Improving automatic classifier performance for endangered Oryzias minnow species using YOLOv8

Armin Lawi[a,b], Muhammad Haerul[a], Iman Mustika Ismail[a], Rahmatullah R.[a], Irma Andriani[c], and Andi Iqbal Baharuddin[d]

[a]Information Systems Study Program, Hasanuddin University, Makassar, Indonesia
[b]B.J. Habibie Institute of Technology, Parepare, Indonesia
[c]Department of Biology, Hasanuddin University, Makassar, Indonesia
[d]Department of Fishery, Hasanuddin University, Makassar, Indonesia

November 8, 2024

Abstract

The small size of fish and the small amount of data that is available making the Detection of endangered animals a difficult problem to tackle. High accuracy are necessity in order to identify the species which are rarely to be found. Conservation wouldn't be possible if model miss to identify the rare species. Hence it's an urgency to achieve a maximum accuracy this applied not only in the conservation of endengared animal but also on other task which require a high accuracy for precise monitoring and protection

The experimental results show the model's exceptional performance, achieving mean average precisions of 88.4% and 87.7% on the DUO and UTDAC2020 datasets, respectively. Notably, the model operates at a rapid detection speed of 156 FPS, fulfilling critical real-time detection needs. With a concise model size of 4.4 M and a moderate computational complexity of 11.6 GFLOPs, it is highly suitable for integration into underwater detection systems.

1 Introduction

Technological advancements had made possible for automation of fish detection. This allows for the conservation of endangered animals can be initiated earlier as the initial stage using the artificial intelligence (AI) technology. The fundamental initial stage is the detection and classification of endangered species through the application of deep learning algorithms as an AI model. The application of AI enables the use of camera traps or advanced sensors to automatically identify endangered species, obviating the necessity for direct human involvement in their habitat. Furthermore, AI is capable of analyzing copious amounts of data from a multitude of sources, including satellite images and sound recordings, in order to discern patterns in animal behavior or distribution that are challenging to discern manually. Furthermore, AI can predict environmental threats to specific animal populations, such as climate change or deforestation, thereby enabling more rapid and accurate conservation efforts. Additionally, AI enables the simulation of the optimal environment for a particular species, thus facilitating more effective conservation planning in the future. AI can even be used in conjunction with robots to automatically manage conservation areas. By leveraging this technology, endangered animal conservation efforts can become more efficient and sustainable.

In the realm of deep learning, image-based detection and classification methods, such as YOLOv8, can be developed for animal conservation. YOLOv8 is an evolution of the SSD (Single Shot MultiBox Detector), a two-stage object detection model. This algorithm is optimized for real-time object detection and consistently achieves high-precision results across various applications.

Several recent studies have utilized YOLOv8 for fish detection and classification, showing the great potential of this technology in aquatic ecosystem conservation. One study successfully used YOLOv8 to identify various fish species in coral reefs with high accuracy, even in low-light conditions or blurred images due to water particles. Another study utilized YOLOv8 to monitor the movement and population of fish in rivers and lakes, where real-time analysis results enabled the detection of endangered species or invasion of non-native species. The speed and efficiency of YOLOv8 in processing underwater image data is an advantage, as the aquatic environment often does not support direct visual observation by humans. In addition, some projects have integrated YOLOv8 with underwater autonomous vehicles (UAVs) or automated underwater cameras, which can work continuously to monitor marine ecosystems without interruption. The results of fish classification using YOLOv8 also make important contributions to the study of fish behavior, migration patterns, and ecosystem interactions, which are important for conservation policies. The use of YOLOv8 in this fish study shows that AI technology is not only relevant for land, but can also play a key role in maintaining the balance of aquatic ecosystems. These studies are clear examples of how advanced AI algorithms can be applied to understand, protect and manage animal populations in aquatic habitats.

While YOLOv8 has many advantages in fish detection and classification, there are some drawbacks that arise when applied to small, rare fish species. YOLOv8, like other object detection algorithms, faces challenges in identifying small objects in the image, especially if the fish is moving fast or among visual complexities such as coral reefs or water particles. YOLOv8's accuracy can also degrade when small rare fish occupy only a fraction of the image pixels, making it difficult for the algorithm to recognize specific details. To overcome this drawback, one idea is to increase the resolution of the input image or utilize super-resolution techniques, which can magnify and clarify the details of small fish before they are processed by the detection algorithm. In addition, the theory of using multi-scale training and anchor-free detection can be applied, where YOLOv8 is trained with different sizes of objects to accommodate the scale variation of fish in real environments. The addition of more specific training data for small fish as well as the use of augmented data can help the model become more adaptive to environmental variations. The integration of other technologies, such as acoustic sensors that complement visual detection, can also be a multidisciplinary approach to improve the accuracy of small fish species classification. By continuously developing this solution, YOLOv8 can be improved to be more effective in detecting and protecting rare small fish species in aquatic ecosystems.

Many studies have focused on modifying the underlying architecture of YOLOv8 to enhance model accuracy such as YOLO-SAG, which introduces changes to the backbone and neck to improve performance. Another common approach involves modifying activation layers to boost accuracy and efficiency. For instance, integrating architectures and recently developed activation functions, e.g., Softplus, has shown promise in improving model performance. However, these approaches are primarily concerned with architectural modifications as the core of the model.

Despite these advancements, there remains a notable gap in research focused on statistical approaches to enhancing YOLOv8's performance without altering its architecture. As new boosting techniques continue to emerge, such as LPBoost, TotalBoost, BrownBoost, XGBoost, MadaBoost, and LogitBoost, the application of these statistical methods to YOLOv8 remains underexplored. Investigating these techniques within the context of object detection models such as YOLOv8 could provide valuable insights and further improve accuracy without the need for complex architectural modifications.

Hence this paper is made to fill the gap with the use of a mixture dataset, i.e., a collection of data compoundsl primary and secondary data. The use of the mixed datasets is necessary since primary data for rare or endangered animals is very difficult to capture in sufficient numbers. Therefore, secondary data from sources such as the internet is used to supplement the dataset. Once the dataset was collected, preprocessing was performed, including object segmentation, to ensure consistency of the data in terms of resolution, size, brightness, and other parameters, resulting in uniform and fair input. The main focus of this research is not on how the datasets are collected, but on how artificial intelligence (AI) technology can be implemented for the conservation of rare and endangered animals. Thus, this research focuses on the utilization of AI in facing conservation challenges, especially through automatic detection and classification that supports conservation efforts more efficiently and effectively.

The data used in this research utilizes a mixture dataset, which is a combined dataset between primary and secondary data. This is done because primary data for rare or endangered animals is very difficult to obtain. So that secondary data is needed as additional data obtained from the Internet. This hybrid data is then preprocessed with object segmentation so that the input data is uniform in terms of resolution, size, brightness and others of equal value (fair). The substance of this paper is not on how the dataset is obtained, but how the implementation of AI can be used for nature conservation issues, especially rare and endangered animals can be done using AI technology.

This paper successfully found a detection and classification model for Medaka fish. (*Oryzias javanicus* and *Oryzias celebensis*) The model used is based on YOLOv8 with a focus on setting up balanced and equitable data with a limited number of sample images with more than 90% accuracy (as the first contribution of this paper). The model used is based on YOLOv8 with a focus on balanced and fair data preparation. Although the data sources in this experiment came from direct observation data (primary data) and data derived from various sources on the Internet (secondary data), the resulting model performed very well. A careful and targeted data acquisition framework is the second contribution of this paper. Data preparation or data pre-processing works with three stages, namely, the business process understanding stage that directs the right data sources, the data understanding stage that produces data specifications according to the required data, and the data preparation or cleaning understanding stage. With these stages, the model built produces high accuracy because the data sources have even and quality specifications. The next contribution is that the model is developed through three experiments, namely, parameter fine tuning techniques, 5-fold cross validation, and ensemble techniques using AdaBoost. The three experiments show that the AdaBoost ensemble method is slightly superior to the 5-fold cross validation method and the parameter fine tuning method. The final contribution is the proposal of several new performance measures including model risk and confidence levels that are developed from accuracy measures.

