transform

The feature selection process reduces dimensionality by eliminating features that are irrelevant, while the wavelet algorithm is used to extract features[19]. In terms of image noise removal, the wavelet symlet is superior to other filters and has a decent level of dependability[20]. The wavelet will break down into four distinct images with varying frequencies that are filtered by column and row. Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH) are the four types of frequencies that are produced[19]. The result of this procedure will reveal that the image has been split into four sections, each of which is $\frac{1}{4}$ the dimensions of the original[21]. An illustration of the outcomes of the wavelet decomposition procedure is provided in Figure 8.



Figure 8. Wavelet Result After Decomposition

2.7 k-Nearest Neighbors Algorithm-Based Classification

New instances are classified using the k-Nearest Neighbor (k-NN) technique according to how similar they are to most of the items in a class[22][23]. The Euclidean distance is used to determine how close two instances are, as seen in equation (6).

$$\delta_{xy} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (6)$$

δ_{xy} = Distance in Euclidean terms
 x_i = reference or instance data
 y_i = fresh test instance
 n = number of property or feature values

The Euclidean distance is typically used during the prediction phase (both for classification and regression), when a new input (test point) needs to be classified or used for regression. The algorithm computes the Euclidean distance between this new input and each sample in the training dataset. The parameter k is defined using the k-NN technique. In the k-NN algorithm, the k value specifies how many neighbors will be looked at to categorize a specific query point. The parameter k (the number of nearest neighbors to consider) plays a crucial role in how well the algorithm performs on a specific dataset. The value of k needs to be suitable for the data to balance bias and variance. The next step is to determine each image feature's Euclidean distance to its neighboring image features (training dataset). One class is created from the features of a picture that are close to each other. The next step is to sort the image's characteristics into class

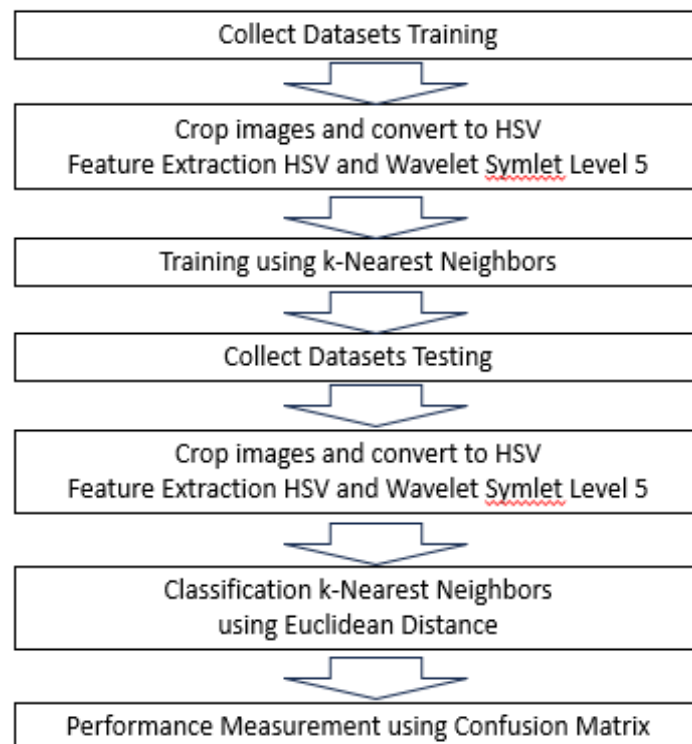


Figure 9. The Training and Classification Step Using k- Nearest Neighbors

groups based on their least Euclidean distance. If the object's distance is the smallest, it is in the nearest neighbouring class[24]. The flow of the classification step using k-Nearest Neighbors includes collecting datasets for testing, as shown in Figure 9.

