Naïve Bayes-Based Freshwater Fish Classification Using HSV Color Features

Hindayati Mustafidah*[1] and Suwarsito[2]
[1]Informatics Engineering, Universitas Muhammadiyah Purwokerto, Jl. KH. Ahmad Dahlan Purwokerto, Central Java, Indonesia
[2]Aquaculture, Universitas Muhammadiyah Purwokerto, Jl. KH. Ahmad Dahlan Purwokerto, Central Java, Indonesia
E-mail: h.mustafidah@ump.ac.id, suwarsito@ump.ac.id
*Corresponding author

Keywords: freshwater fish classification, HSV color feature extraction, image processing techniques, machine learning, naïve bayes

Received: January 15, 2025

This study presents a machine learning-based method for classifying six freshwater fish species commonly consumed in Indonesia: gourami (gurame), catfish (lele), tilapia (nila), barb (melem), Java barb (tawes), and pomfret (bawal). A total of 132 images, with 22 images per species, were collected from online sources and direct field photography. The classification model utilizes a Naïve Bayes algorithm, employing color feature extraction based on the Hue, Saturation, and Value (HSV) color space. The HSV method decomposes image color into three components—Hue (the color type, such as red, blue, or green), Saturation (the intensity or vividness of the color), and Value (the brightness or lightness of the color)—allowing for improved distinction between morphologically similar species, such as barb and tilapia. Image preprocessing included resizing, background removal, and conversion from RGB to grayscale prior to HSV feature extraction. The dataset was split into training and testing subsets, with 20% of the data allocated for testing. The model's performance was evaluated using a confusion matrix, and it achieved a classification accuracy of 79.17%. This result surpasses the accuracy reported in comparable studies, such as one on frozen tuna classification, which achieved 72.73% using similar techniques. The findings validate the effectiveness of the Naïve Bayes classifier for species identification tasks in fisheries. Moreover, the approach offers a computationally efficient solution suitable for environments with constrained data availability and limited computational resources. This study underscores the practical value of machine learning in aquaculture, highlighting its potential for enhancing species monitoring, quality control, and automated recognition using relatively small datasets.

Povzetek: Opisana je uporaba HSV barvnih značilk z naivnim Bayesom, klasifikatorjem, ki omogoča bolj kvalitetno prepoznavo šestih vrst sladkovodnih rib pri majhnih podatkovnih zbirkah.

1 Introduction

Indonesia is known for its rich aquatic biodiversity, with numerous fish species found in its freshwater and marine ecosystems. Among these, several freshwater species, such as gourami (*gurami*), catfish (*lele*), tilapia (*nila*), barb (*melem*), Java barb (*tawes*), and pomfret (*bawal*), are widely cultivated and consumed. However, distinguishing between these species visually can be difficult, especially for non-experts. This difficulty is because many fish share similar physical traits, including the shape of the head, body, tail, and even their coloration. For example, barb and tilapia have nearly identical body shapes and overlapping color patterns, making it hard to tell them apart. This challenge is particularly noticeable when the fish species have similar sizes and body colors, which increases the chances of misidentification. Without the help of more advanced techniques, such as machine learning, visual identification can often lead to errors.

Previous studies have highlighted the challenges of distinguishing between these species based on visual appearance. Ref. [1] notes that tilapia exhibit a wide range of morphological variations, making it difficult to differentiate between tilapia species based solely on morphology. Similarly, [2] reported that tilapia populations from Lake Tempe and Lake Sidenreng in South Sulawesi displayed significant morphometric and meristic differences, further complicating species identification based on external features. Additionally, [3] found that Nile tilapia populations in Uganda showed distinct morphometric differences, emphasizing the difficulties in species identification, even when fish are of the same genus.

In addition to tilapia, the barb species share similar body shapes and coloration, making visual differentiation challenging. Ref. [4] discusses the morphological similarities between barb species and other fish, including tilapia, further demonstrating the need for advanced techniques to identify species in such cases accurately. This difficulty in identification highlights the importance of using machine learning-based methods, such as the Naïve Bayes classifier, to automate the identification process and reduce human error in distinguishing between visually similar fish species.

2 Related works

Various studies and applications of artificial intelligence have been conducted across diverse fields of technology development, addressing numerous real-world problems in the digital era, including the fisheries sector, such as expert systems [5]. In addition to expert systems, identification of fish objects can be done using machine learning (ML) as conducted by [6] and [7], including salinity as a factor that affects fish survival [8]. ML is software that can learn to perform tasks [9]. Identification of fish based on their images using algorithms in machine learning has been carried out by several researchers, including [10] using the Support Vector Machine (SVM) to classify fish in Bangladesh, Ref. [11] does fish freshness classification based on eye color using the K-Nearest Neighbor (K-NN). Identification of formalin fish using K-NN and GLCM was conducted by [12] and using a multilayer perceptron network by [13]. Ref. [14] identified fish types using Convolution Neural Networks (CNN). Identification of betta fish using SVM was carried out by [15] and [16] using GLCM and KNN. At the same time, [17] used the Principal Component Analysis (PCA) and K-Nearest Neighbors (KNN) algorithms, which were also conducted by [18]. The combination of KNN and PCA was carried out by [19]. Next, [20] classified the types of marine fish using the SVM method with HOG and HSV features. Using SVM, African Cichlid ornamental fish types were classified [21] and SVM and CNN were used to establish the classification models based on the LIBS and Raman spectroscopy [22]. The following summarizes several machine learning methods for fish image classification utilizing various datasets, feature extraction techniques, and classification accuracies presented in Table 1.

Table 1: Summary of several machine learning methods for fish image classification.

Study	Dataset	Classification Method	Feature Extraction	Accuracy (%)
[10]	Fish images from Bangladesh	SVM	Hybrid features	80.5
[11]	Fish freshness dataset	K-NN	Eye color	75.2
[12]	Fish images (formalin detection)	K-NN & GLCM	GLCM	71.8
[14]	Fish images	CNN	CNN-based features	85.0
[21]	African Cichlid fish images	SVM	HOG & HSV features	90.1
[23]	Frozen tuna images	Naïve Bayes	HSV features	72.7

Machine learning algorithms have been widely used for various classification tasks, including fish species identification. Techniques such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Naïve Bayes have all been explored in this domain. Among these, Naïve Bayes has gained significant attention due to its simplicity and efficiency, particularly when working with limited datasets. While more complex algorithms like CNNs and SVMs may outperform Naïve Bayes when dealing with larger datasets or more intricate models, Naïve Bayes offers a practical and efficient solution when the goal is to classify species using moderate datasets, especially in environments with resource constraints.

While methods like Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) have shown excellent results in image classification tasks (as in Table 1), their implementation often comes with considerable computational overhead. For instance, CNNs require large datasets and substantial computational resources for practical training, making them less ideal when labeled data is scarce. Furthermore, CNNs involve automatic feature extraction, which may not be optimal for simpler datasets or when the objective is to use a lightweight model suitable for real-time applications. Similarly, SVMs, though powerful, tend to struggle with small datasets and high-dimensional feature sets. They require complex kernel functions and can suffer from overfitting in such cases, in addition to demanding significant computational resources during training and prediction.

In contrast, the Naïve Bayes algorithm, which is based on a probabilistic framework, offers several advantages. It assumes feature independence, simplifies the model, and reduces computational complexity, making it especially useful when extracting features based on straightforward descriptors, like color histograms (such as HSV). Additionally, Naïve Bayes requires fewer training images, making it suitable for environments with limited labeled data. Its speed and efficiency in training and prediction are crucial in practical applications, particularly in systems requiring real-time or near-real-time classification, such as automated fish classification in aquaculture.

Ref. [24] demonstrated using Naïve Bayes for document classification, showcasing its effectiveness in assigning categories to documents based on specific attributes. Although the context was textual data, the principle of Naïve Bayes as a probabilistic classifier based on conditional independence directly applies to our fish classification task. By assuming that features (in this case, color components from the HSV color space) are independent, Naïve Bayes provides a simplified yet powerful method for classification with relatively more minor datasets, which aligns with the constraints of this study.