The key contributions of this paper are as follows:

- A dataset consists of two classes: the endangered fish species Oryzias javanicus and Oryzias celebensis.
- Application of 5-fold cross validation on YOLOv8 object detection model
- Implementation of ensemble techniques using AdaBoost on YOLOv8 object detection model

2 Materials and Method

2.1 Data Sources

Sumber data yang digunakan dalam membangun model komputasi untuk deteksi dan klasifikasi ikan Oryzias sp. menggunakan YOLOyek data primer dan jumlahnya sangat terbatas

Data sources in this paper are mixture which is retained

- Data primer adalah data yang diperoleh secara langsung berdasarkan hasil deteksi tangkapan video dan citra ada di akurium.
- Data sekunder adalah data citra ikan Oryzias yang tersedia banyak di Internet
- Data buatan yang dibangkitkan dengan mekanisme Generative-AI yang mekanismenya

Semuanya akan dipreprocessing dengan metode yang sama

2.2 Research Methodology

The concept of data mining is used in this study as the fundamental method for detecting and classifying two species of Medaka fish, i.e., *Oryzias javanicus* and *Oryzias celebensis*. Using a *deep learning* model with the

YOLOv8 algorithm, one of the most advanced and efficient *object detection* methods, allows for simultaneous detection and classification. To systematically implement the data mining concept, this study proposes a data processing framework in three stages: the data preprocessing or data preparation stage, the data processing or model building stage, and the post-processing or model evaluation stage. The data preprocessing stage is highly crucial and important in building a model with good performance. The preprocessing stage consists of three sub-stages, including *Business Process Understanding*, *Data Understanding*, and *Data Preparation*. Below is an explanation of these three stages as illustrated in 1.

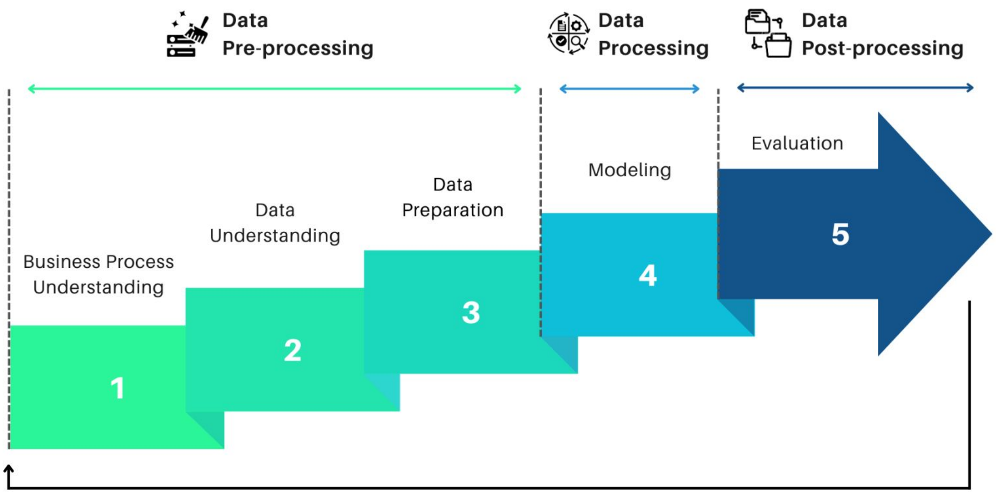


Figure 1: Stages in Data Mining Process

2.3 Data Preprocessing Stage

The data preprocessing stage, consists of 4 activities that are consecutive processes.

2.3.1 Problem Understanding

The main difficulty in designing and building AI intelligent information systems for the endangered Oryzias sp. is that the small size, transparent Datasets for the endangered Medaka fish are difficult to find or detect due to its small size and endemic nature. In addition, conservationists do not have datasets and even the image data of Medaka fish available on the Internet is very limited. However, if directly captured medaka fish image data is combined with medaka fish image data obtained from Internet sources, it can be feasibly used in building detection and classification models using deep learning model methods. With a certain framework based on digital image processing techniques, a feasible dataset can be generated to build the model. To be more convincing, primary data and secondary data are built with a minimum proportion of 70%:30%. Then a data normality test is conducted to ensure that the deviation or difference between primary and secondary data is very small.

In the Business Process Understanding stage, the primary objective of this study is to automatically detect and classify two species of Medaka fish, namely *Oryzias javanicus* and *Oryzias celebensis*, using artificial intelligence (AI) technology. This detection and classification aim to support conservation efforts for rare species and facilitate scientific research in this field. The algorithm used, YOLOv8, was chosen due to its efficient and real-time object detection capabilities, which are highly suitable for the identification of fish species in various conditions.

The data collection process in this research uses modality data in the form of digital images, which consist of images of fish taken directly or obtained from secondary sources through the data scraping method. Since this data is an unstructured dataset, data processing is done using deep learning methods. Deep learning enables effective processing and analysis of complex image data to detect and classify objects in images with high accuracy.

The fish images are collected in RGB (Red, Green, Blue) or BGR (Blue, Green, Red) color formats, which are standard formats in digital image processing. These formats are compatible with most of the deep learning algorithms used in this study. The selection of this format aims to allow the image data to be optimally processed by the detection algorithm.

To solve this fish species detection and classification problem, this research uses the YOLOv8 algorithm, which is one of the latest deep learning methods for object detection. This algorithm is able to detect and classify objects in real-time with high accuracy despite variations in image capture angle, object size, and lighting. With this method, it is expected that the model will be able to correctly recognize fish species in various conditions and help in efforts to conserve and monitor Medaka fish populations in the wild.

2.3.2 Mixture Data Sources

Additional:

1. Data sourced from primary and secondary data
2. The dataset needs to be normalized to ensure that there is no deviation or difference in data so that the mixed dataset is suitable for processing.

This study uses a **dataset mixture**, which is a combination of primary and secondary data. The **primary dataset** was obtained through direct observation, by taking photos of Medaka fish in various locations. This data includes images of two fish species, *Oryzias javanicus* and *Oryzias celebensis*. The **secondary dataset**, on the other hand, was collected from online sources using *data scraping* techniques to gather additional images of both species. The orientation, *size*, and *image parameters* were standardized so that both the primary and secondary data could be used fairly.

The use of a mixed dataset of primary and secondary data in this study has provided significant benefits. The primary dataset was obtained through direct observation, while the secondary data was collected through data scraping techniques. This approach created the variation required to properly train the YOLOv8 model, enabling the detection and classification of two fish species, O. celebensis and O. javanicus, under various lighting conditions and viewing angles.

2.3.3 Data Labeling

Additional: Explanation of the labeling mechanism using Roboflow. In a single file, there may be more than one object with different classes. Capture of still image objects is taken from various pose angles of the moving object.

Object labeling work step:

- All 792 image files (without annotations) were placed in one folder
- Each image is annotated by giving a bounding box and label to each object in the image
- Each annotation result per each image is saved as a txt file with the same name between the image file and the label file

2.3.4 Splitting Data

The splitting ratio of data used in training and testing the model is 4:1. Depending on the experiment conducted, the training data was

2.3.5 Data Understanding

The data acquisition process in this study was conducted through various modalities to ensure that the resulting dataset is comprehensive and of high quality. Data was collected from two main sources: primary data, which was obtained through direct image capture of Medaka fish in various places, and secondary data, which was obtained from online sources using data scraping techniques. The images were captured with various viewing angles and lighting conditions, in order to enrich the variety of data needed to train the deep learning-based object detection model.

The acquired dataset consists of two main components, namely image data (X) and label data (Y). The image data (X) is a digital image of Medaka fish that will be used as input for the model, while the label data (Y) is the detection and classification information associated with the image, namely the fish species (Oryzias javanicus or Oryzias celebensis) and the coordinates of its bounding box in the image. This label data is very important as it is used as a reference in the model training process to learn the relationship between the image and the correct label.

With the combination of image data (X) and label data (Y), the YOLOv8 model can be trained to automatically detect and classify fish species. This varied data collection from different modalities ensures that the model has the ability to recognize Medaka fish in various situations and conditions, which is important to achieve good generalization and high accuracy in detection.

