

Combining U-NET Segmentation and Dimensionality Reduction Methods for K-NN Fish Freshness Classification

I Putu Yudia Artana¹, I Made Dwi Putra Asana², Ni Nyoman Ayu J. Sastaparamitha³, Ketut Jaya Atmaja⁴, Made Leo Radhitya⁵

^{1,2,3,4,5}Department of Informatics, Faculty of Technology and Informatics, Institut Bisnis dan Teknologi Indonesia (INSTIKI)
Jl. Tukad Pakerisan No. 97 Denpasar, Bali, Indonesia

e-mail: yudiaartana87@gmail.com¹, dwiputraasana@instiki.ac.id², ayusasta@instiki.ac.id³, ketutjayaatmaja@instiki.ac.id⁴, leo.radhitya@instiki.ac.id⁵

Received : January, 2025

Accepted : March, 2025

Published : April, 2025

Abstract

Accurate identification of tongkol fish freshness is important for the fisheries industry to ensure product quality. Conventional methods such as organoleptic testing are still subjective and can damage samples, so an automated approach based on image processing is needed. This study developed a tongkol fish freshness classification system with a combination of U-NET segmentation, color feature extraction in HSV space, dimensionality reduction using PCA or 2DPCA, and classification with K-Nearest Neighbors (K-NN). The dataset consists of 64 images of fish heads from Kedonganan Beach, Badung, Bali, which were tested organoleptically. After segmenting the Fish Eye ROI using U-NET, augmentation was performed to increase the amount of data to 640 images. The model was tested with various k values (5, 15, 25, 35, 45) in K-NN, using Group K-Fold ($k=8$) and cumulative variance optimization (50%-95%). The results show that the combination of U-NET+2DPCA is more efficient than the combination of U-NET+PCA, with the highest validation accuracy of 96.88% and a computation time of 9.13 seconds at a variance of 55% and $k = 25$. This combination of methods offers an accurate and fast solution for automatically detecting fish freshness, support the fishing industry in maintaining product quality.

Keywords: K-NN, PCA, 2DPCA, U-NET Segmentation, Fish Freshness

Abstrak

Identifikasi kesegaran ikan tongkol yang akurat penting bagi industri perikanan untuk memastikan kualitas produk. Metode konvensional seperti uji organoleptik masih subjektif dan dapat merusak sampel, sehingga diperlukan pendekatan otomatis berbasis pengolahan citra. Penelitian ini mengembangkan sistem klasifikasi kesegaran ikan tongkol dengan kombinasi segmentasi U-NET, ekstraksi fitur warna dalam ruang HSV, reduksi dimensi menggunakan PCA atau 2DPCA, serta klasifikasi dengan K-Nearest Neighbors (K-NN). Dataset terdiri dari 64 gambar kepala ikan dari Pantai Kedonganan, Badung, Bali, yang diuji secara organoleptik. Setelah segmentasi ROI Mata Ikan menggunakan U-NET, augmentasi dilakukan guna meningkatkan jumlah data menjadi 640 gambar. Model diuji dengan berbagai nilai k (5, 15, 25, 35, 45) pada K-NN, menggunakan Group K-Fold ($k=8$) dan optimisasi varians kumulatif (50%-95%). Hasil menunjukkan bahwa kombinasi U-NET+2DPCA lebih efisien dibandingkan kombinasi UNET+PCA, dengan akurasi validasi tertinggi sebesar 96,88% dan waktu komputasi selama 9,13 detik pada varians 55% dan $k=25$. Kombinasi metode ini menawarkan solusi akurat dan cepat untuk mendeteksi kesegaran ikan secara otomatis, mendukung industri perikanan dalam menjaga kualitas produk.

Kata Kunci: K-NN, PCA, 2DPCA, Segmentasi U-NET, Kesegaran Ikan

1. INTRODUCTION

Indonesia as the largest archipelagic country in the world has very abundant maritime resource potential, with a sea area reaching 3.2 million km² and an Exclusive Economic Zone water area reaching 2.9 million km². The fisheries sector is one of the backbones of the national economy, with marine capture fisheries production reaching 6,767,572 tons in 2021 or an increase of 4.2% from the previous year. One of the leading commodities is tongkol fish which in 2021 reached 593,901 tons[1]. However, the high potential faces challenges in handling the freshness of fish in tropical climates. Traditional handling with cooling is often less than optimal, causing fresh fish to be mixed with stale fish, thus affecting its selling value[2]. Manual identification of fish freshness, which is often carried out by traders or consumers, tends to be subjective, time-consuming and error-prone. [3]–[5]. Consumers, in particular, often have difficulty distinguishing fresh fish from stale fish. This opens up opportunities for irresponsible traders to deceive consumers by selling stale fish, which ultimately harms consumers both financially and health-wise. Consumption of fish that is not fresh or has rotted is a serious threat to public health. Fish that has rotted contains colonies of pathogenic bacteria such as *Escherichia coli*, *Salmonella sp.*, *Vibrio cholerae*, and *Staphylococcus aureus*, which can cause food poisoning, gastrointestinal infections, and systemic diseases such as typhus and cholera. The process of fish spoilage is triggered by natural enzymatic activity after maturity and the growth of microorganisms which is exacerbated by unhygienic handling during distribution and sales. Studies show that 70% of food poisoning cases in coastal areas are related to the consumption of low-quality fish, with symptoms such as acute diarrhea, vomiting, and fever[6].

Therefore, a technology-based system is needed to identify fish freshness automatically, objectively and efficiently. A hybrid method can be developed, namely by combining segmentation techniques to obtain ROI with classification techniques to provide decisions in determining freshness. Research [7] shows that the application of the U-NET architecture for fish eye Region of Interest (ROI) segmentation obtains an Intersection over Union (IoU) accuracy of 0.88, outperforming conventional