Multiple machine-learning models have been tested in the fish species identification domain. For instance, SVM and KNN classifiers have been extensively explored for fish classification tasks. A study by [7] applied SVM to identify fish species through acoustic signals, while [11] applied KNN to classify fish based on eye color. These models, while powerful, often require larger datasets and higher computational resources, especially in the case of deep learning approaches like CNNs. The Naïve Bayes algorithm, on the other hand, excels in environments with constrained resources and smaller datasets, providing a balance between simplicity and performance.

While CNNs, as demonstrated by [8], offer superior performance with large datasets, they are computationally expensive. The study by [24] emphasizes the computational advantages of Naïve Bayes, particularly in scenarios where training time and resource constraints are significant, making it an ideal candidate for applications such as identifying freshwater fish species in resource-limited environments.

In addition to other machine learning classification methods, the Naïve Bayes algorithm is commonly used for object image classification due to its simplicity and effectiveness, particularly in scenarios with limited data. This method assigns objects to specific classes or labels based on their attributes [25]. One key advantage of Naïve Bayes is its assumption that all attributes are independent, simplifying the model and reducing computational complexity, especially when working with smaller training datasets [26]. Given these advantages, Naïve Bayes is well-suited for classification tasks, as [27] demonstrated that it works effectively in object image classification.

The Naïve Bayes algorithm has been successfully applied in various classification tasks, such as determining the quality of tuna based on color features. For example, research by [23] achieved an accuracy of 72.727% using Naïve Bayes for this purpose. This method reinforces the method's utility in achieving reliable results, even with limited data. Given its advantages, Naïve Bayes is an ideal candidate for creating a lightweight fish classification model. Therefore, this study aims to develop a model using Naïve Bayes and HSV (Hue, Saturation, Value) color feature extraction to classify freshwater fish species based on their images. By leveraging basic color statistics, this work proposes a practical and scalable solution for automated fish identification, particularly in the fisheries sector. This research aims to provide insights into the development of a classification-based system in the fisheries sector, particularly for identifying freshwater fish species.

The primary contributions of this research are as follows:

- It introduces a simplified fish classification approach using the Naïve Bayes method, which is well-suited for resource-limited environments.
- It demonstrates the feasibility of using basic HSV color features for distinguishing visually similar freshwater fish species, offering a novel approach to fish identification.
- It presents a practical tool that could support fisheries management and monitoring of aquatic biodiversity, particularly in Indonesia.

3 Methodology

This study proposes a lightweight classification framework for identifying freshwater fish species based on image color features. Unlike many previous studies that rely on large and often publicly unavailable datasets, this work utilizes a curated image dataset consisting of 132 images across six commonly consumed freshwater fish species in Indonesia: tilapia, barb, catfish, Java barb, gourami, and pomfret. These species were selected due to their visual similarity and economic significance in Indonesian aquaculture and fish markets, where accurate classification remains challenging for practitioners and sellers.

The novelty of the dataset lies in its composition: it includes images collected not only from online sources and printed materials but also from original field data photographs taken under varied and uncontrolled conditions. This combination makes the dataset more reflective of real-world environments than many benchmark datasets used in prior studies, causing the first study to compile and utilize a field-augmented dataset of Indonesian freshwater fish species for machine learning-based image classification.

The methodological approach in this study emphasizes simplicity, efficiency, and applicability in resource-limited settings. The methodological approach adopted in this study prioritizes simplicity, computational efficiency, and practical applicability in resource-constrained environments. The research workflow comprises several key stages: data collection, data splitting, preprocessing and feature extraction. The next step is the classification process. First, the Naïve Bayes model is implemented using the training data. Once the model is trained, it is evaluated using the test data, and accuracy is assessed using a confusion matrix, which provides detailed insight into classification accuracy. The complete methodology is illustrated in Figure 1.

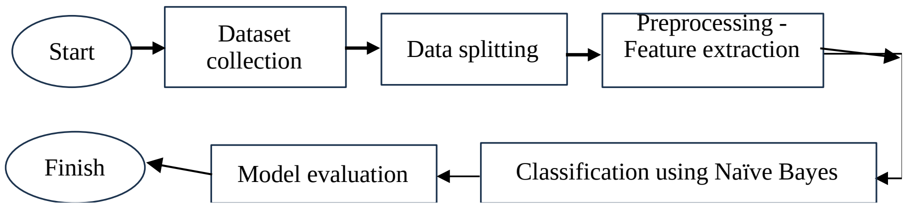

Figure 1: Research flow diagram.

3.1 Dataset collection

The images for this study were collected from three primary sources: publicly available fish images found in books and websites [28], along with original photographs captured using a digital camera. The dataset comprises 132 images representing six freshwater fish species—gourami, catfish, tilapia, barb, Java barb, and pomfret—with 22 images per species. Several examples of images are shown in Figure 2. These images vary in quality, resolution, and lighting to better reflect real-world conditions. The relatively limited dataset size stems from access constraints and practical limitations such as time and available resources. However, it is sufficient for evaluating machine learning, specifically the Naïve Bayes classifier, for fish species identification.

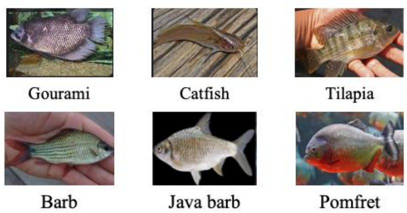

Figure 2: Image examples of freshwater fish species were used as a dataset.

Each image was assigned to one of six classes corresponding to the species, with an effort to keep the class sizes balanced. No explicit rebalancing techniques were used, as the small scale of the dataset and the simplicity of the Naïve Bayes classifier allowed for some natural variation in class representation. While larger, more diverse datasets would enable the use of more complex algorithms and likely yield better results, the current dataset provides a practical foundation for demonstrating the feasibility of this approach. Future research may seek to expand the dataset to improve model generalization and performance.

3.2 Data splitting

The dataset was separated into training and testing sets using an 80:20 ratio, resulting in 108 images for training and 24 for testing, with both sets reflecting all species equally. The training set was used to teach the model to identify freshwater fish from images, while the testing set evaluated how well the model recognized new examples. An 80:20 split was chosen over k-fold cross-validation because the dataset was small; k-fold would have produced folds with too few images, making the results unreliable and potentially unrepresentative of the species distribution. This simpler split offered a direct, computationally efficient way to check model accuracy on unseen data and reduce the chance of overfitting. For future research with more data, k-fold cross-validation could be applied to get even more reliable performance estimates.

3.3 Preprocessing and feature extraction

For image preprocessing, we standardized the dataset by resizing all images to a consistent size, removing background noise, and converting each image from the RGB color space to HSV (Hue, Saturation, Value). This RGB-to-HSV transformation was carried out using the standard mathematical conversion commonly found in libraries like OpenCV and MATLAB. Each pixel's RGB value was directly mapped to its corresponding HSV value, which separates color information (Hue), color intensity (Saturation), and brightness (Value). This consistent conversion process helped ensure uniformity across the dataset and set the stage for reliable feature extraction.

Regarding feature extraction, we focused exclusively on the HSV color information. The decision to use HSV was based on its effectiveness in handling variations in lighting and shadows—factors that often complicate fish image analysis. HSV's separation of hue, saturation, and value made it possible to reliably capture the color patterns that distinguish different fish species, which was particularly important given the dataset's small size and the classification model's simplicity. While shape and texture features can also aid in classification, we opted not to include them to avoid overfitting. Future research could integrate features like texture (via GLCM or LBP) or shape (using contours or HOG), especially with larger datasets, to improve accuracy among visually similar species except for subtle shape or texture differences.