2.8 . Feature Extraction and Classification Using k-Nearest Neighbors

Wavelet Symlet level 5 is used for feature extraction. To create an HSV feature, the H, S, and V channel feature extraction results are then concatenated. Using k-Nearest Neighbors with $k=4$, the HSV feature is to be learned. Gathering training data, cropping the image to 50 x 50 pixels, making the conversion from RGB to HSV, and then extracting characteristics for every H, S, and V channel utilizing level 5 Wavelet Symlet, are the procedures involved in training and classifying using k-Nearest Neighbors. To create the HSV feature, each characteristic was then blended with the others. There are then four classes accessible for this functionality. The test image is assigned to the class with the closest distance.

3 Results and Discussion

3.1 Results

The study's outcome compares two algorithms, k-Nearest Neighbors and Convolutional Neural Network, in order to determine which, one has more accuracy using a confusion matrix. Once the datasets were trained using a Convolutional Neural Network, we used dataset testing to predict the grade quality. The confusion matrix diagram in Figure 10 provides an evaluation of this approach processed with Python. We discovered that the algorithm correctly predicts 13 times for Grade 1, 14 times for Grade 2+, 12 times for Grade 2, and 23 times for Grade 3 based on the confusion matrix displayed in Figure 10.

Table 4 shows the classification performance of the Convolutional Neural Network, which was generated using Python. The precision of Grade 01 referring to equation (2), the model 100% correctly predicted all instances of Grade 01. Refer to equation (3). Grade 01 calculates recall means the model can avoid missing real positives (false negatives) 81%. Referring to equation (4), the result of the F1 score for Grade 01 is 90%. The model accurately classifies positive and

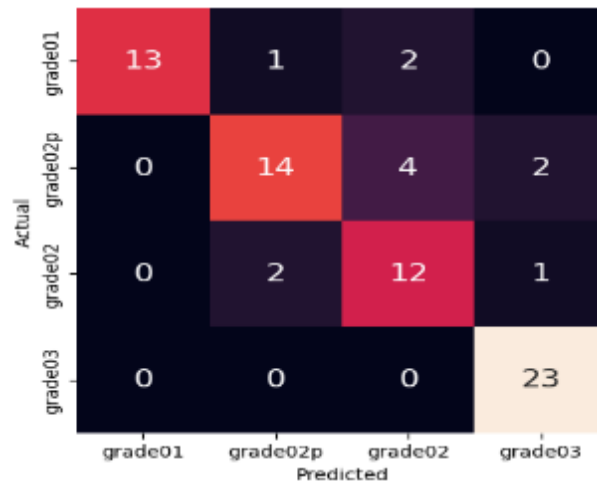


Figure 10. Confusion Matrix of Convolutional Neural Network

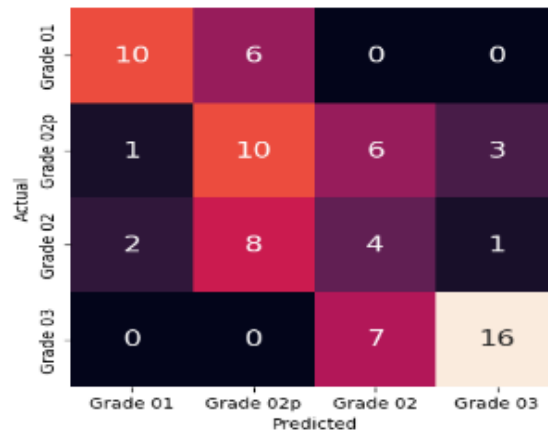


Figure 11. Confusion Matrix of k-Nearest Neighbors

negative classes at 90%. Refer to equation (5); the result of accuracy for Grade 01 means the model makes correct predictions out of the total predictions it makes around 84%. For dataset training, another approach employs k-Nearest Neighbors with k=4. Following a confusion matrix diagram to evaluate this technique, as seen in Figure 11.