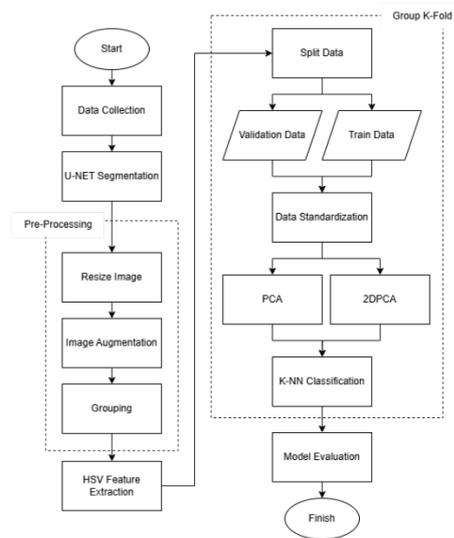
methods such as conventional segmentation as conducted in research [8]. The K-Means clustering method is used to perform segmentation, but it has several limitations, such as dependence on the right number of clusters, sensitivity to background and lighting variations, and lack of utilization of spatial information. K-Means only groups pixels based on color without considering spatial relationships, so the segmentation results can be less accurate and require additional processes for feature extraction. U-NET is superior because it can capture spatial information, is more robust to lighting and background variations, and does not require the selection of the number of clusters, making it a more effective solution than K-Means in tilapia image segmentation. Then in identifying and classifying freshness, image processing with the HSV color model can be a solution to determine the level of freshness of tongkol fish through eye color analysis. Changes of Hue, Saturation, and Value in fish eyes can reflect the level of freshness due to degradation of pigments and tissue structure[9]. Research [10] confirms that color feature information in the HSV color space extracted from fish eyeballs can be used to identify the freshness level of fish. The combination of image processing and machine learning methods such as K-NN has been used in previous studies for fish quality classification, but its accuracy can still be improved by optimizing the dimensionality reduction process such as using PCA or 2DPCA[11]–[13]. Research [11], [14] shows that classification accuracy using K-Nearest Neighbor (K-NN) can still be improved by optimizing the feature extraction process. One approach that has the potential to improve accuracy is to use Principal Component Analysis (PCA) for dimensionality reduction. PCA simplifies the data and allows focus on the most relevant features, thereby reducing computational complexity and improving model performance. This is supported by research [15] which shows that the use of PCA in COVID-19 classification on X-ray images successfully increased model accuracy from 97.6% to 100%, while reducing the risk of overfitting. In addition, research [16] confirmed that PCA can overcome the curse of dimensionality problem in malaria genomic data. By combining PCA with the Genetic Algorithm (GA) optimization algorithm, their research achieved an accuracy of up to 90% using K-NN. Research [17] further

strengthens the role of PCA in image processing by showing that dimensionality reduction using PCA can retain important information in salmon image data, resulting in a freshness prediction accuracy of 92.3%. In addition to PCA, the Two-Dimensional PCA (2DPCA) method also shows great potential in improving classification accuracy. Research by [13] shows that 2DPCA has advantages over PCA in terms of computation time and accuracy because the covariance matrix is calculated directly from the two-dimensional image matrix, without the need to convert the image into a one-dimensional vector first. In their study, the combination of 2DPCA with K-NN produced the highest accuracy of 96.88%, compared to PCA + KNN which only reached 89.38%.

This study proposes a combination of the U-NET method with dimensionality reduction techniques (PCA/2DPCA) and K-NN to develop an image-based tongkol freshness identification model. Previous studies have not integrated automatic ROI detection with dimensionality reduction techniques before classification, so there are still gaps that need to be further studied. In addition, evaluations in previous studies have focused more on accuracy, without considering computational efficiency. Therefore, this study aims to develop an automatic ROI detection system and analyze the effect of PCA and 2DPCA in reducing image dimensions on the accuracy of fish freshness classification using K-NN. The results of this study are expected to evaluate the effectiveness of the combination of UNET, PCA or 2DPCA, and K-NN in improving accuracy and computational efficiency. In addition, this study also aims to produce an automatic fish freshness detection system based on fish head images.

2. RESEARCH METHODS

Picture 1 shows the research flow which includes data collection, fisheye area segmentation with U-NET, pre-processing (resizing, augmentation, and grouping), HSV feature extraction, data splitting, standardization, dimensionality reduction with PCA/2DPCA, and K-NN classification as well as model evaluation to measure its performance and efficiency.



Picture 1. Research Flow

2.1 Data Collection

Image data collection was carried out on tongkol fish obtained directly from fishermen at Kedonganan Beach, Badung, Bali. The images were grouped into 2 classes of fish freshness. First, the fresh fish class, namely tongkol fish obtained from fishermen who had just arrived from the beach, was tested at the UPTD. PPMHP Bali (Testing and Implementation of Fishery Product Quality) using the organoleptic method and obtained a score of $\bar{X} \geq 7$ (fish is categorized as fresh). Second, the non-fresh fish class, namely tongkol fish that had been stored for 10-12 hours at room temperature, was then tested organoleptically and obtained a score of $\bar{X} < 7$ (fish is categorized as not fresh). The parameters used in this organoleptic test include odor, color and texture of meat, eyes, gills, and mucus. These parameters will be used to determine freshness based on the scores of each parameter. A total of 64 validated images were obtained in .jpg format with a size of 3000x3000 pixels. The image obtained is an image of a fish head, which will then be extracted only from the eyes to determine its freshness.

2.2 U-NET Segmentation

The U-Net model is used to extract the Region of Interest (ROI) in the form of the eye area. After that, the extraction results will be used for the dimension reduction process in the next stage.



Picture 2. ROI Extraction using U-NET Segmentation

As seen in Picture 2, the result of U-NET segmentation is a prediction mask in the form of a segmentation map with continuous values indicating the probability of the eye area. This mask is then binarized using a threshold of 0.5 to separate the eye area (value 1) from the background (value 0). Furthermore, the binary mask is resized back to the original image dimensions using linear interpolation. To improve the segmentation quality, the mask is smoothed using Gaussian Blur with an 11x11 kernel. This step helps reduce noise and smooth the edges of the mask. The smoothed mask is then re-binarized to maintain the clarity of the eye area. After that, a morphological opening operation is performed which is a combination of erosion and dilation using a 5x5 kernel. This step is useful for removing noise or small artifacts on the mask and strengthening the main structure of the eye area. The final mask is then used to detect contours. To ensure smoother and more unified contour results, a convex hull is used. Finally, a contour-shaped mask is used to crop the original image. This process is done by applying the mask to the image using a bitwise AND operation, followed by cropping based on the contour bounding box.

2.3 Data Pre-processing

2.3.1 Resize

The preprocessing process begins by resizing all segmented images to 128x128 pixels. This aims to ensure consistency of input sizes in the model and reduce computational complexity without losing important information.

2.3.2 Data Augmentation

After resizing, the data is enlarged through augmentation techniques to increase image variation and strengthen the model's generalization ability. The augmentation techniques used include:

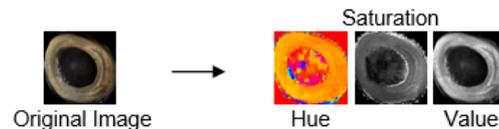
- 1) Rotation: The image is rotated randomly within the range of -45° to 45° , either clockwise or counterclockwise.
- 2) Flipping: Horizontal and vertical flips are performed to simulate various image orientations.
- 3) Shear: The image experiences shear distortion with a range of ± 0.2 , which changes the perspective shape of the image without shifting its center point.