3.4 Classification using Naïve Bayes

We selected the Naïve Bayes classifier for a few practical reasons that fit the needs and limitations of this project. For one, it does not need much training data to get up and running, which is ideal when working with smaller datasets. It is also quick to train and straightforward to code, making it a strong choice when the hardware is limited—think rural fish farms or mobile devices. Because Naïve Bayes treats features as independent and handles both categorical and continuous variables, it meshes well with the data we use here, like HSV color values and object area. Specifically, we used the Gaussian version, which assumes that features are normally distributed [29] as in Equation (1).

$$\hat{y} = \arg\max_{y\in\mathcal{Y}} P\,(y) \prod_{i=1}^{n} P(\,x_i \mid y\,) \qquad (1)$$

$\hat{y}$: The predicted class label

Y: The set of all possible classes

$P(y)$: The prior probability of class y

$x_1, x_2, \ldots, x_n$: The features or attributes of the input data X

$P(x_i|y)$: The likelihood of feature x_i given class y

More advanced models, like CNNs, often do a great job with image classification, but they generally need a lot of labeled data and heavy hardware. These heavyweight models are usually more than necessary for simple classification tasks—especially when features like color and size do most of the work. That is why, in this case, Naïve Bayes was the right starting point: efficient, effective, and a solid baseline.

3.5 Model evaluation

Model performance was measured using a confusion matrix, with accuracy as the primary metric (see Equation (2)). The terms TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative) represent the standard outcomes for classification tasks. Accuracy was chosen because it captures the share of fish images the model correctly classified out of all test cases.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (2)$$

For this study, the Naïve Bayes classifier's effectiveness was analyzed through the confusion matrix, which details how the model's predictions break down. Specifically, the matrix allows us to track:

- TP: When the model correctly identifies fish species.
- TN: When the model correctly recognizes non-target species or rejects a misclassification.
- FP: When the model mistakenly labels a non-target species as the target species.
- FN: When the model fails to identify the target species despite its presence.

These four outcomes are essential for calculating accuracy, precision, recall, and the F1-score. Equation (2) presents how these metrics are mathematically defined, drawing from TP, TN, FP, and FN values to assess classification results.

A significant advantage of this approach lies in its simplicity, which enables robust classification performance without the reliance on specialized equipment or controlled laboratory environments. Consequently, the method is highly applicable in broader contexts, including local fisheries and educational settings, particularly where access to advanced computing infrastructure is limited.

The following details regarding the input and testing data are more explicitly defined and aligned with the model evaluation process, as presented in Table 2.

Table 2: Feature types and input data for classification.

Feature Type	Feature Description	Input Data
HSV Features	The Hue, Saturation, and Value components are extracted from image pixels.	Testing data: Test set of images used to evaluate the model's classification performance.
Area Features	Area-related parameters such as fish size or object area in the image.	Testing data: Area features extracted from test set images to be compared against model predictions.

The "Input" column corresponds to the testing data used to evaluate the model. Specifically, the testing data consists of images not included in the training set and are employed to assess the model's generalization ability. The model's performance is evaluated by comparing its predictions to the test set's actual labels or ground truth. For the HSV features, the input data comprises test set images from which the Hue, Saturation, and Value components are extracted. Regarding the area features, the input data refers to area-based parameters, such as fish size or object area, which are also derived from the test set images and subsequently compared to the predicted values.

4 Result and discussion

The Matlab programming language was employed to develop the model in this study. The procedure began with preprocessing and feature extraction, then implemented the Naïve Bayes algorithm and the subsequent model evaluation. The Naïve Bayes model uses the training data to implement the Naïve Bayes algorithm for the classification process. The model's accuracy is evaluated using the test data after training, and a confusion matrix is employed.

4.1 Preprocessing – feature extraction

The dataset employed in this study consists of 132 images divided into 108 training images and 24 testing images, following an 80:20 split. Color feature extraction is performed before classifying the freshwater fish images using the Naïve Bayes algorithm, as depicted in Figure 3.

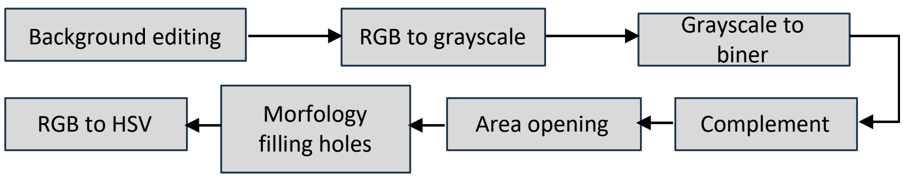

Figure 3: Stages carried out in preprocessing.

The fish images used in this study exhibit considerable size and background composition variation. Manual preprocessing was applied to ensure uniformity and prevent distortion during subsequent processing stages. This process involved two main steps: background removal and image resizing. Each image was cropped to a standardized dimension of 788 × 788 pixels, and the background was removed to isolate the fish object.

The primary purpose of this preprocessing step was to enhance image quality and ensure consistency across the dataset, facilitating more accurate and efficient feature extraction. Standardizing these visual characteristics contributes to the reliability and performance of the classification process.

As illustrated in Figure 4, the initial classification performance was evaluated. An example of a raw fish image is presented in Figure 4a, while its corresponding preprocessed version is shown in Figure 4b. Subsequent refinements, as depicted in Figure 5 and Figure 6, led to notable improvements in performance.

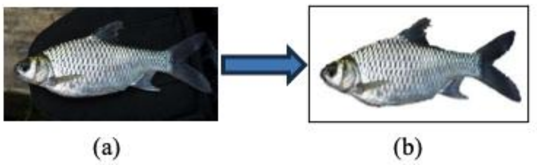

Figure 4: Image of Java barb fish: (a) before editing; (b) after editing.

Following the preprocessing procedures described in Figure 3, the next step involves converting RGB (Red, Green, Blue) images into grayscale. This conversion is performed by averaging the RGB components to produce a single intensity value for each pixel, generating a grayscale image. An example of this transformation is presented in Figure 5a.

The grayscale image is then further processed into a binary (black-and-white) image, as shown in Figure 5b. This binarization reduces image complexity by limiting pixel values to two categories, simplifying the pattern recognition process and enhancing the detectability of relevant features.

A complement operation is applied to the binary image to improve image contrast further. This operation inverts the pixel values by subtracting each from 255, effectively switching object and background representation. As a result, the object is denoted by a pixel value of 1, while the background is represented by 0. This transformation benefits subsequent image-processing steps such as morphological operations and feature extraction. The outcome of the complement operation is illustrated in Figure 5c.

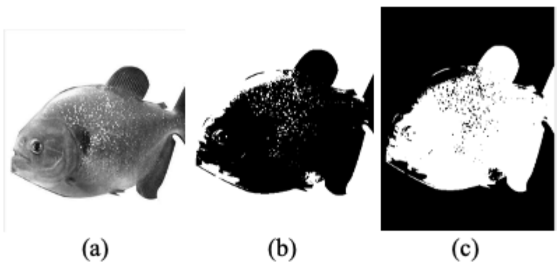

Figure 5: Results of the color conversion process of freshwater fish images: (a) grayscale image; (b) binary image; (c) complement.

The next step involves applying an area-opening operation to the image to remove small, irrelevant objects that may distort the image. This morphological operation helps preserve the integrity of the image. Following this, morphological closing is performed to fill small gaps or holes in the image. Figure 6 presents the image before and after the morphological process. Small objects are removed, and gaps previously with a value of 0 are closed with a value of 1.

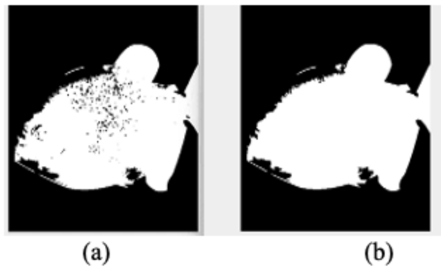

Figure 6: Results of the morphological process: (a) Image after area opening process; (b) image after morphology filling holes process.

The final preprocessing stage involves converting the RGB image to the HSV (Hue, Saturation, Value) color space. This conversion enables the extraction of information related to the image's color, brightness, and purity. The HSV color space is advantageous, as it separates these components, making it easier to detect fish objects even in slightly dark images, as the color purity and light intensity do not significantly affect detection. The conversion from RGB to HSV for the fish image is shown in Figure 7.