2.3.6 Data Preparation

The Data Preparation stage is an important step in ensuring that the dataset is ready to be used for deep learning model training. In this research, several processes are carried out to prepare the data, including image restoration, data uniformity, data augmentation, and division of the dataset into appropriate subsets.

a. Image Restoration
   Image restoration is performed to improve the quality of images that may have suffered damage or noise during the data capture process. This process includes blurred image repair, contrast improvement, and removal of visual artifacts that may interfere with the performance of the detection model. The goal is to produce clearer and more consistent images, so that the model can more easily detect and classify objects.

b. Data Uniformity
   After restoration, the next step is data homogenization to ensure that all images have the same parameters. Some of the aspects that are uniformed include:

   i. Orientation: Images that have different orientations (e.g. upside down or skewed) are adjusted for uniformity.

   ii. Resize: All images are resized to have the same dimensions, so that they can be processed by the model efficiently.

   iii. Image Parameters: Image parameters such as contrast, saturation, and blur are also adjusted to ensure that there are no significant visual differences between the images, which may affect the performance of the model. In this way, the input images have a consistent state so that the model can focus on the important features of the object.

c. Data Augmentation
   To increase the amount of data and enrich the variety of images, data augmentation is performed. Augmentation helps the model become more robust to variations in new images that may not be encountered during training. The augmentation techniques used in this research include:

   i. Flip: The image is flipped vertically and horizontally to produce variations in the position of the object.

   ii. Rotate: The image is rotated in several fixed angles of 90 degrees, 180 degrees, and 270 degrees to increase the variety of viewpoints of the objects in the image. This augmentation technique is very effective for improving the generalization ability of the model without the need to augment the original data.

d. Dataset split (Train, Validation, Test)
   After the data is processed, the dataset is divided into three subsets: training set, validation set, and test set. This division is done in a certain proportion to ensure that the model is properly trained, validated, and tested. Generally, the data is divided with a 4:1 composition between train data and test data.

i. Training sets are used to train the model and help the model learn patterns from the data.

ii. Validation sets are used to monitor the performance of the model during training and assist in the hyperparameter tuning process.

iii. Test sets are used after training is complete to test the model's capabilities on data that has never been seen before, in order to assess the model's generalization performance in the real world.

2.4 Data Training or Modeling

In the Modeling stage, three main experiments were conducted to improve the performance of Medaka fish species detection and classification using a deep learning-based approach. Each experiment had different objectives and methods, namely fine-tuning YOLOv8, 5-fold cross-validation YOLOv8, and using the AdaBoost ensemble method. The following is an explanation of each experiment:

2.5 Data Post-processing or Evaluation

Model evaluation was conducted to assess the performance of the YOLOv8 model in detecting and classifying Medaka fish species. The evaluation process includes the use of a loss function to assess the quality of training as well as a confusion matrix-based evaluation matrix to measure the model's performance on test data. The following is an explanation of the evaluation methods used:

2.5.1 Evaluation based on Loss Function (Data Validation)

During the training and validation process, the model is evaluated using several loss functions that measure the prediction error on various aspects:

a. Bounding Box Loss: Measures how well the model predicts the coordinates of the bounding box surrounding the fish object in the image. The lower the box loss, the more accurate the prediction of the object's location in the image.

b. Classification Loss (Cls Loss): Measuring the misclassification of fish species (*Oryzias javanicus* or *Oryzias celebensis*) predicted by the model. A low Cls loss indicates that the model is able to classify the object correctly.

c. Distribution Focal Loss (DFL Loss): Measures how well the model predicts the confidence score distribution of the bounding box prediction.

The results of each loss function are evaluated at train loss (during training) and validation loss (during validation) to identify whether the model is overfitting or underfitting. A decrease in the training and validation loss values indicates that the model is learning well from the data.

2.5.2 Evaluation Based on Confusion Matrix (Data Test)

After training, the performance of the model is evaluated on the test data (test set) using the confusion matrix, which is a table that shows the classification results of the model against the test data. From the confusion matrix, several important evaluation metrics are calculated:

a. Accuracy: Measures the proportion of correct predictions out of all predictions. This metric shows how often the model makes correct predictions overall.

b. Precision (Sensitivity): Measures how many true positives out of all positive predictions are accurate. High precision indicates that the model rarely makes mistakes in classifying objects as a particular species.

c. Recall: Measures how well the model detects all true positive objects. A high recall indicates that the model is able to capture most of the objects that actually exist.

d. F1-Score: Is a combination of precision and recall, providing a more balanced picture of the model's performance in cases where it is important to balance between precision and recall.

Confusion matrix is also used to calculate True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), which form the basis for calculating other evaluation metrics. Using the confusion matrix, model performance can be evaluated in more detail, providing an understanding of how the model performs on actual test data.

Table 1: Confusion Matrix

	Predicted (+)	Predicted (−)
Actual (+)	TP	FN
Actual (-)	FP	TN

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

$$\text{Accuracy} = \frac{1}{N}\sum_{i}\{\text{I}(\hat{y} = +|y = +) \text{ or } \text{I}(\hat{y} = -|y = -)\} \tag{2}$$

$$Precsission = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$\begin{aligned} F1 - score &= \frac{TP + TN}{TP + \frac{1}{2}(FP + FN)} \\ &= \frac{2 \times precission \times recall}{precission + recall} \end{aligned} \tag{5}$$

3 Experiments

3.1 Experimental environment and parameter configuration

The experimental setup for this study utilized an NVIDIA GeForce RTX4080 GPU with 16GB of memory, Python 3.8.18, PyTorch 1.7.1 framework, and CUDA 12.0. The training is performed using a homemade wildlife dataset, and the experimental parameters are shown in Table 2.

3.2 Evaluation index

(1) Precision evaluation metrics: Precision (P), Recall (R), mean Average Precision (mAP), and F1 score. mAP@0.5 and mAP@0.5–0.95 denote the mAP value at an IoU threshold of 0.5 and the average mAP value as the IoU ranges from 0.5 to 0.95 with a step size of 0.05, respectively. The formulas for each index are shown in Eqs. 12–15, where TP represents the number of true positive predictions, FP denotes the number of false positive predictions, and FN signifies the number of false negative predictions. (2) Speed evaluation metrics: inference time (time from preprocessed image input into the model to model output result), post-processing time, floating-point operations(GFLOPs), model size, and parameter. P = TP TP + FP (12) R = TP TP + FN

3.3 Experiment 1: Fine-Tuning YOLOv8

The first experiment was conducted by applying fine-tuning to the YOLOv8 model. YOLOv8 is one of the latest object detection models designed to detect and classify objects in real-time. In this experiment, the pre-trained YOLOv8 model was fine-tuned to the collected Medaka fish dataset. Fine-tuning involves adjusting the pre-trained weights using the Medaka fish data to allow the model to recognize the unique features of the species. *Oryzias javanicus* and *Oryzias celebensis*.

The fine-tuning process is performed using a dataset that has been prepared through preprocessing and augmentation stages, with optimal hyperparameter settings. The goal of this experiment is to improve the accuracy of fish species detection and classification by using the knowledge gained from large datasets and adapting it to more specific datasets.

Fine-tuning the YOLOv8 model is a crucial step to adapt the model's performance to the specific characteristics of the dataset used in this study. YOLOv8, as one of the most advanced object detection algorithms today, requires a fine-tuning process to optimize its ability to detect rare or endangered species. In this process, the pre-trained weights of the model were adjusted using training data from the mixture dataset we collected, which is a combination of primary and secondary data.

The fine-tuning process allows the model to learn unique features of our dataset, such as specific environmental variations, lighting conditions, as well as special traits present in the target species. By applying an approach where some network layers remain frozen while other layers are updated, we were able to utilize the general knowledge gained from the large-scale dataset while still tailoring the model to the unique characteristics of this dataset. This strategy is important for improving detection accuracy, reducing the risk of overfitting, and ensuring the model is reliable in various environmental conservation scenarios.

In this study, the YOLOv8 model is fine-tuned using a mixture dataset that has been divided into 80% for training data, 20% for testing data. The model was trained with optimally tuned hyperparameters, namely for 100 epochs, using a batch size of 16, with an automatic optimizer, and a learning rate of 0.01. The model training lasted for approximately 2 hours, utilizing GPU-based computing infrastructure to speed up the process.

3.4 Experiment 2: 5-Fold Cross Validation

In the second experiment, the model was evaluated using the 5-fold cross validation technique. Cross-validation is a method used to ensure that the model has stable performance and can be generalized to various subsets of data. In 5-fold cross-validation, the train data is divided into five subsets, where at each iteration, four subsets are used to train the model, and one subset is used for validation. This process is repeated five times so that each subset serves as a validation set once.