The selection of this augmentation technique aims to replicate various possible real-world conditions, such as shooting angles and perspectives. Thus, the model can be more robust in recognizing relevant patterns and avoiding overfitting to limited training data. After the augmentation process, the total amount of data increased to 640 images.

2.3.3 Grouping

This grouping process is carried out with the aim of avoiding data leakage, the images are grouped into 32 groups, consisting of 16 groups for fresh fish, and 16 groups for non-fresh fish. Each group contains 20 fish images that correspond to the same sample including augmented samples. This group division ensures that data from the same group is not used simultaneously in the training and validation sets.

2.4 HSV Feature Extraction



Picture 3. BGR to HSV Transformation

Picture 3 shows the process of extracting HSV features separately from BGR images. HSV color space represents color with three components. Hue (H), which indicates the type of color. Saturation (S), which indicates the intensity or saturation of the color. Value (V), which represents the brightness level of the color.

2.5 Group K-Fold Cross Validation

The validation process is carried out using Group K-Fold Cross Validation with $k=8$. This technique divides the dataset into 8-folds and goes through 8 iterations/training scenarios, where in each iteration, 7-folds are used for training and 1-fold for validation. So, this technique ensures that no data from the same group appears in both sets (training and validation). In training, each iteration consists of 28 different groups, namely 14 groups from the fresh class and 14 groups from the non-fresh class. In validation, each iteration consists of 4 different groups, namely 2 groups from the fresh class and 2 groups from the non-fresh class. This approach provides a more representative evaluation of model performance while reducing the risk of

overfitting due to data similarities between the training and validation sets.

2.6 Data Standardization

After the data is divided into training set and validation set, standardization is then carried out on each channel in the HSV feature. Standardization on training and validation is done separately, on training data it is done by calculating the average and standard deviation of the original data, then transformation is carried out. While on testing data, transformation is carried out using parameters from training data to ensure that there is no information leakage from testing data during the training process. The standardization process is repeated on each iteration/scenario Group K-Fold. Data standardization can be calculated using equation 1.

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

Where z is the data value, μ is the mean the feature, and σ is the standard deviation of the feature[18].

2.7 PCA (Principal Component Analysis)

Before applying PCA, each image pixel matrix is transformed into a one-dimensional vector (flattening). PCA is then applied separately to each color channel (Hue, Saturation, and Value) to calculate principal components and project the data into a lower-dimensional space. The optimal number of principal components (n) is determined based on the target cumulative variance to be retained in the data, which ranges from 50% to 95%. The higher the target cumulative variance, the more principal components are retained, but this can also increase the computational complexity. This PCA process is performed only on the training data to calculate the transformation parameters (mean and eigenvectors), which are then used to transform the validation data. Then, the reduced features from the three channels are combined horizontally to form the final data set. The number of PCA principal components for each target cumulative variance used in this study can be seen in Table 1.

Table 1: Number of PCA Component for Each Channel and Iteration Based on Cumulative Variance Target

		Cumulative Variance										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	
Number Of Components Hue (n_component)	iteration	1	15	23	34	50	72	102	143	198	270	370
		2	15	22	33	48	68	97	136	189	260	361
		3	14	21	31	45	65	94	133	187	259	361
		4	18	26	37	53	75	106	148	203	276	376
		5	16	24	35	51	73	104	146	201	274	375
		6	17	25	36	52	75	105	146	201	273	373
		7	17	25	37	53	75	105	147	202	274	375
		8	15	23	34	50	72	103	144	199	272	373
Number Of Components Saturation (n_component)	iteration	1	7	11	17	27	43	69	109	167	248	363
		2	6	9	14	23	37	59	96	152	234	354
		3	6	10	15	25	40	64	102	159	241	359
		4	6	9	15	23	38	61	98	154	236	354
		5	7	10	16	25	40	64	102	159	241	359
		6	7	10	16	25	40	64	101	159	240	358
		7	7	10	15	24	39	63	101	158	240	358
		8	6	8	13	21	36	59	96	152	235	354
Number Of Components Value (n_component)	iteration	1	9	11	15	19	26	36	51	75	118	210
		2	9	12	16	21	28	38	53	78	120	211
		3	10	12	16	21	28	39	54	80	124	217
		4	9	12	15	20	27	37	53	78	122	215
		5	9	12	15	20	27	37	52	77	121	213
		6	8	11	14	19	25	35	49	72	113	202
		7	9	12	15	20	27	37	53	77	120	210
		8	8	11	14	19	25	34	49	72	114	205

Table 1 shows the number of principal components (n) generated for different cumulative variance targets in reducing feature dimensionality. For example, in the hue channel with a target of 95% and iteration 1, only 370 principal components are needed to explain 95% of the data variance. Thus, in a sample initially consisting of 16,384 features, 95% of the data variance can be explained by only 370 principal components. The value of n varies across channels and iterations due to differences in variance distribution, degree of interpixel correlation, and noise. Channels with high variance require fewer components, while channels with low noise or correlation require more components to achieve the same target [19].

2.8 2DPCA (Two-Dimensional Principal Component Analysis)

Unlike traditional PCA that converts images into 1D vectors before dimensionality reduction, 2DPCA directly reduces images in the form of 2D matrices. After HSV color channel extraction, image data is centered by subtracting the mean pixel value, followed by the calculation of the

covariance matrix to capture the relationship between pixels in the spatial dimension, including the distribution pattern between pixel rows and columns. Eigen decomposition is performed to obtain eigenvectors and eigenvalues, where the eigenvector with the largest eigenvalue is selected as the principal component. The number of principal components is determined based on the target cumulative variance reflecting the proportion of information retained. Projection is performed directly on the height and width dimensions of the image, resulting in a more comprehensive representation than PCA with 1D. The reduction process performed separately on the height and width dimensions doubles the processing, allowing for deeper pattern analysis. The 2DPCA projection results of each color channel are then combined to form the final feature representation. The number of 2DPCA principal components on the wide side for each cumulative variance target used in this study can be seen in full in Table 2 and the number of 2DPCA principal components on the high side for each cumulative variance target used in this study can be seen in full in Table 3.