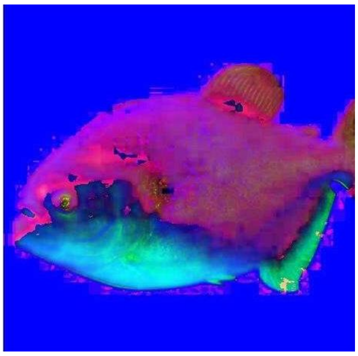

Figure 7: Results of RGB to HSV image conversion.

This study uses the HSV color space for feature extraction to distinguish between the different fish species. The HSV model separates an image's color into three components: Hue, Saturation, and Value. Each component plays an essential role in representing the color characteristics of the fish.

- Hue (H): This represents the color type of the image, such as red, green, or blue. The Hue value is typically measured on a scale from 0 to 360 degrees, where each degree corresponds to a different color. For instance, 0°corresponds to red, 120° corresponds to green, and 240° corresponds to blue.

- Saturation (S): This component indicates the intensity or vividness of the color. It ranges from 0 to 1, in which zero means the color is a shade of gray (i.e., no saturation), and one means the color is fully saturated, appearing as its purest form. Saturation helps determine how vibrant or dull a color appears.
- Value (V): This represents the brightness or lightness of the color, ranging from 0 to 1, where 0 corresponds to black (no brightness), and 1 corresponds to the brightest possible color. The Value component determines the color's light or dark in the image.

Feature extraction in this study focuses on extracting the HSV color features and the object area. At this stage, the Hue (H), Saturation (S), and Value (V) components, along with the area of the object in the image, are extracted. The background pixels in the H, S, and V components are set to 0 to ensure that only object pixels are processed in the subsequent steps, specifically in calculating the average values. Additionally, the average area of the object is calculated. Table 3 and Table 4 provide examples of color feature extraction and object area calculation results.

Table 3: Results of color feature extraction and area of freshwater fish images on training data.

No.	**Hue**	**Saturation**	**Value**	**Area**
1	0.403318323419042	0.260346695106469	0.536819084229093	84286
2	0.335057730985058	0.152018628974081	0.394937026807920	71350
3	0.270479850906780	0.258428038903359	0.331773459548743	74277
4	0.330772366940356	0.435419654272074	0.318038088329406	98443
5	0.589932605894102	0.158395421440156	0.507638669354549	62707
...				
108	0.124554169492385	0.251562245511451	0.403377826856695	141056

Table 4: Results of color feature extraction and area of freshwater fish images on testing data.

No.	**Hue**	**Saturation**	**Value**	**Area**
1	0.500006801952211	0.280613369049232	0.662378360420152	60908
2	0.372640832644990	0.220835127629331	0.380358140777996	139611
3	0.170120032846037	0.315069446363992	0.434321305868149	192706
4	0.433518886610450	0.262447886902986	0.349181193584367	107974
5	0.117228933141659	0.223715743829939	0.460529587190712	196720
...				
24	0.320989767431954	0.273306776067016	0.497645896540657	179498

4.2 Naïve Bayes algorithm implementation

In this study, we applied the Naïve Bayes algorithm to classify freshwater fish species using visual data, focusing on the HSV color feature extraction method. The stages of implementing the Naïve Bayes algorithm are as follows:

- Read the train data
- Counting the number of classes/labels
- Calculate the probability value of the hypothesis using Equation (1).

Table 5 displays the output of the Naïve Bayes algorithm's prediction results in solutions.

In the HSV conversion process, the function "$value_{ofColor}$" refers to the Value component in the HSV model, which is derived from the RGB values of each pixel in the image. The term "value_ofColor" is used to indicate the brightness of the pixel's color, which is an important factor in differentiating between various fish species with similar Hue and Saturation characteristics. The value categories referenced in the computations of Table 5 are derived from the data shown in Table 3 and Table 4, specifically:

Table 5: Illustration of computational results using the Naïve Bayes algorithm.

No.	**Hue**	**Saturation**	**Value**	**Area**	**Class**
1	2	2	2	1	pomfret
2	2	1	2	2	pomfret
3	1	2	2	2	pomfret
4	2	2	1	2	pomfret
5	1	1	2	2	gourami
6	1	1	2	2	gourami
7	2	2	2	2	gourami
8	2	2	2	2	gourami

$$\frac{Hue}{Saturation} = \begin{cases} 1;\ if\ value_{ofColor} \leq 0.2 \\ 2;\ if\ 0.2 < value_{ofColor} \leq 0.6 \end{cases}$$

$$Value = \begin{cases} 1;\ if\ value_{ofColor} \leq 0.34 \\ 2;\ if\ 0.34 < value_{ofColor} \leq 0.6 \end{cases}$$

As explained previously, the Naïve Bayes (NB) algorithm is implemented using the MATLAB programming language in this case. Figure 8 shows an example of a program code snippet that utilizes the "fitcnb" function to perform classification using the Gaussian Naïve Bayes (GNB) algorithm with parameters in the form of "ciri_latih" and "kelas_latih."

```
% klasifikasi citra menggunakan Naive Bayes
Mdl = fitcnb(ciri_latih,kelas_latih);
```

Figure 8: The “fitcnb” function for training the fish image dataset.

In the program snippet that invokes the "fitncb" function, the GNB Algorithm carries out the data training process. This process generates a prediction model for freshwater fish species based on all the input data, specifically the variables "ciri_latih" and "kelas_latih," using the Bayes probability concept as described in Equation (1). The output of this function is stored in the variable "Mdl."

Once the training process recognizes the pattern, the resulting model undergoes evaluation. This evaluation aims to assess the model's performance, particularly its accuracy in classifying freshwater fish species based on their images. A program excerpt demonstrating the model evaluation process is presented in Figure 9, where the "predict" function is utilized.

```
% membaca kelas keluaran hasil pengujuan
hasil_uji = predict(Mdl,ciri_uji);
```

Figure 9: Program code snippet for evaluating the model.

The "Mdl" variable stores the training process results, while the "ciri_uji" variable contains the testing data along with the extracted features. The "predict" function is employed to make predictions on the testing data using the model that was previously trained. The resulting predictions are stored in the "hasil_uji" variable, with the corresponding data presented in Table 6.

Table 6: Prediction results for freshwater fish species based on 24 testing data samples.

No.	Hue	Saturation	Value	Area	Input	Output
1	0.500006801952211	0.280613369049232	0.662378360420152	60908	pomfret	pomfret
2	0.372640832644990	0.220835127629331	0.380358140777996	139611	pomfret	pomfret
3	0.170120032846037	0.315069446363992	0.434321305868149	192706	pomfret	gourami
4	0.433518886610450	0.262447886902986	0.349181193584367	107974	pomfret	pomfret
5	0.117228933141659	0.223715743829939	0.460529587190712	196720	gourami	gourami
6	0.116742908634686	0.223105665366549	0.460431169214331	196720	gourami	gourami
7	0.320989767431954	0.273306776067016	0.497645896540657	179498	gourami	tilapia
8	0.320782153662419	0.271708458927080	0.497442417724275	179507	gourami	tilapia
9	0.229210511201839	0.0964511843275545	0.349412906160667	23362	catfish	catfish
10	0.106550890144485	0.158231246737144	0.371697006596770	21062	catfish	catfish
...						
24	0.222269932465241	0.120431023603942	0.606964743707740	108652	Java barb	barb

As indicated in Equation (2), the data presented in Table 6 must be evaluated using a measurement model represented by a confusion matrix, as shown in Table 7. This confusion matrix reflects the results obtained from Equation (2) implementation, capturing the relationship between predicted and actual classifications. A detailed matrix analysis reveals the fish species that were most frequently misclassified, thereby providing valuable insights into the model's performance and highlighting its strengths and limitations.