The results of each iteration are then averaged to obtain more accurate performance metrics, such as precision, recall, and mean average precision (mAP). By using cross-validation, the risk of overfitting the model can be minimized, allowing the model to perform optimally on unseen data.

Cross-validation is used to evaluate the reliability and generalizability of YOLOv8 model performance. In this study, we applied 5-fold cross-validation, where the dataset is divided into five subsets (folds). At each iteration, the model is trained using four subsets and tested on the remaining subset. This process is repeated five times, with each fold acting as a one-time validation set. Performance metrics, such as precision, recall, and mean average precision (mAP), are calculated at each iteration and then averaged to provide an accurate and reliable estimate of the model's overall capability.

Cross-validation is important to reduce the risk of overfitting, because by dividing the data into multiple subsets and testing the model iteratively, we can ensure that the model performs consistently even if the data is divided in different ways. It also helps in ensuring that the model is more resilient to variations in the dataset and reliable when applied to data outside the training sample.

In this research, the dataset is divided into 80% (508 images) for the training set and 20% (158 images) for the test set. The training set was further divided into five subsets (A, B, C, D, and E), each containing 127 images. The model was trained using four subsets as training set and one subset as validation set at each iteration. This training process was repeated five times, rotating among the five subsets. Illustration of experiment 2 or cross validation as shown in Figure 2.

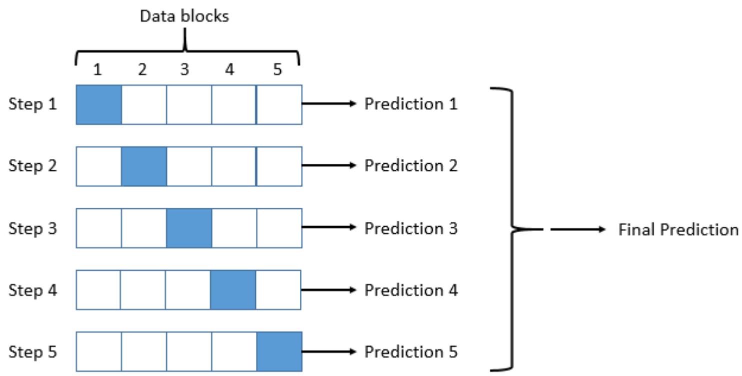


Figure 2: 5-Fold Cross Validation

The model was trained using optimally tuned hyperparameters for 100 epochs, with batch size 16, auto optimizer, and learning rate 0.01. Each training iteration took about 2 hours, so the total time required to complete five cross-validations was approximately 10 hours.

3.5 Experiment 3: Ensemble Method (ADABoost)

The third experiment involved applying ensemble methods to further improve detection and classification performance. The method used is AdaBoost, an ensemble algorithm that works by combining multiple models to improve prediction accuracy. In this approach, the YOLOv8 model is combined with several other models, or variations of the YOLOv8 model with different hyperparameter settings, to form an ensemble system.

The main motivation for using AdaBoost is to improve model performance by correcting the weaknesses of individual models that may fail to recognize certain patterns. The AdaBoost algorithm gives greater weight to the prediction errors of the previous model, so that the next iteration of the model focuses more on correcting those errors. In this way, the resulting model is more robust and accurate, and has a better ability to generalize predictions, even on complex or variable data.

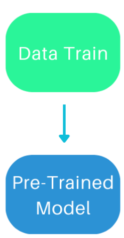
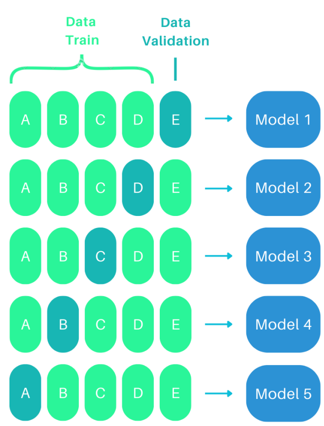
Ensemble learning is a technique used to improve model performance by combining predictions from multiple models. In this study, we apply an ensemble approach by combining multiple YOLOv8 models, where each model is trained with slightly different hyperparameters or trained on different subsets of data. The purpose of this ensemble is to reduce the variance and bias that may exist in each individual model, resulting in more accurate and reliable predictions. This ensemble approach can be done by averaging outputs (such as bounding box coordinates and confidence scores) or using a majority voting mechanism for classification. This technique helps improve overall robustness and predictive ability, especially when dealing with real-world data complexity and diversity.

Experiments using the ADABoost ensemble technique were conducted by utilizing 5 models from the cross validation results in experiment 2. The ADABoost algorithm is as follows

Algorithm: AdaBoost

1. **Input**: Dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$,
   Learner Γ and the number of learning iteration T
2. Initialize weight sample $w_i = \frac{1}{N}, \forall i = 1, 2, \ldots, N$
3. Iterate, **for** $t = 1$ **to** T **do**
   (a) Train a weak learner h_t from $D_t (\in D)$ to train sample w_i. $h_t = \Gamma(D, D_t)$
   (b) Compute error of h_t: $\varepsilon_t = \frac{\sum_{i=1}^{N} w_i \cdot I(h_t(x_i) \neq y_i)}{\sum_{i=1}^{N} w_i}$
   (c) Compute the weight of h_t: $\alpha_t = \frac{1}{2} \ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right)$
   (d) Assign: $w_i \leftarrow w_i \cdot e^{(\alpha_t \cdot I(h_t(x_i) \neq y_i))}$
4. **Output:** $H(x) = \text{sign} \sum_{t=1}^{T} (\alpha_t \cdot h_t(x))$

Table 2: table

<table>
<tr><th>Fine-Tuning</th><th>Cross Validation</th><th>Ensemble (AdaBoost)</th></tr>
<tr><td>
</td><td>
</td><td>gambar</td></tr>
</table>

4 Results

4.1 Mixture Data Preparation

This subsection presents the results of the mixed data preparation process with the stages described in the subsection 2.2.

Data preparation sequence:

1. download digital image file
2. Restoration of all digital objects in the file by removing the noise in the file. For video files or moving images, restoration is performed on each frame of the captured image per-second.
3. The file restoration process is performed by removing noise applying a Gaussian kernel function.
4. All image files are then normalized to 640x640 pixels.
5. Normalized files containing multiple objects are annotated according to their respective species names.

4.1.1 Data Acquisition and Labeling

The data preparation stage was initially performed by collecting primary data from captured still images and moving videos into a main folder[1]. Next, all digital objects in a file were provided with bounding boxes and annotated with labels corresponding to their species. (The annotation results of the digital objects were stored as metadata along with other information such as location, time, authorship, etc., in a file). A summary of the dataset obtained using the methods described in subsection 2.3.2 can be seen in Table 3

Table 3: Recapitulation of Primary and Secondary Data

No.	Fish Class	Data Primer	Data Sekunder	Jumlah
1.	*O. Javanicus*	257	178	435
2.	*O. Celebensis*	287	70	357
	Total	**544**	**248**	**792**

The mixture data consisted of 792 total images, i.e., 435 (55%) images of O. Javanicus and 357 (45%) O. celebensis fish images which after preprocessing were divided into subsets for each experiment. Primary data yielded 544 (69%) fish and secondary data 248 (31%) fish.

4.1.2 Image Restoration

Remove noise by applying brightness, contrast and gaussian kernel function settings. Size

4.1.3 Normality Test of Data

A normality test is performed to show that the primary and secondary data in the dataset are equivalent.

4.1.4 Data Splitting

Experiment 1: Data Train 634 (80%), Data Test 158 (20%) - Model Experiment 1

Experiment 2: Train data is divided into 5 groups with an arrangement of 127 (16%) each. Each group is named dataset A, B, C, D, and E. Then the model is cross validated with a combination of train data and validation data ratio of 4:1 or 80% training data and 20% validation data from 634 train data in experiment 1. Overall, train data 508 (64%), Validation Data 127 (16%), and Test Data 158 (20%). The simulated cross validation experiment design can be organized as follows.
Model 1: ABCD as Train, E as Validation (80,20)
Model 2: ABCE as Train, D as Validation
Model 3: ABDE as Train, C as Validation

[1]Digital still image and moving video of an object hereafter are called digital object

Model 4: ACDE as Train, B as Validation
Model 5: BCDE as Train, A as Validation
Selection of the best model is obtained by selecting the model with Accuracy and Best Fitting.

Experiment 3: AdaBoost ensemble with weak learner model taken from experiment 2.