Table 2: Number of 2DPCA Components in the Width Section for Each Channel and Iteration Based on the Cumulative Variance Target

		Cumulative Variance										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	
Number Of Components Hue (n_component)	iteration	1	4	5	7	8	11	14	20	28	42	66
		2	4	5	7	8	11	14	19	28	41	65
		3	4	5	6	8	10	14	19	27	41	65
		4	5	6	7	9	11	15	20	29	42	67
		5	4	5	7	8	11	14	20	29	42	67
		6	4	6	7	9	11	15	20	29	43	67
		7	4	6	7	9	11	15	20	29	43	67
		8	4	5	7	8	11	14	20	29	43	67
Number Of Components Saturation (n_component)	iteration	1	3	4	5	6	8	11	16	25	40	65
		2	3	3	4	6	7	10	14	22	37	63
		3	3	3	4	6	8	10	15	24	39	64
		4	3	3	4	6	7	10	14	23	38	63
		5	3	3	4	6	8	10	15	23	38	64
		6	3	4	5	6	8	10	15	24	38	63
		7	3	3	4	6	8	10	15	23	38	64
		8	3	3	4	5	7	10	14	23	38	63
Number Of Components Value (n_component)	iteration	1	3	4	4	5	6	7	8	11	16	26
		2	3	4	4	5	6	7	9	12	16	27
		3	3	4	4	5	6	7	9	12	16	27
		4	3	4	4	5	6	7	9	11	16	27
		5	3	4	4	5	6	7	9	11	16	26
		6	3	4	4	5	6	7	8	11	15	25
		7	3	4	4	5	6	7	9	11	16	26
		8	3	4	4	5	6	7	8	11	16	26

Table 2 shows the number of principal components (n) generated for various cumulative variance targets in reducing the feature dimensionality in the width dimension.

For example, in the hue channel with a target of 95% and iteration 1, only 66 principal components are needed to explain 95% of the data variance.

Table 3: Number of 2DPCA Components in the Height Section for Each Channel and Iteration Based on the Cumulative Variance Target

		Cumulative Variance										
		55%	55%	60%	65%	70%	75%	80%	85%	90%	95%	
Number Of Components Hue (n_component)	iteration	1	1	2	2	3	5	7	10	15	29	57
		2	2	2	3	3	5	7	9	15	27	56
		3	1	2	2	3	4	6	9	14	26	55
		4	2	2	3	4	5	7	10	16	28	58
		5	1	2	3	3	5	7	10	15	28	58
		6	1	2	3	4	5	7	10	15	28	57
		7	1	2	3	4	5	7	10	15	29	58
		8	1	2	3	3	5	7	10	15	29	58
Number Of Components Saturation (n_component)	iteration	1	1	2	2	2	3	5	7	12	23	54
		2	1	2	2	2	3	4	6	10	20	51
		3	1	2	2	2	3	4	7	11	22	53
		4	1	2	2	2	3	4	6	10	21	52
		5	1	2	2	2	3	4	7	11	21	53
		6	1	2	2	2	3	4	7	11	21	52
		7	1	2	2	2	3	4	7	10	21	53
		8	1	1	2	2	3	4	6	10	20	51
Number Of Components Value (n_component)	iteration	1	2	3	3	4	4	5	6	7	11	19
		2	2	3	3	4	4	5	6	8	11	19
		3	3	3	3	4	5	5	6	8	11	19
		4	3	3	3	4	5	5	6	7	11	19
		5	2	3	3	4	4	5	6	7	11	19
		6	2	3	3	4	4	5	6	7	11	18
		7	2	3	3	4	4	5	6	7	11	19
		8	2	3	3	3	4	5	6	7	10	18

Table 3 shows the number of principal components (n) generated for various cumulative variance targets in reducing the dimensionality of features in high dimensions. For example, in the hue channel with a target of 95% and the 1st iteration, only 57 principal components are needed to explain 95% of the data variance.

achieve the same variance target. This difference reflects the diversity of spatial patterns in the data, so each channel requires several principal components that are adjusted to its characteristics to ensure optimal data representation according to the cumulative variance target to be achieved[20].

After the dimensionality reduction process in the hue channel and the 1st iteration with a cumulative variance of 95%, the image that was originally 128x128 will be reduced to 66x57 in the hue channel. The variation in the number of principal components between channels and iterations is caused by differences in variance distribution, correlation between pixels, and noise. Channels with high variance require fewer components, while channels with noise or low correlation require more components to

2.9 K-NN (K-Nearest Neighbors)

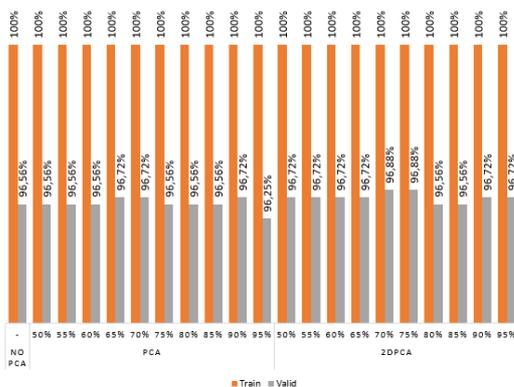
The KNN model is trained using the training data and then the 'k' nearest neighbors of the validation data are identified based on the distance in the feature space to evaluate its performance. The class or label of the validation data is determined based on the majority class of the 'k' nearest neighbors. In this study, five k parameters are used, namely k = 5, k = 15, k = 25, k = 35, k = 45. The determination of the five k parameters in this study is based on previous studies showing that the selection of the k value

greatly affects the performance of KNN. Research [21] stated that k that is too small can make the model too sensitive to noise, while k that is too large can reduce the sharpness of the classification. Meanwhile, research [22] found that a smaller k value tends to be more optimal in low noise conditions, while increasing k can decrease accuracy. Therefore, the variation of k values used in this study aims to evaluate the balance between model generalization and sensitivity to changes in the data.

3. RESULTS AND DISCUSSION

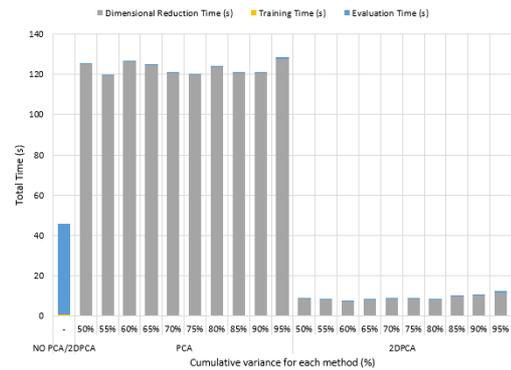
This research was conducted using Google Collaboratory which utilizes 12.7 GB of CPU and RAM computing power. The optimal parameters for the KNN model, such as the number of nearest neighbors (k) and cumulative variance for PCA and 2DPCA, were determined through a grid search process with cross-validation. The grid search results showed the best combination of parameters that produced the highest accuracy on the validation data and were computationally efficient, which were then used to build the final classification model.