Table 4. Classification Report of Convolutional Neural Network

No	Grade	Precision	Recall	F1-Score	Accuracy
1	1	100%	81%	90%	
2	2+	82%	70%	76%	
3	2	67%	80%	73%	
4	3	88%	100%	94%	
5	All				84%

The algorithm has corrected the prediction ten times for Grade 1, ten times for Grade 2+, four times for Grade 2, and sixteen times for Grade 3 based on k-Nearest Neighbors, as shown in Figure 11.

Table 5 shows the classification performance of k-Nearest Neighbors, which was generated using Python. Referring to equation (2), regarding the precision of Grade01, the model 77% correctly predicted all instances of Grade01. Regarding recall, Grade 01 calculates recall referring to equation (3); the model can avoid missing real positives (false negatives) 62% of the time. Referring to equation (4), the result of the F1 score for Grade 01 means the model accurately

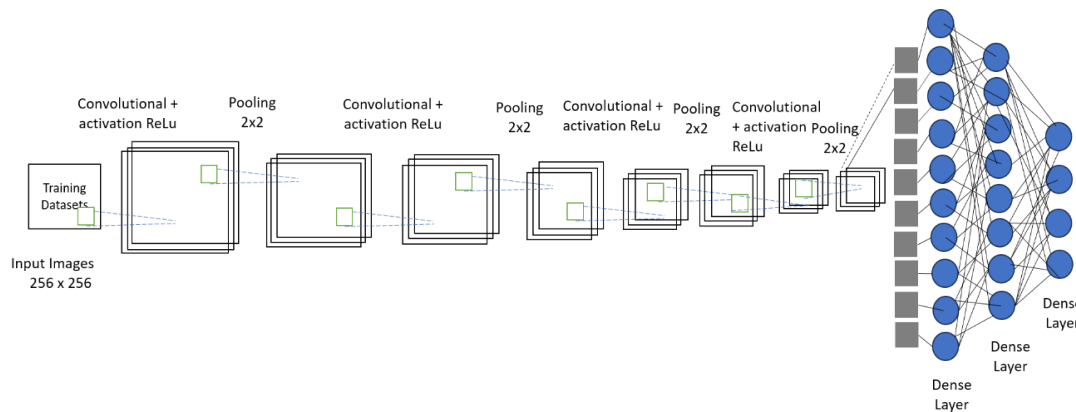


Figure 14. Model of Convolutional Neural Network AlexNet[26]

Table 6. Classification Report of CNN VGG16

No	Grade	Precision	Recall	F1-Score	Accuracy
1	1	93%	88%	90%	
2	2+	50%	65%	57%	
3	2	56%	33%	42%	
4	3	83%	87%	85%	
5	All				70%

grade01 grade02p grade02 grade03
 Predicted

Figure 13. Confusion Matrix of CNN VGG16

classifies positive and negative classes at 69%. Refer to equation (5); the result of accuracy means the model makes correct predictions out of the total predictions it makes, around 54%.

Table 5. Classification Report of k-Nearest Neighbors

No	Grade	Precision	Recall	F1-Score	Accuracy
1	1	77%	62%	69%	
2	2+	42%	50%	45%	
3	2	24%	27%	25%	
4	3	80%	70%	74%	
5	All				54%

3.2 Discussion

3.2.1. Classification using models of Convolutional Neural Network VGG16

In this study, we have trained 240 training datasets using Convolutional Neural Network VGG16 models with dimensions of 150x150 pixels according to Figure 12. Figure 12 displays the VGG16 model of a CNN, which consists of two fully connected/dense layers, four convolutional layers, and four pooling layers. The research wants to examine using input images 150 x 150 to investigate the accuracy of the models compared with the models that use images 50 x 50.

For Grades 1, 2+, 2, and 3, respectively, the algorithm has corrected predictions 14 times, 13 times, 5 times, and 20 times, respectively, according to the CNN VGG16 confusion matrix shown in Figure 13.