4.2 The Model Results

4.2.1 Experiment 1: Single Model (YOLOv8 Fine-Tuning)

In the first experiment, the YOLOv8 model was trained with data divided in the ratio of 70:20:10 for training, validation, and testing. The dataset consists of: Train data: 556 images (70%) Validation Data: 160 images (20%) Test Data: 77 images (10%)

The model was trained for 100 epochs with a batch size of 16. Figure 1 shows a consistent downward trend in train/box_loss, train/cls_loss, and train/dfl_loss, indicating the model was able to learn effectively. The loss on the validation data also decreased despite slight fluctuations at the beginning of training, which stabilized near the end of training.

Figure 2 displays the Precision and Recall metrics, both of which show significant improvement. The model's Precision stabilized around 0.8, while the Recall reached 0.9, demonstrating the model's ability to correctly classify both fish species.

The model was evaluated using Confusion Matrix (Figure 3), where the model successfully detected O. celebensis with a precision of 0.96 and recall of 0.95, and O. javanicus with a precision of 0.87 and recall of 0.81. Although there were some errors in the classification of O. javanicus, the overall results show that the model was quite effective in this fish species detection and classification task.

4.2.2 Experiment 2: 5-Fold Cross Validation

In the second experiment, the 5-fold cross-validation method is applied to evaluate the stability and generalization ability of the YOLOv8 model. The dataset is divided into three main subsets: Train data: 508 images (64%) Validation Data: 127 images (16%) Test Data: 158 images (20%)

The training process was conducted with a variety of subsets used as validation sets within each fold. Five models were trained using different combinations of data in each fold. The data sharing scheme for each model is as follows: Model 1: Trained on subsets A, B, C, D and validated on subset E Model 2: Trained on subsets A, B, C, E and validated on subset D Model 3: Trained on subsets A, B, D, E and validated on subset C Model 4: Trained on subsets A, C, D, E and validated on subset B Model 5: Trained on subsets B, C, D, E and validated on subset A

Each cross-validation iteration allows each subset to act as the validation set once, while the other subset is used as the training set. After the 5-fold cross-validation process is complete, the model performance metrics are averaged to obtain a more robust and generalizable performance estimate.

The cross-validation results show that the average mAP50 value of all folds is 0.78, with an average precision of 0.77 and an average recall of 0.82. The application of the cross-validation method is essential to ensure that the model not only performs well on the training data, but is also able to maintain its performance on unseen data. This proves the model's ability to recognize fish species with a high degree of accuracy outside the training data, ensuring better generalization ability and reducing the risk of overfitting.

Based on the results of 5-fold cross-validation and visual analysis on the performance metrics of each model, Model 4 showed the best results with more consistent and higher precision, recall, and mAP values than the other models.

4.2.3 Experiment 3: Ensemble Method (AdaBoost)

In the third experiment, the ensemble approach was applied by combining five models from the cross-validation results using the AdaBoost algorithm. The aim was to improve the prediction accuracy by correcting the weaknesses of each individual model. Each model was given a different weight based on their performance, with the model that had more prediction errors gaining more weight in subsequent iterations.

After using the AdaBoost ensemble, mAP50 increased to 0.81, and mAP50-95 increased to 0.63. In addition, precision increased to 0.82, while recall increased to 0.86. Combining these models proved to be effective in correcting errors that may occur in individual models, especially in the case of images that are more difficult to identify due to differences in lighting or object position.

In the third experiment, the ensemble approach was applied by combining five models from the cross-validation results using the AdaBoost algorithm. The aim was to improve the prediction accuracy by correcting the weaknesses of each individual model. Each model was given a different weight based on their performance, with the model that had more prediction errors gaining more weight in subsequent iterations.

After using the AdaBoost ensemble, mAP50 increased to 0.81, and mAP50-95 increased to 0.63. In addition, precision increased to 0.82, while recall increased to 0.86. Combining these models proved to be effective in correcting errors that may occur in individual models, especially in the case of images that are more difficult to identify due to differences in lighting or object position.

4.3 Performance Evaluation

Performance evaluation of the ensemble models showed that combining the five models with the AdaBoost technique resulted in a significant improvement in the detection and classification of both fish species. With an mAP50 of 0.80 and mAP50-95 of 0.65 on the test dataset, this indicates that the ensemble method is highly effective in dealing with real-world image complexity.

The Confusion Matrix of the ensemble model shows an increase in better detection for both species. O. celebensis had a precision of 0.96 and recall of 0.95, while O. javanicus had a precision of 0.87 and recall of 0.81. These results show a steady improvement from each experiment, with the ensemble model giving the best performance.