3.1 Value $k = 5$



Picture 4. Train Validation Accuracy in PCA and 2DPCA with $k=5$

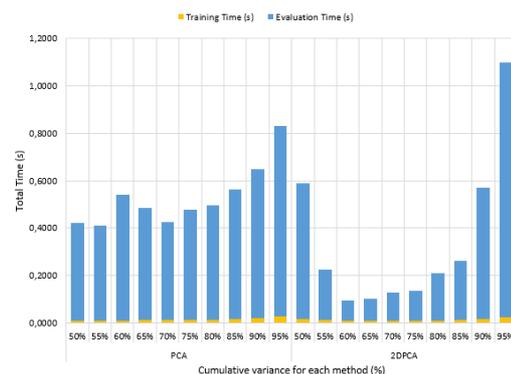
Based on the results visualized in Picture 4, the KNN model with $k=5$ achieved the highest validation accuracy of 96.88% using 2DPCA at a cumulative variance of 70-75%. Meanwhile, PCA achieved optimal results at cumulative variances of 65%, 70%, 90% with a validation accuracy of 96.72%. There is a high gap between training accuracy and validation accuracy in each experiment. The combination of 2DPCA with a cumulative variance of 70-75% shows the smallest gap value of 3.12% which still indicates the potential for overfitting.



Picture 5. Total Computing Time in NO PCA/2DPCA, PCA and 2DPCA with $k=5$

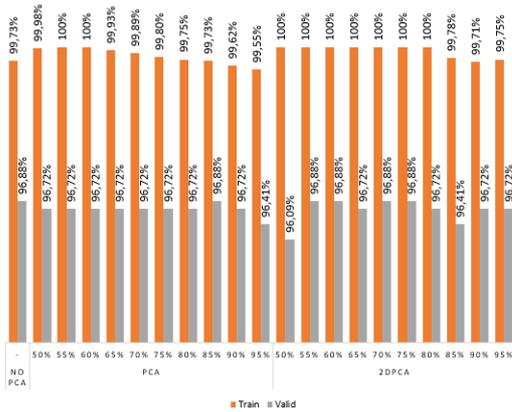
Picture 5 shows a comparison of the total computation time, consisting of the combination of dimension reduction time, training time, and evaluation time, for $k=5$. From the graph, it can be observed that without the application of dimension reduction, the evaluation time increases significantly to 45.26 seconds. The 2DPCA method has advantages in terms of computational speed, especially in the dimension reduction stage. The average computation time required by 2DPCA for each cumulative variance is 9.11 seconds, much faster than PCA which requires an average time of 123.45 seconds.

Then, when viewed in terms of training and evaluation time, 2DPCA excels at cumulative variances of 55%-90% then PCA excels at cumulative variances of 50% and 95% as seen in Picture 6.



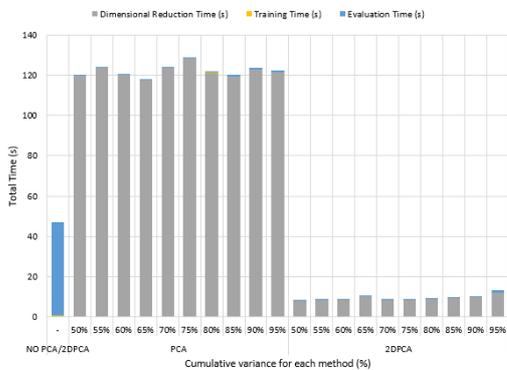
Picture 6. Training and Evaluation Time in PCA and 2DPCA with $k=5$

3.2 Value $k = 15$



Picture 7. Train Validation Accuracy in PCA and 2DPCA with k=15

Based on the visualization results in Picture 7, the KNN model with $k = 15$ obtained the highest validation accuracy of 96.88% in 2DPCA, PCA, and without dimension reduction (NO PCA/2DPCA). 2DPCA excels at cumulative variance of 55%-75%, and at cumulative variance of 85% the accuracy produced by PCA begins to provide better accuracy results. There is a high gap between training accuracy and validation accuracy in each experiment. The smallest gap achieved with $k=15$ is 2.86% and this is still relatively high and indicates the potential for overfitting.

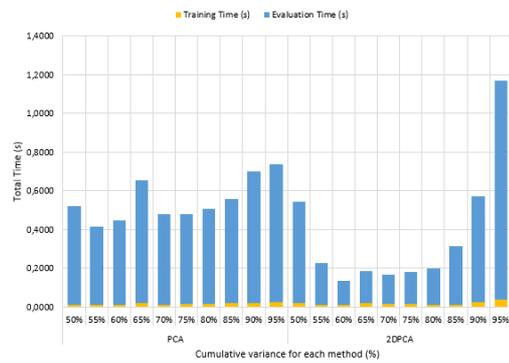


Picture 8. Total Computing Time in NO PCA/2DPCA, PCA and 2DPCA with k=15

Picture 8 shows a comparison of the total computation time, consisting of the combination of dimensionality reduction time, training time, and evaluation time, for $k=15$. From the graph, it can be observed that without the application of dimensionality reduction, the evaluation time increases significantly to 46.23 seconds. The 2DPCA method has advantages in terms of computational speed, especially in the dimensionality reduction stage. The average computation time required by 2DPCA for each cumulative variance is 9.59 seconds, much

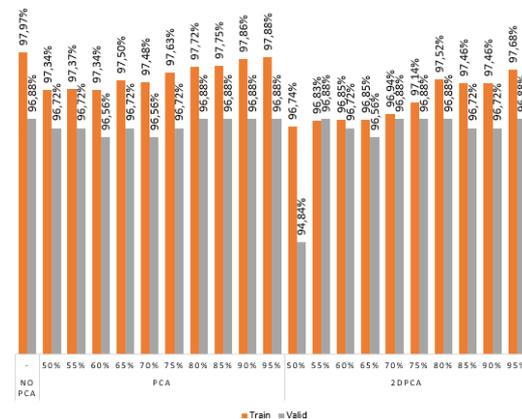
faster than PCA which requires an average time of 122.44 seconds.

Then, when viewed in terms of training and evaluation time, 2DPCA excels at cumulative variances of 55%-90% then PCA excels at cumulative variances of 50% and 95% as seen in Picture 9.