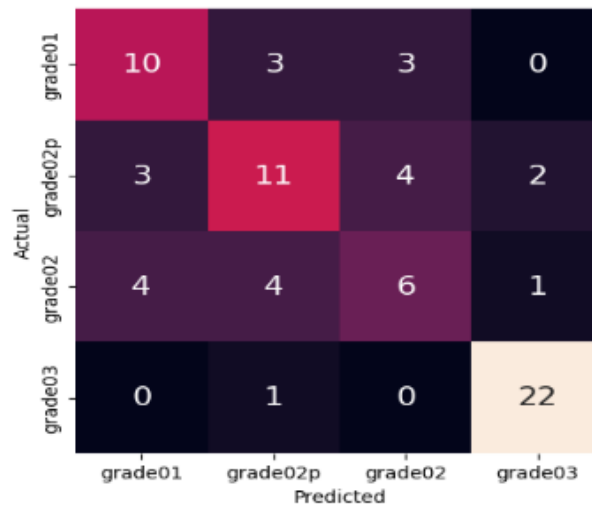


Figure 15. CNN AlexNet's Confusion Matrix

Table 7. Classification Report of CNN AlexNet

No	Grade	Precision	Recall	F1-Score	Accuracy
1	1	59%	62%	61%	
2	2+	58%	55%	56%	
3	2	46%	40%	43%	
4	3	88%	96%	92%	
5	All				66%

CNN VGG16's classification performance, produced by Python, is displayed in Table 6. Refer to equation (2); regarding the precision of Grade01, the model 93% correctly predicted all instances of Grade01. Regarding recall, Grade 01 calculates recall referring to equation (3); the model can avoid missing real positives (false negatives) 88% of the time. Refer to equation (4); the result of the F1 score for Grade 01 is 0.9. This means the model is accurately classifying positive and negative classes at 90%. Referring to equation (5), the accuracy result is 0.70. The model makes correct predictions out of the total predictions it makes around 70%. Grade 1 has the highest precision percentage (93%), followed by Grade 3 (83%), Grade 2, (56%), and Grade 2+ (50%) in that order.

3.2.2. Classification using CNN AlexNet

We have used CNN AlexNet models with eight layers and 256x256 pixel sizes to learn 240 training datasets. The training phase comes after image acquisition. Convolutional neural network models are used for model training, as illustrated in Figure 14.

The CNN's AlexNet model consists of four convolutional layers, including a max pooling layer and three fully connected/dense layers respectively, which is shown in Figure 14. The research wants to examine using input images 250 x 250 to find out impact to the accuracy of the models compared with the models which using images 50 x 50. After classifying and grading 74 test datasets, we discovered the confusion matrix which is displayed in Figure 15.

According to CNN AlexNet, as shown in Figure 15, the algorithm has been able to correct predictions ten times for Grade 1, eleven times for Grade 2+, six times for Grade 2, and twenty-two times for Grade 3.

According to CNN AlexNet's classification report utilizing a dataset of 74 images, as shown in Table 7, Grade 1 has the highest precision rate (93%), followed by Grade 3 (83%), Grade 2, and Grade 2+ (56%). Grade 2+ has the lowest precision rate (50%).

3.2.3. Classification using Customized CNN Algorithm

The CNN's classification report is displayed in Table 8. Refer to equation (2), regarding of the precision of Grade01, the model 100% correctly predicted of all instance of Grade01. Regarding to recall, Grade 01 calculate recall refer to equation (3), the model can avoid missing real positives (false negative) is 81%. Refer to equation (4), the result of F1 score for Grade 01 means the model is accurately classifying positive and negative class is at 90%. Refer to equation (5), the result of accuracy means the model makes correct predictions out of the total prediction it makes is around 84%. The model indicates that Grade 1 has the highest percentage of precisions (100%) followed by Grade 3 (88%), Grade 2+ (82%), and Grade 2 (67%). There are 74 image datasets in all used for testing in this work.