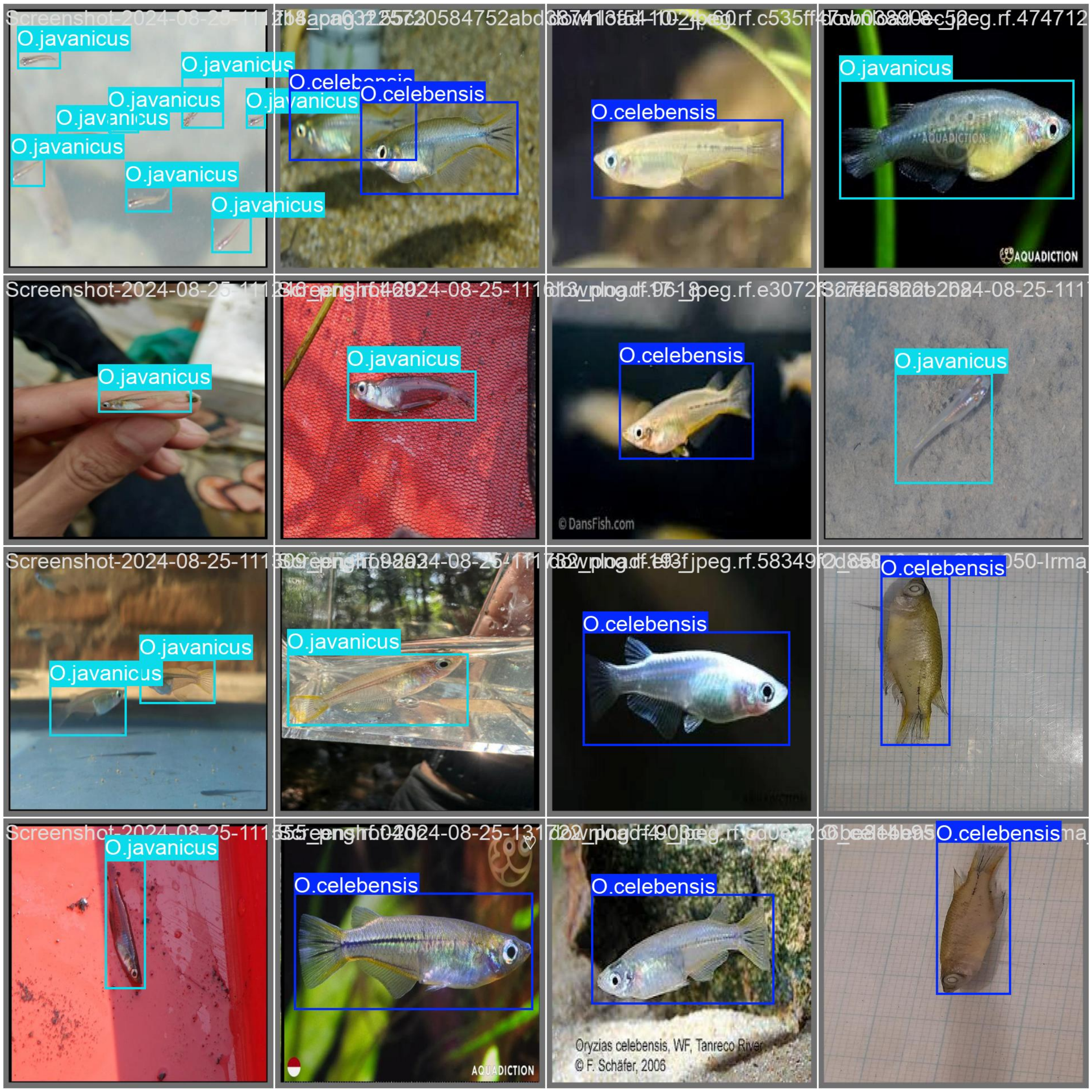


Figure 3: Resulted mixture images data with their annotation respectively

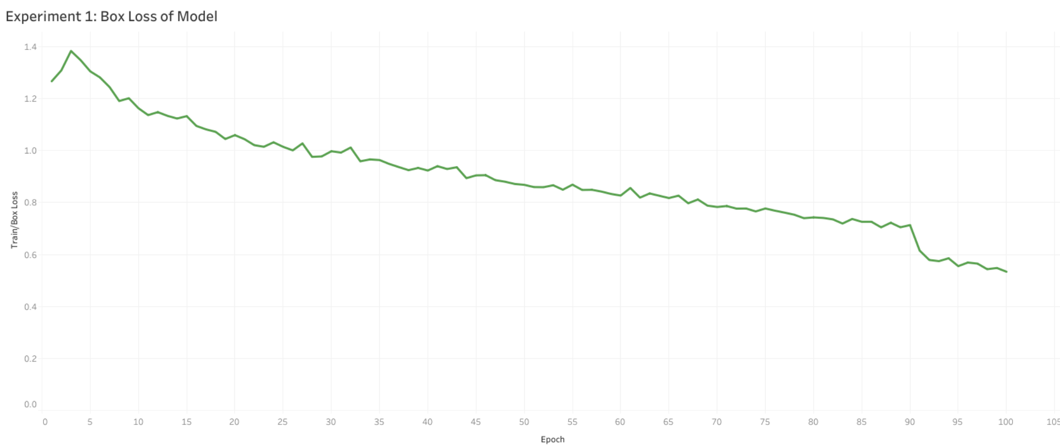


Figure 4: Caption

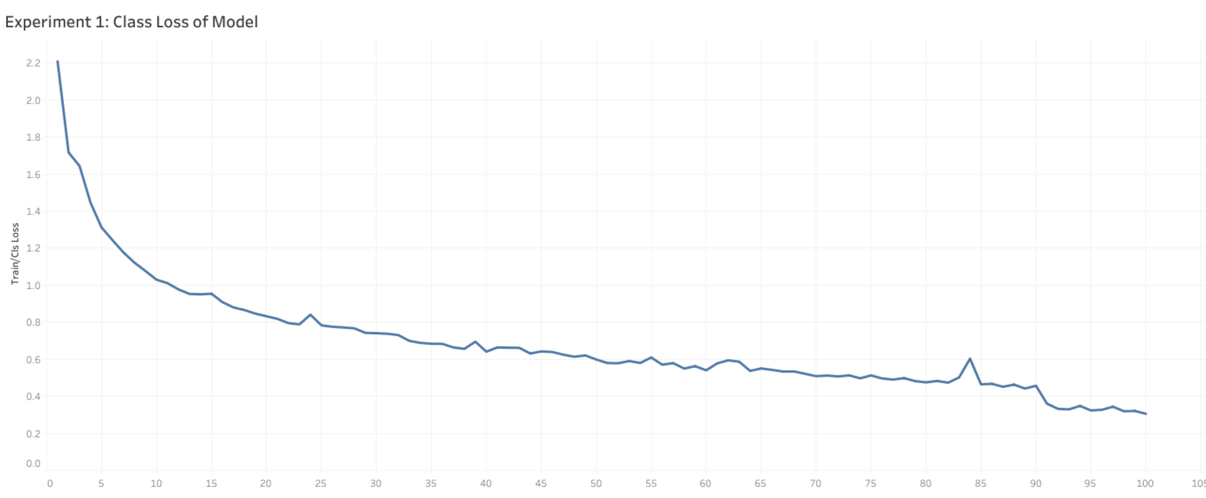


Figure 5: Caption

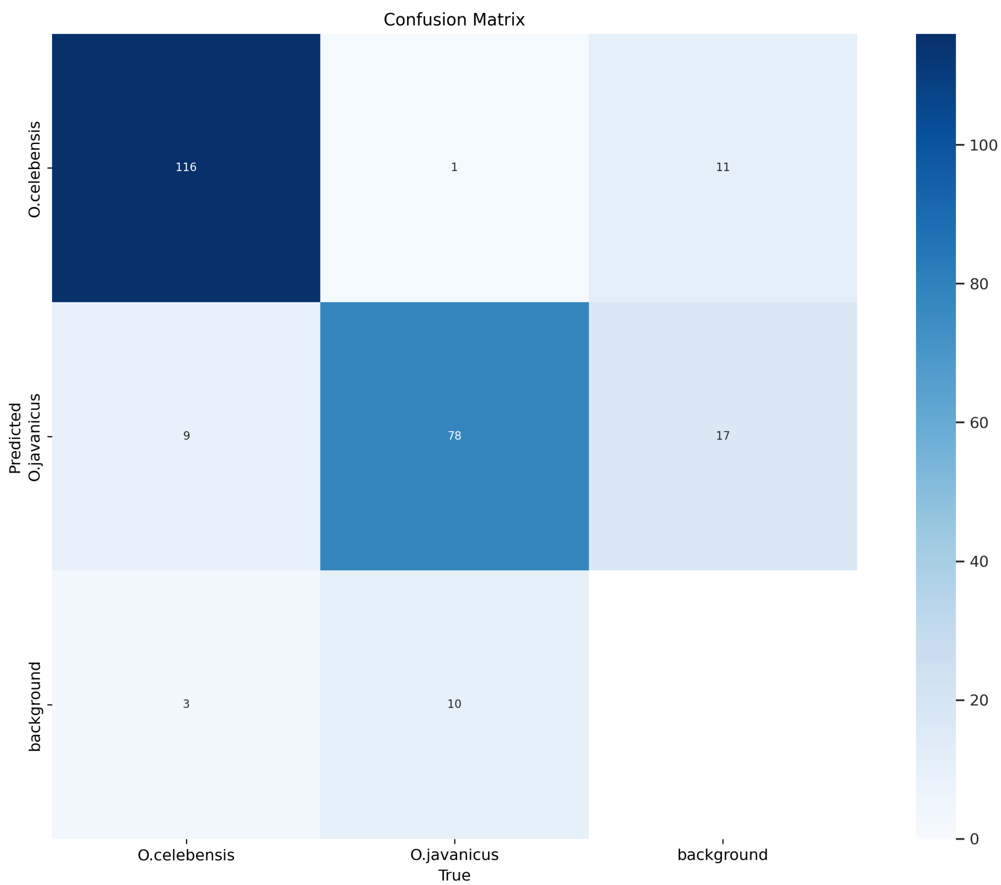


Figure 6: Caption

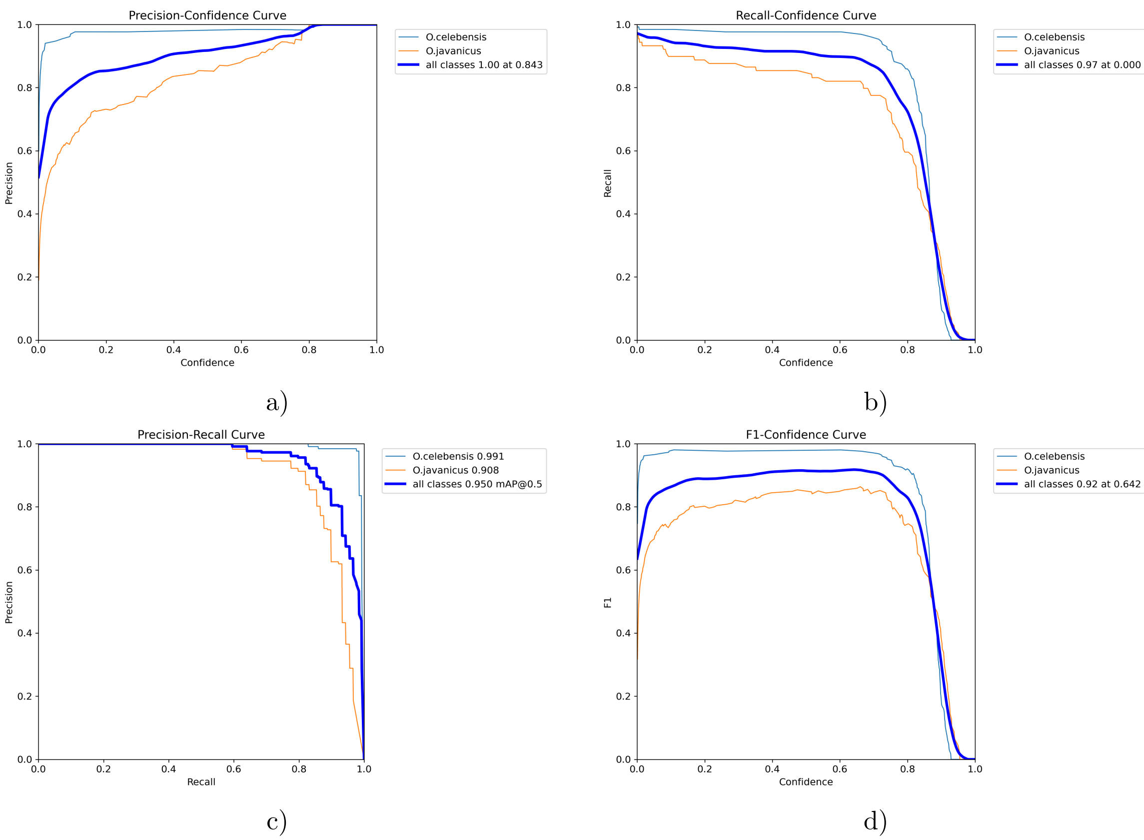
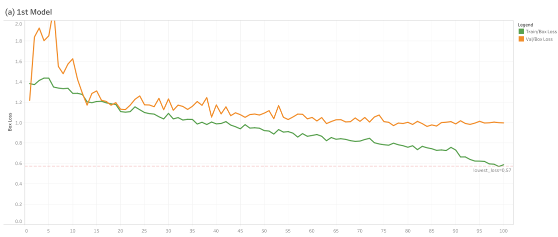
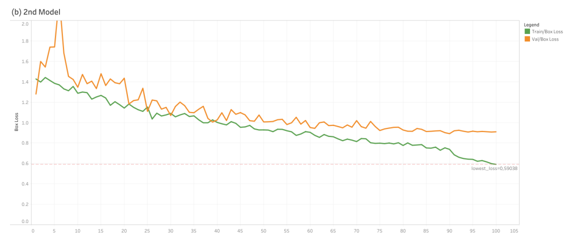


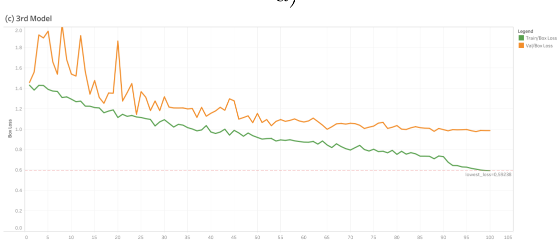
Figure 7: Caption



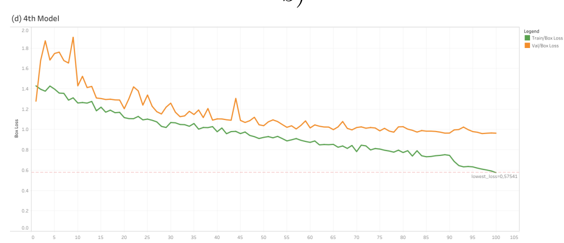