Picture 9. Training and Evaluation Time in PCA and 2DPCA with k=15

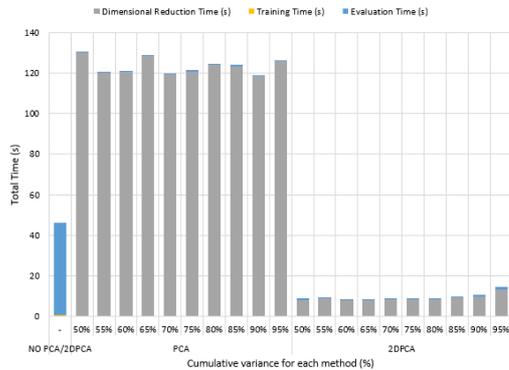
3.3 Value $k = 25$



Picture 10. Train Validation Accuracy in PCA and 2DPCA with k=25

Based on the visualization results in Picture 10, the KNN model with $k=25$ obtained the highest validation accuracy of 96.88% in 2DPCA, PCA, and without dimension reduction (NO PCA/2DPCA). 2DPCA achieved the highest accuracy at cumulative variances of 55%, 70%, 75%, 80%, then at cumulative variances of 80%-95% the validation accuracy produced by PCA began to provide better accuracy results compared to 2DPCA. At a value of $k=25$, the gap between training accuracy and validation accuracy in each experiment became smaller and showed significant improvement. The combination of 2DPCA with a cumulative

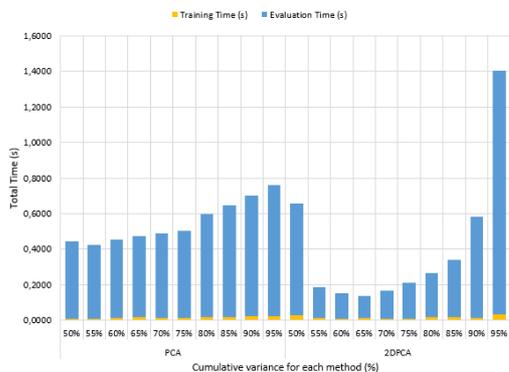
variance of 55% produced the smallest gap of (-0.04%). This shows that the model is able to generalize data well.



Picture 11. Total Computing Time in NO PCA/2DPCA, PCA and 2DPCA with k=25

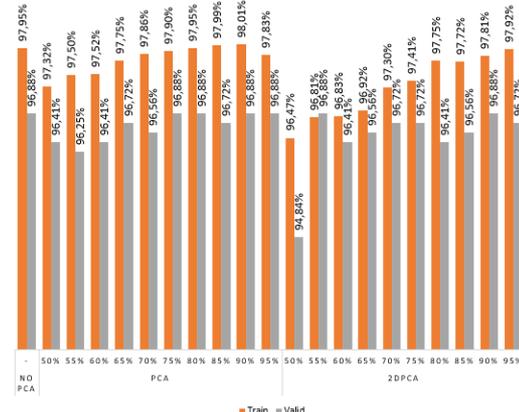
Picture 11 shows a comparison of the total computation time, consisting of the combination of dimension reduction time, training time, and evaluation time, for k=25. From the graph, it can be observed that without the application of dimension reduction, the evaluation time increases significantly to 45.59 seconds. The 2DPCA method has advantages in terms of computational speed, especially in the dimension reduction stage. The average computation time required by 2DPCA for each cumulative variance is 9.49 seconds, much faster than PCA which requires an average time of 123.70 seconds.

Then, when viewed in terms of training and evaluation time, 2DPCA excels at cumulative variances of 55%-90% then PCA excels at cumulative variances of 50% and 95% as seen in Picture 12.



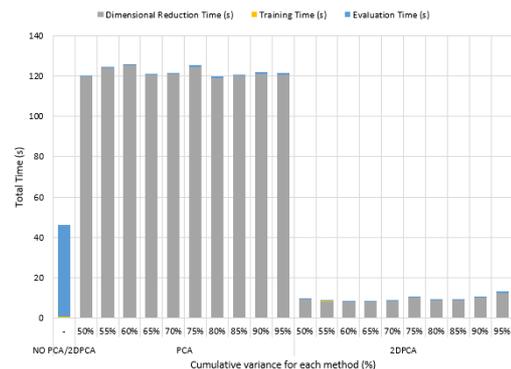
Picture 12. Training and Evaluation Time in PCA and 2DPCA with k=25

3.4 Value k = 35



Picture 13. Train Validation Accuracy in PCA and 2DPCA with k=35

Based on the visualization results in Picture 13, the results show that the KNN model with k=35 maintains the highest validation accuracy of 96.88% on most cumulative targets, with 2DPCA, PCA, or without dimension reduction (NO PCA / 2DPCA). 2DPCA achieves the highest accuracy at cumulative variances of 55% and 90%, then at cumulative variances of 75%, 80%, 90%, and 95%, PCA achieves its highest validation accuracy. At a value of k=35, the gap between training accuracy and validation accuracy in each experiment still looks good and relatively small but not as good as at k=25. The combination of 2DPCA with a cumulative variance of 55% produces the smallest gap of (-0.07%). This shows that the model is still able to generalize the data well.

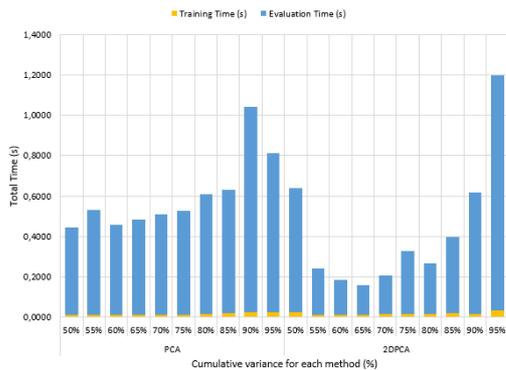


Picture 14. Total Computing Time in NO PCA/2DPCA, PCA and 2DPCA with k=35

Picture 14 shows a comparison of the total computation time, consisting of the combination of dimensionality reduction time, training time, and evaluation time, for k=35. From the graph, it can be observed that without the application of dimensionality reduction, the

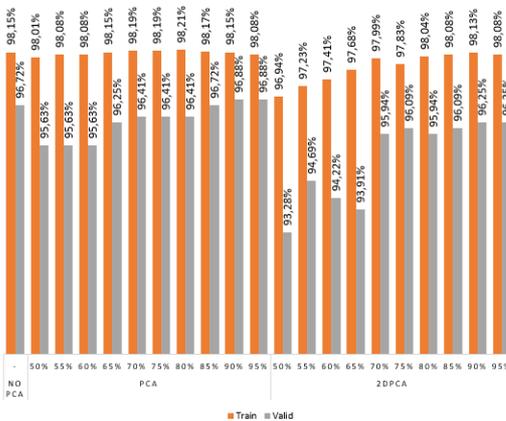
evaluation time increases significantly to 45.34 seconds. The 2DPCA method has advantages in terms of computational speed, especially in the dimensionality reduction stage. The average computation time required by 2DPCA for each cumulative variance is 9.68 seconds, much faster than PCA which requires an average time of 122.31 seconds.

Then, when viewed in terms of training and evaluation time, 2DPCA excels at cumulative variances of 55%-90% then PCA excels at cumulative variances of 50% and 95% as seen in Picture 15.