Table 7: Model performance evaluation results using the confusion matrix.

Predicted\Actual	Pomfret	Gourami	Catfish	Barb	Tilapia	Java barb
Pomfret	3	1	0	0	0	0
Gourami	0	2	0	0	2	0
Catfish	0	0	4	0	0	0
Barb	0	0	0	4	0	0
Tilapia	0	0	0	0	4	0
Java barb	0	0	0	2	0	2

Based on Table 7, the values for TP, FP, TN, and FN can be determined, allowing for the calculation of the accuracy, as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{19}{24} = 0.79167 \approx 79.17\%$$

A detailed examination of the classification outcomes indicates that the most frequent misclassifications occurred between the Java barb and barb species. These species exhibit similar color patterns, likely leading to confusion during classification. Notably, the Java barb was misclassified as a barb on two occasions, suggesting that the HSV (Hue, Saturation, Value) color features used in the model could not capture the subtle visual distinctions between these two species. A similar trend was observed in the misclassification of gourami as tilapia, which also share comparable color tones. These findings emphasize the limitations of relying solely on color-based features, particularly HSV, which are sensitive to lighting variations and often fail to differentiate objects with closely related visual characteristics.

Despite these misclassifications, the Naïve Bayes model successfully distinguished species such as catfish, barb, and tilapia, which possess more distinct and contrasting color features. However, the model's performance declined when classifying species with overlapping attributes, such as body shape and color hue—evident in the confusion between Java barb and tilapia. Such challenges are common in image-based classification tasks, where visual similarity across classes can significantly hinder accuracy. Integrating additional feature descriptors—particularly those based on texture and shape—is recommended to address these limitations. These enhancements would provide more discriminative power and improve the model's robustness in accurately identifying fish species with subtle inter-class differences.

In this study, HSV color features were primarily utilized for classification as a feature engineering aspect. However, it is essential to investigate whether other feature extraction techniques could enhance the model's performance, particularly for species with similar visual characteristics, such as the Java barb and barb.

Several alternative feature extraction methods could complement the HSV color features to improve classification accuracy. These methods include:

- **Gabor Filters**: Gabor filters are effective for texture-based feature extraction and are commonly used in image processing due to their ability to capture local spatial frequency patterns. Gabor filters could be particularly beneficial in distinguishing between species that exhibit subtle textural differences, which may not be fully captured by color features alone [30].
- **Histogram of Oriented Gradients (HOG)**: HOG is a feature descriptor designed to capture object shapes and patterns by analyzing the distribution of gradient orientations. This method is frequently employed in object detection tasks and could improve classification performance for fish species that differ in shape but share similar color patterns [31].
- **Gray-Level Co-occurrence Matrix (GLCM)**: GLCM captures texture information by analyzing the spatial relationships between pixel values. By incorporating GLCM features, the model could become more robust in classifying species with similar appearances but distinct textures [32].

A comparative analysis could be conducted to evaluate the effectiveness of these methods by extracting features using these techniques and comparing the classification performance with the HSV-based approach. Including these additional features would provide complementary information, especially for fish species that share similar color patterns but differ in texture or shape. Implementing multiple feature extraction techniques would be a natural progression of this study and could yield improved results, particularly for misclassified species like the Java barb and barb.

With an accuracy of 79.17%, the model demonstrates promising potential for practical applications in fish species identification. However, to emphasize the novelty of our approach and its improvements over existing techniques, we now compare our method with several other machine learning approaches commonly employed in fish classification tasks.

To validate the performance of the Naïve Bayes model, statistical significance tests were conducted to determine whether the accuracy of 79.17% significantly differed from that of other classifiers. A t-test was performed to compare the accuracy of the Naïve Bayes model with that of two alternative classifiers: Support Vector Machine (SVM) and K-Nearest Neighbors (KNN). These classifiers were chosen due to their frequent use in classification tasks and distinct model construction approaches.

The model was executed multiple times (five repetitions) to account for any potential variation in performance, with the accuracy averaged across these runs. The t-test results indicated that, while the Naïve Bayes model performed adequately, the difference in accuracy between it and the SVM or KNN classifiers was not statistically significant ($p > 0.05$). This finding suggests that, although more complex models like SVM and KNN may show slightly superior performance, the Naïve Bayes algorithm offers competitive accuracy with a significantly lower computational cost and more straightforward implementation.

4.3 Comparison with previous studies

The Naïve Bayes model in this study achieved an accuracy of 79.17% in classifying six freshwater fish species, surpassing the 72.73% accuracy reported by [23], who used Naïve Bayes and HSV features to classify frozen tuna quality. Although the datasets and tasks differed, this comparison highlights the robustness of the Naïve Bayes algorithm across various image-based classification contexts. Given the diverse nature of the dataset—comprising images from the internet, books, and camera captures—there is potential for further optimization in feature extraction, as demonstrated by [33].

To provide context, we compared the Naïve Bayes model with other classifiers, such as Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), using the same dataset and preprocessing steps. As shown in Table 8, Naïve Bayes achieved 79.17% accuracy, while SVM and KNN achieved 82.00% and 80.48%, respectively. Although SVM slightly outperformed the others, it has a significantly higher computational cost, especially regarding training time and memory usage. KNN also performed well but struggles with high-dimensional data, making it less efficient for more complex tasks. Naïve Bayes, with its competitive performance and more straightforward implementation, proves particularly suitable for real-time classification tasks, where speed and resource limitations are crucial.

Table 8: Performance comparison for Naïve Bayes, SVM, and KNN classifiers.

Classifier	**Accuracy**	**Feature Extraction Methods**
Naïve Bayes	79.17%	HSV color features
SVM	82.00%	HOG + HSV
KNN	80.48%	GLCM + HSV

Several studies have applied machine learning algorithms for fish classification. For instance, [10] used SVM to classify indigenous fish species in Bangladesh, achieving an accuracy of 80%, which is similar to our results. However, SVM is computationally intensive and may not perform efficiently with smaller datasets. In contrast, Naïve Bayes offers comparable accuracy with a more straightforward and less computationally demanding approach, making it more accessible for practical applications involving limited data.

The K-Nearest Neighbors (KNN) algorithm has also been used in fish freshness classification based on eye color, as shown by [34]. While KNN achieved good accuracy, it is sensitive to the curse of dimensionality, especially when multiple features are involved. Naïve Bayes, assuming feature independence, is more robust in cases where features are not strongly correlated, such as in color-based classification tasks.

Convolutional Neural Networks (CNNs) have shown excellent performance in image classification, as demonstrated by [35], who applied CNNs for fish classification, achieving an accuracy of 93%. While CNNs perform well with large datasets and complex feature extraction, they require substantial labeled data. They are computationally expensive, making them less suitable for smaller datasets or real-time applications. In comparison, Naïve Bayes, although less complex, remains a competitive solution for smaller datasets with efficient implementation.

To summarize, while more complex models such as CNNs outperform Naïve Bayes' accuracy, they come with significantly higher computational costs. Our Naïve Bayes model, which uses HSV color features, provides a more straightforward, more efficient solution with competitive results, especially for resource-constrained applications. Additionally, incorporating more advanced feature extraction techniques, such as Histogram of Oriented Gradients (HOG) or Gray-Level Co-occurrence Matrix (GLCM)—used in SVM and KNN classifiers—could further enhance Naïve Bayes performance by capturing shape and texture information that HSV alone may not fully address.

4.4 Limitations of this study

4.4.1 Model implementation: Hyperparameter selection and tuning

While the Naïve Bayes classifier was implemented using the fitcnb function in MATLAB, the manuscript currently lacks detailed information regarding the selection of hyperparameters for the model. Hyperparameter tuning is crucial in optimizing machine learning models for optimal performance. In this study, we did not explicitly perform hyperparameter tuning, as Naïve Bayes is known to perform effectively with its default settings. However, hyperparameter tuning is typically critical to achieving high performance for more complex models such as Support Vector Machine (SVM) or K-Nearest Neighbors (KNN) [36].