Table 8. Classification Report of Customized CNN

No	Grade	\Precision	Recall	F1-Score	Accuracy
1	1	100%	81%	90%	
2	2+	82%	70%	76%	
3	2	67%	80%	73%	
4	3	88%	100%	94%	
5	All				84%

3.2.4. Comparison Classification Report Between Customized CNN, VGG16 and AlexNet

Based on classification report between Customized CNN, CGG16 CNN and AlexNet CNN, could be described in Table 9.

Table 9. Comparison of Precision for 3 CNN Model

	VGG16	AlexNet	Customized CNN
Grade 01	93%	59%	100%
Grade 2P	50%	28%	82%
Grade 02	56%	46%	67%
Grade 03	83%	88%	88%

One indicator that provides information about the quality of positive prediction is precision. A comparison of the accuracy values in the VGG16, AlexNet, and Convolutional Neural Network models is shown in Table 9 and Figure 16, where the Convolutional Neural Network model has the highest precision value, followed by the VGG16 model and the AlexNet model.

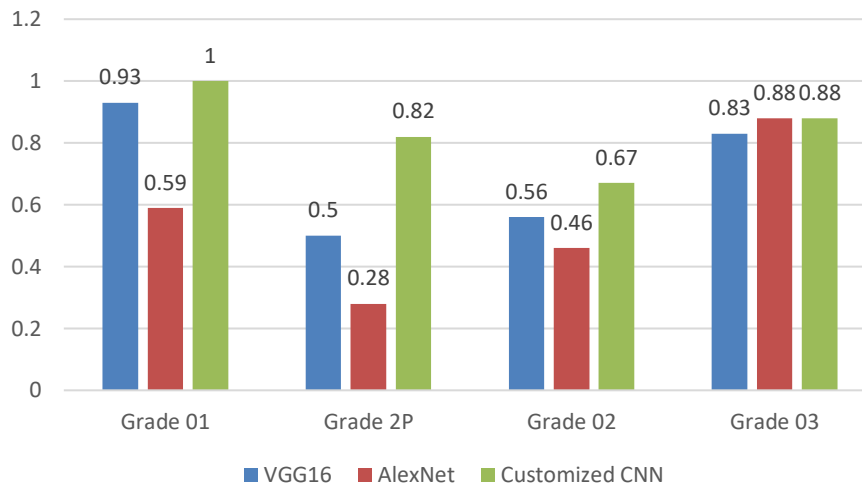


Figure 16. Percentage of Precision for Each Grade

The frequency with which a machine learning model accurately predicts the result is known as accuracy. Table 10 shows that the Convolutional Neural Network model has the best accuracy value, followed by the VGG16 model, while the AlexNet model has the lowest precision value. The custom CNN was likely better tailored to the dataset (e.g., image size = 50×50) and utilized fewer parameters, which caused less overfitting and better generalization.

Table 10. Accuracy for 3 CNN Model

	VGG16	AlexNet	Customized CNN
Accuracy	70%	66%	84%

While VGG16 might not have been fully optimized for the smaller input (150×150), it performs adequately but not best. However, AlexNet has low precision, suggesting many false positives due to overfitting or poor generalization. A mismatch between input image size (250×250) and model expectation, large architecture, and unsuitable preprocessing may slow down performance. VGG16 is a deeper architecture and often requires careful fine-tuning for each dataset. While it can perform well on complex datasets, its heavier architecture can make it prone to overfitting. If it does not receive proper training on enough data, it might overfit certain patterns. VGG16 may overfit certain patterns in the training set, reducing its generalization to new data. Slow convergence: due to its depth, VGG16 might require more epochs and better optimization to match or surpass the performance of simpler models like custom CNNs. AlexNet is an older architecture; it might struggle in certain modern tasks, simpler architecture. AlexNet has fewer layers and may not be able to capture as much detailed information as deeper models like VGG16. AlexNet might be struggling with overfitting or underfitting depending on the dataset size and complexity and limited regularization.