Picture 15. Training and Evaluation Time in PCA and 2DPCA with k=35

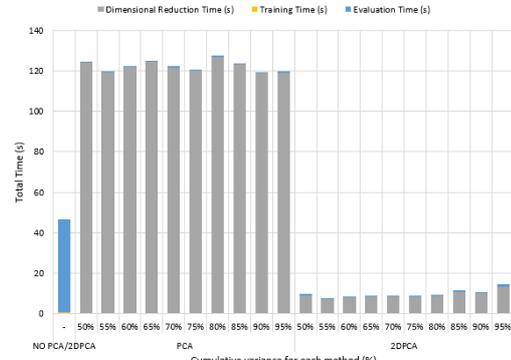
3.5 Value k = 45



Picture 16. Train Validation Accuracy in PCA and 2DPCA with k=45

Based on the visualization results in Picture 16, the results show that the KNN model with k=45 maintains the highest validation performance achieved at 90-95% cumulative variance, with a validation accuracy of 96.88% for PCA. PCA consistently shows higher accuracy than 2DPCA. At k=45 the gap value between training accuracy and validation accuracy in each experiment again shows a high gap. The smallest gap value

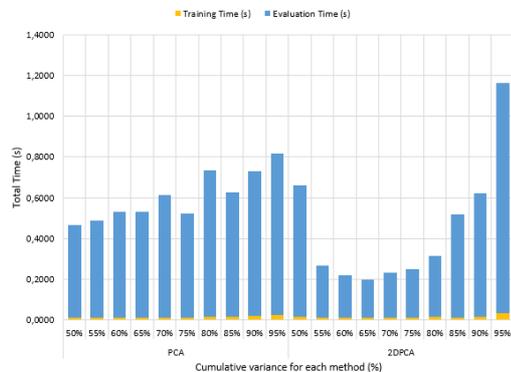
achieved in the combination of PCA with 95% cumulative variance shows a gap value of 1.21% which begins to show the potential for overfitting.



Picture 17. Total Computing Time in NO PCA/2DPCA, PCA and 2DPCA with k=45

Picture 17 shows a comparison of the total computation time, consisting of the combination of dimension reduction time, training time, and evaluation time, for k=45. From the graph, it can be observed that without the application of dimension reduction, the evaluation time increases significantly to 46.06 seconds. The 2DPCA method has advantages in terms of computation speed, especially in the dimension reduction stage. The average computation time required by 2DPCA for each cumulative variance is 9.77 seconds, much faster than PCA which requires an average time of 122.54 seconds.

Then, when viewed in terms of training and evaluation time, 2DPCA excels at cumulative variances of 55%-90% then PCA excels at cumulative variances of 50% and 95% as seen in Picture 18.



Picture 18. Training and Evaluation Time in PCA and 2DPCA with k=45

Based on the results of the research that has been conducted, the effect of using PCA and 2DPCA on model performance varies depending on the value of k . At low k values ($k=5$), neither PCA nor 2DPCA showed a significant increase in accuracy compared to the model without dimensionality reduction. This shows that at small k values, the contribution of dimensionality reduction to accuracy is still limited. Conversely, at higher k values ($k=45$), PCA tends to provide more stable accuracy results than 2DPCA or models without dimensionality reduction, especially at cumulative variances of 90% -95%. In addition, the accuracy gap between training and validation also depends on the value of k . At low k values ($k=5$), the accuracy gap tends to be larger, indicating potential overfitting. At medium k values ($k=25$), the accuracy gap between training and validation becomes smaller, especially when using a combination of 2DPCA with a cumulative variance target of 55%. This combination produces a very small difference in training and validation accuracy (-0.04%), indicating excellent model generalization capabilities. However, at high k values ($k=45$), the gap between training and validation accuracies begins to increase again. Thus, selecting the optimal k value and the appropriate dimensionality reduction method (PCA or 2DPCA) play an important role in achieving a balance between accuracy, stability, and generalization of the KNN model. In terms of time efficiency, 2DPCA is superior in the dimensionality reduction process compared to PCA, with a shorter computation time. However, in terms of training and evaluation time, 2DPCA is more efficient at variances of 55%-85%, while PCA has advantages at variances of 50% and 90%-95%, although the difference in efficiency is not too significant. Overall, PCA and 2DPCA successfully improve the stability of the KNN model, although their effects on accuracy are relatively small and need to be considered in the context of time efficiency. 2DPCA shows significant computational efficiency, especially in dimensionality reduction.

This study has several limitations, especially the reliance on U-NET segmentation which can be affected by noise and ROI accuracy. Inaccuracy in segmentation affects the extraction of color features and the performance of the K-NN model. The U-NET segmentation model in this study still has a loss of 0.014. Practically, this

model has the potential to be applied in an automated system to detect fish freshness in the market or fishing industry. However, further testing is needed in real conditions with varying lighting and backgrounds. Integration into mobile devices or embedded systems can make it easier for users to classify fish freshness in real time.

The main contribution of this study is the combination of U-NET, PCA/2DPCA, and K-NN for tongkol fish freshness classification based on HSV imagery, which has not been widely applied in other studies. Evaluation of the effectiveness of PCA and 2DPCA in terms of accuracy and computational efficiency is an important aspect in real implementation.

4. CONCLUSION

This study shows that the combination of U-NET methods for segmentation, dimensionality reduction using PCA or 2DPCA, and classification with KNN can be used effectively in tongkol fish freshness classification based on fish eye images. The experimental results reveal that the selection of appropriate parameters, especially the value of k in KNN and the cumulative variance of dimensionality reduction, play an important role in achieving high accuracy and model stability. The selection of the best model considers three main factors, namely high validation accuracy, minimal accuracy gap between training and validation to prevent overfitting or underfitting, and efficiency of total computation time. The combination of U-NET+2DPCA with a cumulative variance of 55%, and KNN with $k = 25$ is the best combination of parameters, resulting in a validation accuracy of 96.88%, with a very small difference in accuracy between training and validation (-0.04%). This shows that the model is able to generalize well without experiencing overfitting or underfitting. In addition, the computation time ensures the efficiency of this method can be applied in real scenarios. Overall, the results of this study indicate that a deep learning-based approach to segmentation combined with dimensionality reduction techniques and distance-based classification can be an effective solution in determining tongkol fish freshness based on fish eye images.