For Naïve Bayes, the primary hyperparameter is the distribution type used for the features. The "fitcnb" function in MATLAB offers several options for selecting the distribution type, including normal distribution (default), kernel distribution, or multinomial distribution [37].

Cross-validation could be employed to evaluate the model's accuracy under different settings and assess whether these distribution types could improve classification performance. This method would allow for identifying the distribution type that best suits the feature set, potentially enhancing classification results [38].

Furthermore, future studies could apply advanced techniques such as Bayesian optimization or grid search for automated hyperparameter tuning. These methods have effectively optimized hyperparameters, improving model performance. Implementing these tuning methods would be particularly beneficial if additional feature extraction techniques are incorporated. Different features may require distinct distribution models to capture their underlying characteristics effectively, thus further improving the model's performance [36].

4.4.2 Data augmentation

Given the relatively small dataset size of 132 images, data augmentation is essential to improve the model's generalization and robustness. In the current study, no data augmentation was applied to the dataset, which may have limited the model's ability to generalize to unseen data. Data augmentation can significantly expand the diversity of the dataset, particularly in situations where collecting more labeled data is challenging [39].

To address this limitation, future work should experiment with standard augmentation techniques such as:

- **Rotation**: Rotating images by small degrees can simulate orientation changes, ensuring the model can recognize fish species from various angles [40].
- **Scaling**: Scaling images allows the model to learn scale-invariant features, which could help recognize varying-sized fish species [40].
- **Color Jittering**: Modifying the color saturation, contrast, and brightness can help the model become more robust to changes in lighting conditions, which is especially useful when images are taken in different environments [41], [42].

By incorporating these techniques, the model's performance could be improved, particularly in real-world applications where images vary in orientation, size, or lighting. Data augmentation enhances the model's robustness and reduces the risk of overfitting, thereby improving its generalization capabilities [39].

4.4.3 Robustness evaluation

An essential aspect of the study is how well the model performs under real-world conditions. In this experiment, the dataset used for training and testing was limited to images with controlled lighting and backgrounds. However, real-world images of fish may exhibit significant lighting, background, and noise variations, affecting the model’s ability to make accurate predictions [43].

To assess the robustness of the Naïve Bayes model, future studies should evaluate its performance on external datasets collected under diverse conditions, including:

- **Varying Lighting**: Fish images taken in different lighting conditions can cause color variations, which could challenge the model. The inclusion of color jittering during training, as mentioned in the previous section, could mitigate this issue [40].
- **Different Backgrounds**: Images captured against cluttered or inconsistent backgrounds can introduce noise that affects the model’s performance. Preprocessing techniques like background removal or segmentation could help these cases [44].

Additionally, the model should be tested on real-world images collected from different sources to validate its ability to generalize across diverse environments. This external validation is crucial for ensuring the model performs well in practical applications, such as automated fish identification in aquaculture systems [43].

4.4.4 Limitations of the Naïve Bayes approach

While the Naïve Bayes algorithm has demonstrated reasonable performance in this study, it has several limitations compared to more complex models such as Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs). One of the primary limitations of Naïve Bayes is its assumption of feature independence, which, although beneficial for simplifying the model, is often unrealistic in real-world data where features may be correlated [45]. For instance, in fish images, attributes such as color, shape, and texture are typically interdependent, yet Naïve Bayes treats them as independent, which can lead to suboptimal performance [46].

Furthermore, Naïve Bayes does not automatically learn the optimal features from raw image data. In contrast, CNNs excel in automatically extracting hierarchical features from images, which often results in higher classification performance, especially when the dataset is large and complex [47]. While Naïve Bayes can handle smaller datasets and is computationally efficient, its performance may be limited in more challenging classification tasks, particularly those involving high-dimensional image data with intricate details.

Another limitation is relying on manually selected features, such as HSV values, rather than learning features from the data. This learning is a significant advantage of deep learning models like CNNs, which can automatically learn features directly from raw data [47]. This reliance on predefined features restricts Naïve Bayes' ability to generalize to unseen data or more complex image conditions, a critical aspect of modern image classification tasks.

4.4.5 Computational efficiency

This study chose the Naïve Bayes classifier primarily for its computational efficiency, making it suitable for real-time applications and environments with limited resources. Compared to more complex classifiers like Support Vector Machines (SVM) and Convolutional Neural Networks (CNNs), Naïve Bayes requires significantly less computational power and training time. For instance, a study by [48] reported that Naïve Bayes had a training time of 0.35 seconds and an inference time of 0.01 seconds. In contrast, SVM took 15.62 seconds for training and 0.02 seconds for inference. These results underscore the trade-off between model complexity and computational efficiency, highlighting Naïve Bayes as an attractive option for applications where speed and simplicity are prioritized.

Further supporting this, research by [49] compared Naïve Bayes, SVM, and K-Nearest Neighbors (KNN) in sentiment analysis of public opinion regarding COVID-19 vaccination on Twitter. They found that Naïve Bayes achieved an accuracy of 94%, with faster training and inference times than SVM and KNN.SVM achieved the highest accuracy at 96.3% but with longer training and inference times. KNN achieved an accuracy of 91%, with inference times faster than SVM but slower than Naïve Bayes. This study utilized a Twitter dataset of 35,644 tweets and applied TF-IDF feature extraction and TextBlob for labeling.

4.5 Recommendations and future works

To enhance the performance and applicability of the Naïve Bayes model in fish species classification, several avenues for future research are proposed:

- **Integration of advanced feature extraction techniques**: Combining feature extraction methods, such as Gabor filters, Histogram of Oriented Gradients (HOG), and Gray Level Co-occurrence Matrix (GLCM), can improve the model's ability to distinguish visually similar species. For instance, Gabor filters have effectively identified goldfish species [50] while HOG features have been applied in fish freshness detection [51].
- **Hyperparameter tuning**: Utilizing techniques such as cross-validation or Bayesian optimization can help select the optimal distribution type for the features, thereby improving the performance of the Naïve Bayes model [52].
- **Data augmentation**: Implementing data augmentation techniques, including rotation, scaling, and color jittering, can artificially expand the dataset, improving model generalization and reducing overfitting, especially when working with relatively small datasets [53].
- **Utilization of K-Fold Cross-Validation**: Applying k-fold cross-validation instead of a simple train-test split can provide more robust performance estimates and ensure the model does not overfit the training data [54].
- **Development of hybrid models**: Combining Naïve Bayes with other machine learning algorithms, such as K-Nearest Neighbors (KNN) or Support Vector Machines (SVM), can leverage the strengths of different models, potentially improving classification accuracy [55].
- **Robustness testing**: Testing the model on external datasets under diverse real-world conditions, such as varying lighting and backgrounds, can assess its robustness and ensure reliable performance in real-world applications [56].
- **Enhancement of the dataset**: Collecting a larger and more diverse dataset, particularly under varying lighting conditions, can help the model perform better across different environmental settings and reduce misclassifications caused by subtle image variations [39].

4.6 Novelty and improvements

Although this study has limitations, one key improvement in our approach is the careful preprocessing and feature extraction strategy. By converting RGB images to HSV color space, we separate the image's color, saturation, and brightness components, which enhances the classification model's ability to identify fish species under varying lighting conditions. This step is crucial in the context of fish species that share similar morphological features, which is a challenge addressed by our method.

Moreover, our model leverages a relatively small dataset of 132 images, smaller than datasets typically used in deep learning approaches like CNNs. Despite this, our Naïve Bayes classifier achieved a competitive accuracy, showcasing its potential for efficient classification in scenarios where large datasets are unavailable. This aspect of our approach is particularly beneficial for resource-constrained environments where data collection may be limited.

Finally, our study contributes to the field by demonstrating that the Naïve Bayes classifier can be effectively applied in fish identification, even with the relatively simple HSV feature extraction method. This finding contrasts with other more complex algorithms, making Naïve Bayes an attractive choice for practitioners seeking a balance between simplicity, speed, and accuracy.