4. Conclusion

The study concludes that the proposed approach involves a series of steps, namely image acquisition, preprocessing, training, and image classification, to determine the quality grade of tuna meat. Using 240 training datasets and 74 testing datasets, the findings show that Convolutional Neural Networks (CNN) outperform k-Nearest Neighbours (k-NN), achieving an accuracy of 84% compared to k-NN's 54%. Furthermore, based on a comparison of CNN, VGG16, and AlexNet, CNN emerges as the most effective classification model, with an accuracy of 84%, surpassing VGG16 (70%) and AlexNet (66%).

References

- [1] T. Rudi Hartanto, S. Suharno, and B. Burhanuddin, "Daya Saing Ekspor Ikan Tuna-Cakalang-Tongkol Indonesia di Pasar Amerika Serikat," *J. Pengolah. Has. Perikan. Indones.*, vol. 24, no. 2, 2021, doi: 10.17844/jphpi.v24i2.36075.
- [2] Ediyanto, "Manajemen Pengelolaan Sumber Daya Ikan Tuna Indonesia," *PROSIDING*, vol. 1, no. 01, 2022, doi: 10.59134/prosidng.v1i01.82.
- [3] Z. Syam, V. Silvia, and T. C. Dawood, "Determinants of Indonesia Frozen Yellowfin Tuna Exports to Main Destination Countries," *Tech. Soc. Sci. J.*, vol. 21, 2021.
- [4] D. M. Power, P. Taoukis, D. Houhoula, T. Tsironi, and E. Flemetakis, "Integrating Omics Technologies for Improved Quality and Safety of Seafood Products," *Aquac. Fish.*, vol. 8, no. 4, 2023, doi: 10.1016/j.aaf.2022.11.005.
- [5] I. G. S. E. Putra and N. L. P. Labasaryani, "Sistem Telusur Produk Perikanan Berdasarkan Lokasi Pendaratan Kapal Menggunakan QR Code," *JOINTECS (Journal Inf. Technol. Comput. Sci.)*, vol. 5, no. 3, 2020, doi: 10.31328/jointecs.v5i3.1369.
- [6] Y. Fang and F. Asche, "Can U.S. import regulations reduce IUU fishing and improve production practices in aquaculture?," *Ecol. Econ.*, vol. 187, 2021, doi: 10.1016/j.ecolecon.2021.107084.
- [7] I. G. S. E. Putra, "Seafood Traceability System Based on Landing Site Using Batch Code Identifier," *Int. J. Comput. Sci. Eng.*, vol. 9, no. 4, 2020, doi: 10.21817/ijcsenet/2020/v9i4/200904026.
- [8] R. DiGregorio, *Tuna Grading and Evaluation*, First Edit. Urner Barry, 2012.
- [9] I. G. S. E. Putra, I. K. G. Darma Putra, M. Sudarma, and A. A. Kompiang Oka Sudana, "Classification of Tuna Meat Grade Quality Based on Color Space Using Wavelet and k-Nearest Neighbor Algorithm," in *2023 International Conference on Smart-Green Technology in Electrical and Information Systems (ICSGTEIS)*, Nov. 2023, pp. 35–40. doi: 10.1109/ICSGTEIS60500.2023.10424189.
- [10] T. Dmytrenko, T. Derkach, and A. Dmytrenko, "Using JAVA and C # Programming Languages For Server Platforms and Workstations," *Системи управління, навігації та зв'язку. Збірник наукових праць*, vol. 3, no. 73, 2023, doi: 10.26906/sunz.2023.3.093.
- [11] I. A. Becker, E. A. Yakimov, and N. P. Skrylyov, "Calculating the Complexity of an Algorithm Implemented in the C# Programming Language," *«System Anal. Appl. Inf. Sci.*, no. 4, 2024, doi: 10.21122/2309-4923-2023-4-14-19.
- [12] C. Hardman, *Game Programming with Unity and C#*. 2024. doi: 10.1007/978-1-4842-9720-9.
- [13] P. A. Bilokon, *Python, Data Science and Machine Learning*. 2025. doi: 10.1142/11701.
- [14] K. Kumar Mohbey and M. Acharya, "Python for Machine Learning," in *Basics of Python Programming: A Quick Guide for Beginners*, 2023. doi: 10.2174/9789815179637123010013.
- [15] B. Sobirov, G. Sinha, G. Chhabra, V. Vimal, C. Virmani, and S. Ganguli, "Application of Deep Convolutional Neural Networks Systems in Autonomous Vehicles," *Int. J. Syst. Syst. Eng.*, vol. 15, no. 1, 2025, doi: 10.1504/ijssse.2025.10059050.
- [16] B. S.P and S. C, "Deep Convolutional Neural Networks based Cervical Cancer Detection and Classification," *Int. J. Syst. Syst. Eng.*, vol. 15, no. 2, 2025, doi: 10.1504/ijssse.2025.10059072.
- [17] Purwono, A. Ma'arif, W. Rahmaniari, H. I. K. Fathurrahman, A. Z. K. Frisky, and Q. M. U. Haq, "Understanding of Convolutional Neural Network (CNN): A Review," *Int. J. Robot. Control Syst.*, vol. 2, no. 4, 2022, doi: 10.31763/ijrcs.v2i4.888.
- [18] A. Vanacore, M. S. Pellegrino, and A. Ciardiello, "Fair evaluation of classifier predictive performance based on binary confusion matrix," *Comput. Stat.*, vol. 39, no. 1, 2024, doi: 10.1007/s00180-022-01301-9.
- [19] E. Sentosa, H. Armanto, C. Pickerling, and L. Zaman PCSW, "Pengenalan Ekspresi Wajah dengan CNN dan Wavelet," *J. intelligent Syst. Comput.*, vol. 4, no. 02, pp. 69–76, 2022, doi: 10.52985/insyst.v4i2.209.
- [20] S. Khansa, "Penerapan Ekstraksi Ciri Transformasi Wavelet Dalam Pembuatan Model Klasifikasi Kesegaran Ikan Selar," *Senamika*, 2020.
- [21] S. Bahri, L. Awalushaumi, and N. Fitriyani, "Enhanced Performance of Dynamic Neural