REFERENCES

- [1] BPS, "Produksi Perikanan Tangkap di

- Laut Menurut Komoditas Utama (Ton), 2020-2021." Accessed: Jun. 19, 2024. [Online]. Available: <https://www.bps.go.id/id/statistics-table/2/MTUxNSMy/produksi-perikanan-tangkap-di-laut-menurut-komoditas-utama.html>
- [2] R. A. Amaral *et al.*, "Chemical-Based Methodologies to Extend the Shelf Life of Fresh Fish—A Review," *Foods*, vol. 10, no. 10, p. 2300, Sep. 2021, doi: 10.3390/foods10102300.
- [3] X. Guo, S. Gu, and D. Wei, "Fish freshness evaluation based on MobileNetV1 and attention mechanism," in *Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering*, New York, NY, USA: ACM, Oct. 2022, pp. 270–276. doi: 10.1145/3573428.3573475.
- [4] H. Mohammadi Lalabadi, M. Sadeghi, and S. A. Mireei, "Fish freshness categorization from eyes and gills color features using multi-class artificial neural network and support vector machines," *Aquac. Eng.*, vol. 90, no. October 2019, p. 102076, 2020, doi: 10.1016/j.aquaeng.2020.102076.
- [5] K. Takano, X. Lu, C. Yuan, and A. Kimura, "Freshness prediction for seafood using deep neural network," in *International Workshop on Advanced Imaging Technology (IWAIT) 2023*, M. Nakajima, P. Y. Lau, J.-G. Kim, T. Yamasaki, K. Seo, J.-M. Guo, and Q. Kema, Eds., SPIE, Mar. 2023, p. 58. doi: 10.1117/12.2666963.
- [6] N. Rahmi, P. Wulandari, and L. Advinda, "Pengendalian Cemaran Mikroorganisme pada Ikan— Mini Review," *Pros. Semin. Nas. Biol.*, vol. 1, no. 2, pp. 611–623, 2022, [Online]. Available: <https://semnas.biologi.fmipa.unp.ac.id/index.php/prosiding/article/view/170>
- [7] F. N. Fajri, M. Dzikrillah, and A. Khairi, "Digital Fish Image Segmentation Using U-Net for Shape Feature Extraction," *J. Teknol. DAN OPEN SOURCE*, vol. 7, no. 2, pp. 195–201, Dec. 2024, doi: 10.36378/jtos.v7i2.3968.
- [8] R. Nuraini, "Implementasi Euclidean Distance dan Segmentasi K-Means Clustering Pada Identifikasi Citra Jenis Ikan Nila," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 3, no. 1, pp. 1–8, 2022.
- [9] C. Y. Jerandu, P. Batarius, and A. A. J. Sinlae, "Identifikasi Kualitas Kesegaran Ikan Menggunakan Algoritma K-Nearest Neighbor Berdasarkan Ekstraksi Ciri Warna Hue, Saturation, dan Value (HSV)," *Build. Informatics, Technol. Sci.*, vol. 4, no. 3, Dec. 2022, doi: 10.47065/bits.v4i3.2613.
- [10] J. Sandra, "Penggunaan Ciri Warna Hsv Pada Bola Mata Ikan Untuk Identifikasi Tingkat Kesegaran Ikan Menggunakan Algoritma KNN," *J. Teknol. Inf.*, vol. 2, no. 2, p. 85, Nov. 2023, doi: 10.35308/jti.v2i2.7711.
- [11] I. Gede Sujana Eka Putra, I. Ketut Gede Darma Putra, M. Sudarma, and A. A. K. O. Sudana, "Classification of Tuna Meat Grade Quality Based on Color Space Using Wavelet and k-Nearest Neighbor Algorithm," *Proc. - Int. Conf. Smart-Green Technol. Electr. Inf. Syst. ICSGTEIS*, no. November, pp. 35–40, 2023, doi: 10.1109/ICSGTEIS60500.2023.10424189.
- [12] B. M. S. Hasan and A. M. Abdulazeez, "A Review of Principal Component Analysis Algorithm for Dimensionality Reduction," *J. Soft Comput. Data Min.*, vol. 2, no. 1, pp. 20–30, 2021, doi: 10.30880/jsedm.2021.02.01.003.
- [13] S. Sutarti, A. T. Putra, and E. Sugiharti, "Comparison of PCA and 2DPCA Accuracy with K-Nearest Neighbor Classification in Face Image Recognition," *Sci. J. Informatics*, vol. 6, no. 1, pp. 64–72, May 2019, doi: 10.15294/sji.v6i1.18553.
- [14] M. B. Yildiz, E. T. Yasin, and M. Koklu, "Fisheye freshness detection using common deep learning algorithms and machine learning methods with a developed mobile application," *Eur. Food Res. Technol.*, no. 0123456789, 2024, doi: 10.1007/s00217-024-04493-0.
- [15] J. Rasheed, A. A. Hameed, C. Djeddi, A. Jamil, and F. Al-Turjman, "A machine learning-based framework for diagnosis of COVID-19 from chest X-ray images," *Interdiscip. Sci. – Comput. Life Sci.*, vol. 13, no. 1, pp. 103–117, 2021, doi: 10.1007/s12539-020-00403-6.
- [16] M. O. Arowolo, M. O. Adebisi, A. A.

- Adebiyi, and O. Olugbara, "Optimized hybrid investigative based dimensionality reduction methods for malaria vector using KNN classifier," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00415-z.
- [17] T. Wu, L. Yang, J. Zhou, D. C. Lai, and N. Zhong, "An improved nondestructive measurement method for salmon freshness based on spectral and image information fusion," *Comput. Electron. Agric.*, vol. 158, no. October 2018, pp. 11–19, 2019, doi: 10.1016/j.compag.2019.01.039.
- [18] A. Ambarwari, Q. Jafar Adrian, and Y. Herdiyeni, "Analysis of the Effect of Data Scaling on the Performance of the Machine Learning Algorithm for Plant Identification," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 1, pp. 117–122, 2020, doi: 10.29207/resti.v4i1.1517.
- [19] D. Hong, K. Gilman, L. Balzano, and J. A. Fessler, "HePPCAT: Probabilistic PCA for Data With Heteroscedastic Noise," *IEEE Trans. Signal Process.*, vol. 69, pp. 4819–4834, 2021, doi: 10.1109/TSP.2021.3104979.
- [20] U. Radojicic, N. Lictzen, K. Nordhausen, and J. Virta, "Dimension Estimation in Two-Dimensional PCA," in *2021 12th International Symposium on Image and Signal Processing and Analysis (ISPA)*, IEEE, Sep. 2021, pp. 16–22. doi: 10.1109/ISPA52656.2021.9552114.
- [21] R. Sakti and A. Daulay, "Analisis Kritis dan Pengembangan Algoritma K-Nearest Neighbor (KNN): Sebuah Tinjauan Literatur," vol. 4, no. 2, pp. 131–141, 2024.
- [22] A. S. YUNATA, A. HALIM, and H. LUTHFIYAH, "Analisis Pengaruh Noise pada Performa K-Nearest Neighbors Algorithm dengan Variasi Jarak untuk klasifikasi Beban Listrik," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 12, no. 3, p. 745, 2024, doi: 10.26760/elkomika.v12i3.745.