5 Conclusion

In conclusion, the Naïve Bayes classifier demonstrated competitive performance for fish species classification, achieving an accuracy of 79.17%. While slightly outperformed by Support Vector Machine (SVM), the model's simplicity, computational efficiency, and suitability for smaller datasets make it a practical choice for real-world applications in the field. This study underscores the applicability of the Naïve Bayes algorithm for automated fish species identification using image data. Despite its satisfactory performance, future improvements can be made by incorporating additional feature extraction techniques, optimizing hyperparameters, and exploring more complex models, such as hybrid approaches and Convolutional Neural Networks (CNNs). Expanding the dataset size and employing data augmentation techniques would also enhance classification accuracy and robustness, allowing the model to differentiate visually similar species better. Although more advanced models may demand higher computational resources, these advancements could significantly increase the model's accuracy, making it a more effective and reliable solution for fish classification in practical applications.

Acknowledgments

The author would like to thank the Universitas Muhammadiyah Purwokerto for supporting and providing funds to implement this research through the Research and Community Service Board with contract number A.11-III/6939-S.Pj./LPPM/II/2024.

References

[1] E. Trewavas, *Tilapiine fishes of the genera Sarotherodon, Oreochromis, and Danakilia*. 1983.

[2] S. Juliani, "Karakteristik Morfometrik dan Meristik Ikan Nila, Oreochromis niloticus (Linnaeus 1758) di Danau Tempe, Kabupaten Wajo dan Danau Sidenreng, Kabupaten Sidenreng Rappang Provinsi Sulawesi Selatan," Universitas Hasanuddin, 2023.

[3] G. Kwikiriza, M. J. Yegon, N. Byamugisha, A. Beingana, F. Atukwatse, A. Barekye, J. K. Nattabi, and H. Meimberg, "Morphometric variations of Nile tilapia (Oreochromis niloticus) (Linnaeus, 1758) local strains collected from different fish farms in South Western Highland Agro-Ecological Zone (SWHAEZ), Uganda: screening strains for aquaculture," *Fishes*, vol. 8, no. 4, p. 217, 2023, https://doi.org/10.3390/fishes8040217.

[4] Y. Hadjidji, "HUBUNGAN KEKERABATAN BEBERAPA JENIS IKAN (Pisces) AIR TAWAR BERDASARKAN KARAKTER MORFOLOGI DAN PEMANFAATANNYA SEBAGAI MEDIA PEMBELAJARAN," 2019, *Universitas Tadulako, Palu, Sulawesi Tengah*.

[5] H. Mustafidah, B. R. Alfiansyah, and N. Hidayat, "Expert System Using Forward Chaining to Determine Freshwater Fish Types Based on Water Quality and Area Conditions," in *2023 Eighth International Conference on Informatics and Computing (ICIC)*, IEEE, 2023, pp. 1–5.

[6] S. Zhao, S. Zhang, J. Liu, H. Wang, J. Zhu, D. Li, and R. Zhao, "Application of machine learning in intelligent fish aquaculture: A review," *Aquaculture*, vol. 540, p. 736724, 2021, https://doi.org/10.1016/j.aquaculture.2021.736724.

[7] A. Yassir, S. J. Andaloussi, O. Ouchetto, K. Mamza, and M. Serghini, "Acoustic fish species identification using deep learning and machine learning algorithms: A systematic review," *Fish Res*, vol. 266, p. 106790, 2023, https://doi.org/10.1016/j.fishres.2023.106790.

[8] W. Rahman *et al.*, "A Deep CNN-Based Salinity and Freshwater Fish Identification and Classification Using Deep Learning and Machine Learning," *Sustainability*, vol. 16, no. 18, p. 7933, 2024, https://doi.org/10.3390/su16187933.

[9] T. P. Trappenberg, *Fundamentals of Machine Learning*. Oxford University Press, 2020.

[10] M. A. Islam, M. R. Howlader, U. Habiba, R. H. Faisal, and M. M. Rahman, "Indigenous fish classification of Bangladesh using hybrid features with SVM classifier," in *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, IEEE, 2019, pp. 1–4. 10.1109/IC4ME247184.2019.9036679.

[11] Y. Krisdiani, "Klasifikasi kesegaran ikan berdasarkan warna mata menggunakan metode K-Nearest Neighbor," 2019, *Universitas Negeri Malang*.

[12] A. Pariyandani, D. A. Larasati, E. P. Wanti, and M. Muhathir, "Klasifikasi Citra Ikan Berformalin Menggunakan Metode k-NN dan GLCM," in *Semantika (Seminar Nasional Teknik Informatika)*, 2019, pp. 42–47. [Online]. Available: https://semantika.polgan.ac.id/index.php/Semantika/article/view/49

[13] E. P. Wanti and M. Muhathir, "Pengidentifikasian Citra Ikan Berformalin Dengan Menggunakan Metode Multilayer Perceptron," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 5, no. 1, pp. 491–502, 2021, http://dx.doi.org/10.30645/j-sakti.v5i1.342.

[14] A. Azis, "IDENTIFIKASI JENIS IKAN MENGGUNAKAN MODEL HYBRID DEEP LEARNING DAN ALGORITMA

KLASIFIKASI," *Sebatik*, vol. 24, no. 2, pp. 201–206, 2020, https://doi.org/10.46984/sebatik.v24i2.1057.

[15] I. R. Diesta and W. F. Al Maki, "Klasifikasi Ikan Cupang Menggunakan Support Vector Machine," in *eProceedings of Engineering*, 2021, pp. 10556–10565.

[16] M. A. D. Akbar, A. B. Setiawan, and R. K. Niswatin, "Klasifikasi Jenis Ikan Cupang Menggunakan Metode GLCM Dan KNN," in *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 2021, pp. 152–158.

[17] I. E. Hasym and I. Susilawati, "Klasifikasi Jenis Ikan Cupang Menggunakan Algoritma Principal Component Analysis (PCA) Dan K-Nearest Neighbors (KNN)," *KONSTELASI: Konvergensi Teknologi dan Sistem Informasi*, vol. 1, no. 1, pp. 168–179, 2021, https://doi.org/10.24002/konstelasi.v1i1.4242.

[18] S. Suwarsito, H. Mustafidah, T. Pinandita, and P. Purnomo, "Freshwater Fish Classification Based on Image Representation Using K-Nearest Neighbor Method," *JUITA: Jurnal Informatika*, vol. 10, no. 2, pp. 183–189, 2022, http://dx.doi.org/10.30595/juita.v10i2.15471.

[19] R. Nuraini, A. Wibowo, B. Warsito, W. A. Syafei, and I. Jaya, "Combination of K-NN and PCA Algorithms on Image Classification of Fish Species," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 5, pp. 1026–1032, 2023, https://doi.org/10.29207/resti.v7i5.5178.

[20] N. Rachmat, Y. Yohannes, and A. Mahendra, "Klasifikasi Jenis Ikan Laut Menggunakan Metode SVM dengan Fitur HOG dan HSV," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 8, no. 4, pp. 2235–2247, 2021, https://doi.org/10.35957/jatisi.v8i4.1686.

[21] D. Yusup, S. Faisal, and A. Pratama, "Klasifikasi Jenis Ikan Hias African Cichlid Menggunakan Algoritma Support Vector Machines," *Scientific Student Journal for Information, Technology and Science*, vol. 4, no. 1, pp. 31–38, 2023, http://repository.ubpkarawang.ac.id/id/eprint/2909/

[22] L. Ren *et al.*, "Rapid identification of fish species by laser-induced breakdown spectroscopy and Raman spectroscopy coupled with machine learning methods," *Food Chem*, vol. 400, p. 134043, 2023, https://doi.org/10.1016/j.foodchem.2022.134043.

[23] F. A. Susilo, H. Fitriyah, and G. E. Setyawan, "Sistem Klasifikasi Kualitas Ikan Tongkol Beku Berdasarkan Fitur Nilai Warna HSV Menggunakan Metode Naïve Bayes," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 1, pp. 753–760, 2019, Accessed: Jul. 02, 2025. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/4218

[24] R. Wang, "Automatic classification of document resources based on Naive Bayesian classification algorithm," *Informatica*, vol. 46, no. 3, 2022, https://doi.org/10.31449/inf.v46i3.3970.