a)



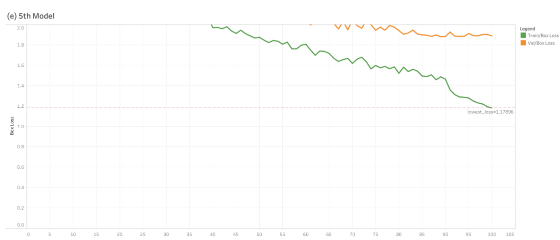
b)



c)

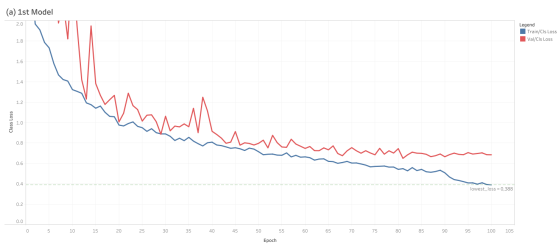


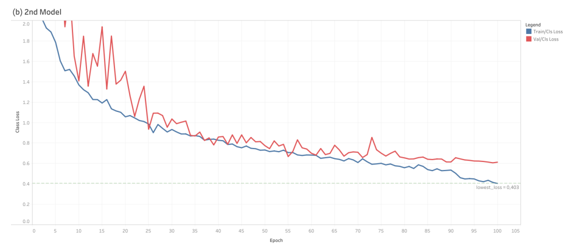
d)



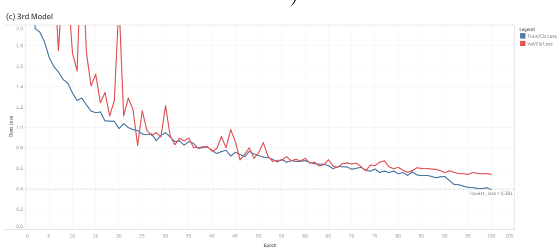
e)

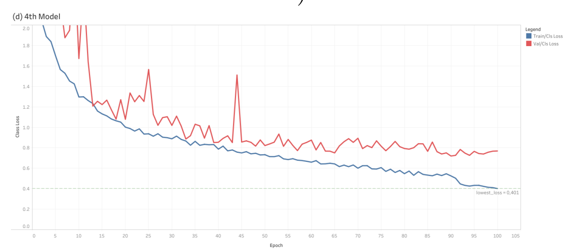
Figure 8: Caption



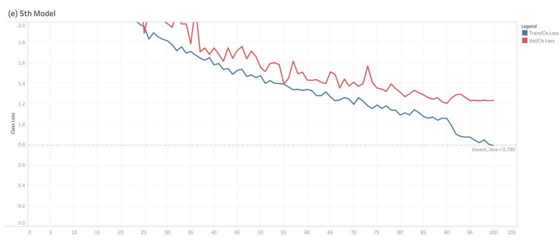


a) b)



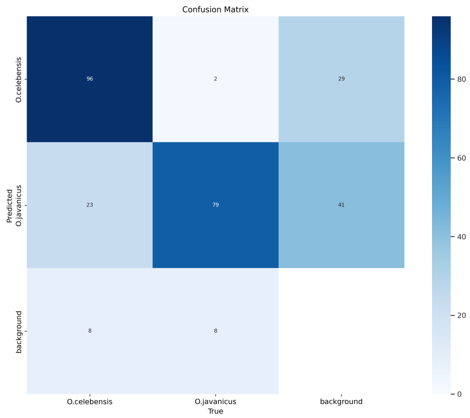


c) d)

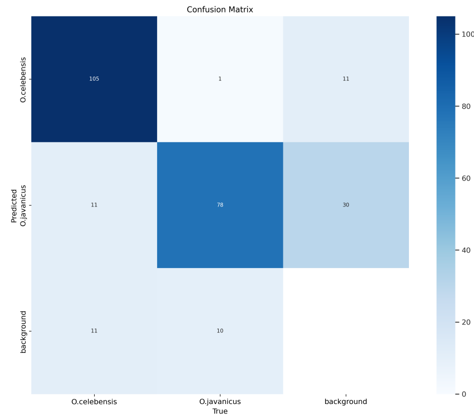


e)

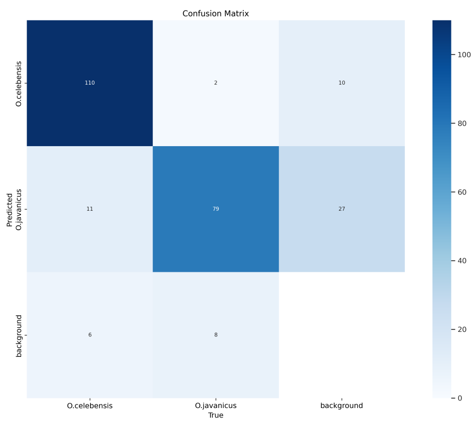
Figure 9: Caption



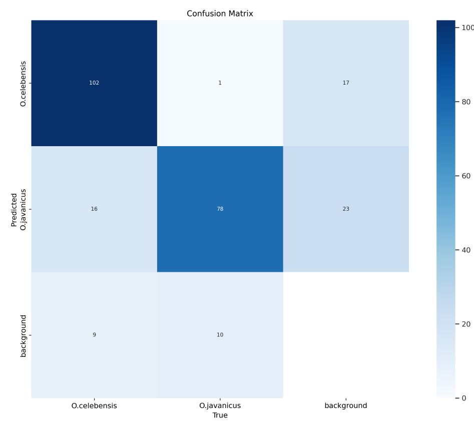
a)