- Network Model using Wavelet Activation Functions,” *Lontar Komput. J. Ilm. Teknol. Inf.*, vol. 14, no. 3, p. 150, 2023, doi: 10.24843/lkjiti.2023.v14.i03.p03.
- [22] W. Sun, Y. Ma, and R. Wang, “k-NN attention-based video vision transformer for action recognition,” *Neurocomputing*, vol. 574, 2024, doi: 10.1016/j.neucom.2024.127256.
- [23] J. R. Bermúdez, F. R. López-Estrada, G. Besançon, L. Torres, and I. Santos-Ruiz, “Leak-Diagnosis Approach for Water Distribution Networks based on a k-NN Classification Algorithm,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 16651–16656, 2020, doi: 10.1016/j.ifacol.2020.12.795.
- [24] Hayatou Oumarou and N. Rismayanti, “Automated Classification of Empon Plants: A Comparative Study Using Hu Moments and K-NN Algorithm,” *Indones. J. Data Sci.*, vol. 4, no. 3, 2024, doi: 10.56705/ijodas.v4i3.115.
- [25] Y. Zhang, Y. L. Ye, D. J. Guo, and T. Huang, “PCA-VGG16 model for classification of rock types,” *Earth Sci. Informatics*, vol. 17, no. 2, 2024, doi: 10.1007/s12145-023-01217-y.
- [26] S. Medhat, H. Abdel-Galil, A. E. Aboutabl, and H. Saleh, “Iterative magnitude pruning-based light-version of AlexNet for skin cancer classification,” *Neural Comput. Appl.*, vol. 36, no. 3, 2024, doi: 10.1007/s00521-023-09111-w.