[25] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan normalisasi data untuk klasifikasi wine menggunakan algoritma K-NN," *CESS (Journal of Computer Engineering, System and Science)*, vol. 4, no. 1, pp. 78–82, 2019, https://doi.org/10.24114/cess.v4i1.11458.

[26] N. M. S. Iswari, W. Wella, and R. Ranny, "Perbandingan Algoritma kNN, C4. 5, dan Naive Bayes dalam Pengklasifikasian Kesegaran Ikan Menggunakan Media Foto," *Ultimatics: Jurnal Teknik Informatika*, vol. 9, no. 2, pp. 114–117, 2017, https://doi.org/10.31937/ti.v9i2.659.

[27] E. Ellif, S. H. Sitorus, and R. Hidayati, "Klasifikasi Kematangan Pepaya Menggunakan Ruang Warna HSV dan Metode Naive Bayes Classifier," *Coding Jurnal Komputer dan Aplikasi*, vol. 9, no. 01, pp. 66–75, 2021, http://dx.doi.org/10.26418/coding.v9i01.45906.

[28] ResepKoki, "10 Jenis Ikan Air Tawar di Indonesia Yang Sering Dikonsumsi," https://resepkoki.id/10-jenis-ikan-air-tawar-di-indonesia-yang-sering-dikonsumsi/. Accessed: Nov. 07, 2021. [Online]. Available: https://resepkoki.id/10-jenis-ikan-air-tawar-di-indonesia-yang-sering-dikonsumsi/

[29] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[30] L. Zhu, T. Chen, J. Yin, S. See, and J. Liu, "Learning gabor texture features for fine-grained recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 1621–1631. Accessed: Jul. 15, 2025. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Zhu_Learning_Gabor_Texture_Features_for_Fine-Grained_Recognition_ICCV_2023_paper.html

[31] G. Ajmera, "HOG (Histogram of Oriented Gradients): An Amazing Feature Extraction Engine for Medical Images," https://medium.com/@girishajmera/hog-histogram-of-oriented-gradients-an-amazing-feature-extraction-engine-for-medical-images-5a2203b47ccd.

[32] F. Xiao, L. Kaiyuan, W. Qi, Z. Yao, and Z. Xi, "Texture analysis based on gray level co-occurrence matrix and its application in fault detection," in *International Geophysical Conference, Beijing, China, 24-27 April 2018*, Society of Exploration

Geophysicists and Chinese Petroleum Society, 2018, pp. 836–839.

[33] A. Priyambodo and P. Prihati, "Klasifikasi Gambar Aksara Jawa Menggunakan Optimalisasi Parameter SVM dengan Kernel Cosine," *MEANS (Media Informasi Analisa dan Sistem)*, pp. 137–142, 2024, Accessed: Jul. 21, 2025. [Online]. Available: https://ejournal.ust.ac.id/index.php/Jurnal_Means/article/view/4357

[34] D. Dahlan, R. Iskandar, N. Ekawati, and C. A. Sugianto, "Mackerel Tuna Freshness Identification Based on Eye Color Using K-Nearest Neighbor Enhanced by Contrast Stretching and Histogram Equalization," *Scientific Journal of Informatics*, vol. 11, no. 4, pp. 1035–1042, 2024, https://doi.org/10.15294/sji.v11i4.14494.

[35] N. Fatima and V. Yadav, "Fish Species Classification Using Convolutional Neural Networks," in *International Conference on IoT, Intelligent Computing and Security: Select Proceedings of IICS 2021*, Springer, 2023, pp. 413–423. https://doi.org/10.1007/978-981-19-8136-4_34.

[36] B. Bischl *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges," *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 13, no. 2, p. e1484, 2023, https://doi.org/10.1002/widm.1484.

[37] MathWorks, "Classification Naive Bayes - Naive Bayes classification for multiclass classification," https://www.mathworks.com/help/stats/classificationnaivebayes.html.

[38] MathWorks, "fitcnb - Train Multiclass Naive Bayes model," https://www.mathworks.com/help/stats/fitcnb.html.

[39] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J Big Data*, vol. 6, no. 1, pp. 1–48, 2019, https://doi.org/10.1186/s40537-019-0197-0.

[40] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," *arXiv preprint arXiv:1712.04621*, 2017, https://doi.org/10.48550/arXiv.1712.04621.

[41] C. Hu *et al.*, "Impact of light and shadow on robustness of deep neural networks," *arXiv preprint arXiv:2305.14165*, 2023, https://doi.org/10.48550/arXiv.2305.14165.

[42] O. Mazhar and J. Kober, "Random shadows and highlights: A new data augmentation method for extreme lighting conditions," *arXiv preprint arXiv:2101.05361*, 2021, https://doi.org/10.48550/arXiv.2101.05361.

[43] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*, vol. 1. MIT press Cambridge, MA, USA, 2017.

[44] C. Zhou *et al.*, "Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision," *Aquaculture*, vol. 507, pp. 457–465, 2019, https://doi.org/10.1016/j.aquaculture.2019.04.056.

[45] R. Blanquero, E. Carrizosa, P. Ramírez-Cobo, and M. R. Sillero-Denamiel, "Variable selection for Naïve Bayes classification," *Comput Oper Res*, vol. 135, p. 105456, 2021, https://doi.org/10.1016/j.cor.2021.105456.

[46] P. Langley and S. Sage, "Induction of Selective Bayesian Classifiers," in *Uncertainty in Artificial Intelligence Proceedings of the Tenth Conference*, 2013. https://doi.org/10.1016/B978-1-55860-332-5.50055-9.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, https://doi.org/10.1038/nature14539.

[48] D. R. Amancio *et al.*, "A systematic comparison of supervised classifiers," *PLoS One*, vol. 9, no. 4, p. e94137, 2014, https://doi.org/10.1371/journal.pone.0094137.

[49] K. Munawaroh and A. Alamsyah, "Performance comparison of SVM, naïve bayes, and KNN algorithms for analysis of public opinion sentiment against COVID-19 vaccination on Twitter," *Journal of Advances in Information Systems and Technology*, vol. 4, no. 2, pp. 113–125, 2022, https://doi.org/10.15294/jaist.v4i2.59493.

[50] Y. Heningtyas, A. Syarif, and A. G. Pratidina, "Implementation of Gabor Filter for Carassius Auratus's Identification," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 11, no. 2, pp. 566–571, 2021, https://doi.org/10.18517/ijaseit.11.2.8128.

[51] D. Y. Kim, S. W. Park, and H. S. Shin, "Fish Freshness Indicator for Sensing Fish Quality during Storage," *Foods*, vol. 12, no. 9, May 2023, 10.3390/foods12091801.

[52] A. Sharma, "Naive Bayes with Hyperparameter Tuning," https://www.kaggle.com/code/akshaysharma001/naive-bayes-with-hyperpameter-tuning.

[53] X. Ding, "Texture Feature Extraction Research Based on GLCM-CLBP Algorithm," in *7th International Conference on Education, Management, Information and Mechanical Engineering (EMIM 2017)*, 2017. 10.2991/emim-17.2017.36.

[54] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation,"

https://www.machinelearningmastery.com/k-fold-cross-validation/, Oct. 04, 2023.

[55] D. C. Asogwa, S. O. Anigbogu, I. E. Onyenwe, and F. A. Sani, "Text classification using hybrid machine learning algorithms on big data," *International Journal of Trend in Research and Development*, vol. 6, no. 5, 2019, https://doi.org/10.48550/arXiv.2103.16624Focusto learnmore.

[56] F. Monteiro *et al.*, "Classification of fish species using multispectral data from a low-cost camera and machine learning," *Remote Sens (Basel)*, vol. 15, no. 16, p. 3952, 2023.