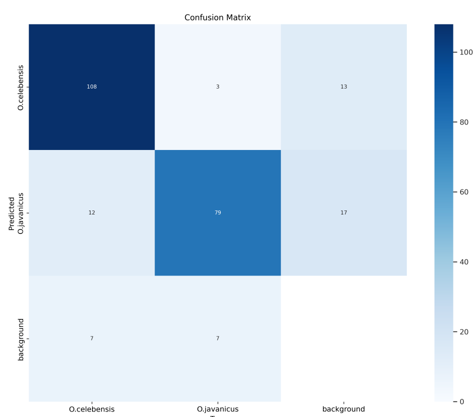
b)



c)



d)



e)

Figure 10: Caption

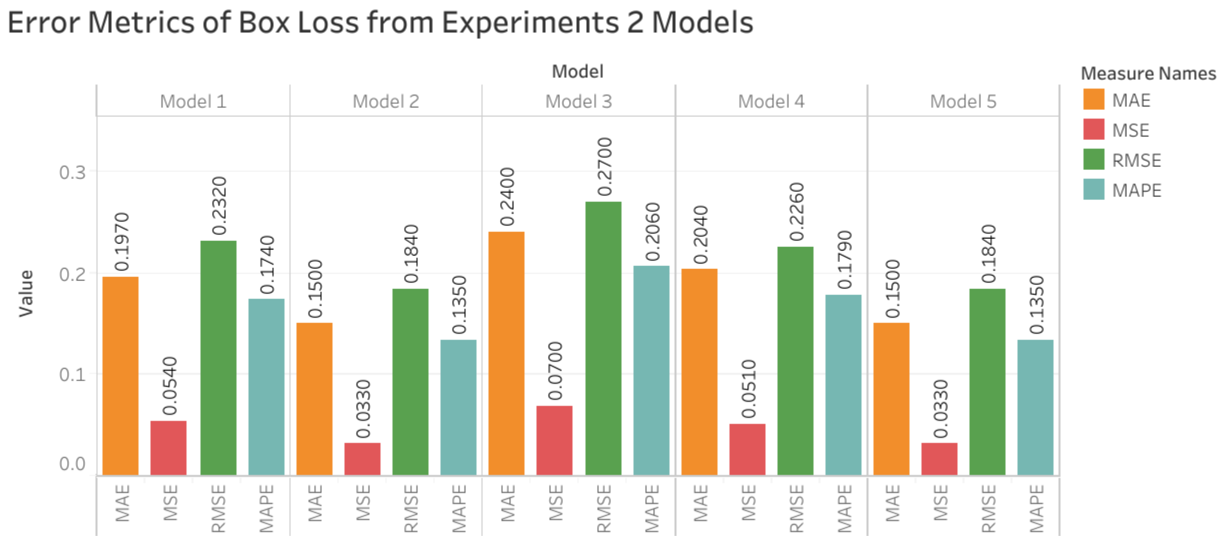


Figure 11: Caption

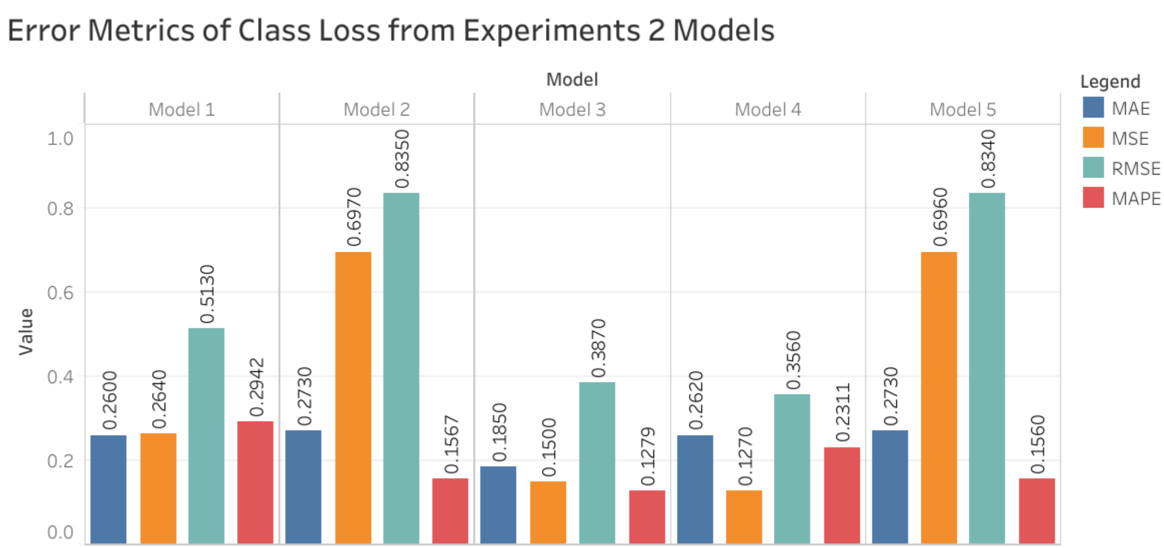


Figure 12: Caption

5 Discussion

5.1 AdaBoost Ensemble Method

One of the main contributions of this research is the application of the AdaBoost ensemble method to the YOLOv8 model. The AdaBoost algorithm helps mitigate the limitations of individual models by focusing on misclassified samples. In each iteration, greater weight is assigned to samples that were incorrectly predicted in the previous iteration, allowing the ensemble to improve detection accuracy for more challenging images, such as those involving small or partially obscured Medaka fish.

By integrating AdaBoost with YOLOv8, the model becomes more robust, especially in scenarios where the data is imbalanced or complex. This ensemble approach also contributes to reducing bias, enhancing the model's ability to generalize beyond the training data. The improved mAP values, particularly in difficult image scenarios, demonstrate the ensemble's effectiveness in boosting prediction accuracy.

5.2 5-Fold Cross-Validation for Model Generalization

Another major contribution of this paper is the implementation of 5-fold cross-validation, which played a critical role in preventing overfitting and ensuring that the model performed well on unseen data. By splitting the dataset into five subsets and training the model iteratively, the process provided more reliable and stable performance metrics. Each subset was used once as the validation set while the other four subsets served as training data, resulting in averaged performance metrics that better reflect the model's true capabilities.

This method enabled a thorough evaluation of YOLOv8's performance under different data conditions, showing that the model was capable of detecting and classifying Medaka fish species consistently, even in challenging real-world images. The consistent high precision and recall values across the folds demonstrate the model's strong generalization ability.

5.3 Key Contributions to Real-World Application

The integration of these techniques into a YOLOv8-based system for endangered species detection contributes significantly to the field of conservation technology. The ability to detect small, rare species with high accuracy has immediate applications in monitoring populations, tracking habitat changes, and assisting conservation efforts in real-time. AdaBoost and 5-fold cross-validation ensure the model's robustness and reliability in varying environments, offering a practical solution for real-world applications where data might be limited and imbalanced.

6 Conclusion

In this research, we applied AdaBoost and 5-fold cross-validation techniques to enhance the YOLOv8 model for the detection and classification of rare Medaka fish species. These contributions are particularly significant in the context of species conservation, where the accurate identification of endangered animals is critical. Our experiments demonstrated that combining AdaBoost with YOLOv8 improved overall precision and recall, making the model more robust in complex environments, such as underwater habitats where visual challenges are frequent.

Additionally, the use of 5-fold cross-validation allowed for a more thorough evaluation of the model's performance, minimizing the risk of overfitting and enhancing the generalization capability. These techniques resulted in more accurate predictions, particularly for small objects, addressing a notable gap in the application of object detection algorithms to conservation challenges.

While boosting algorithms like AdaBoost provide substantial performance improvements, their computational cost should be considered, especially for real-time applications. Future work can explore other boosting techniques, such as XGBoost or Gradient Boosting, to further optimize the model's performance without excessively increasing computational demands. Hybrid approaches that combine ensemble learning with architectural innovations in deep learning could yield even more efficient models for endangered species detection and other complex image classification tasks.