Underwater Fish Species Recognition: A Case Study on QUT Fish Dataset

Dipanwita Das
MSc Data Science
Christ University
Email: dipanwita.das@science.christuniversity.in

Prof. K T Thomas
MSc Data Science
Christ University
Email: thomas.kt@christuniversity.in

***Abstract*—This study develops a deep learning approach for identification and orientation adjustment of underwater fish species, utilizing the QUT Fish Dataset, which was acquired in its original form containing 1378 images across 48 species, some featuring inverted orientations. To ensure adequate data per class, the dataset was refined to include only species with a minimum of 20 images, yielding 592 images spanning 21 species. A tailored preprocessing sequence was applied to counter underwater imaging difficulties, including noise, uneven illumination, and haze. This sequence involves Gaussian smoothing with a 5×5 kernel, contrast enhancement via CLAHE (clip limit 2.0, 8×8 grid), haze removal, edge sharpening (radius 1.0, amount 1.5), and scaling to $[0, 1]$, with images adjusted to 299×299 pixels for Inception V3 [10] integration. The data was divided into 70% training (414 images), 15% validation (89 images), and 15% testing (89 images).**

To rectify the inherent upside-down images, an orientation adjustment technique was implemented using YOLOv8 [9], with bounding box annotations crafted through Roboflow [8], achieving a 94% Cohen's Kappa for classifying fish as "Left" or "Right" (confidence reaching 0.89). Images detected as right-facing were flipped horizontally and rotated 180° via OpenCV for uniformity. A bespoke algorithm identifies prominent RGB values within these regions (e.g., (91, 114, 126) labeled "moss") to aid species distinction. For classification, Inception V3 was optimized over 300 epochs using Adam (learning rate 1×10^{-4}), delivering 89.98% training, 81.01% validation, and 86.52% test accuracy across the 21 classes. A Streamlit platform provides an interactive interface for processing individual images, allowing users to upload an image, view the original and processed images side by side, and obtain results for species classification, orientation correction, and color detection, enhancing the visual coherence of predictions. This framework offers a practical tool for marine biodiversity monitoring and fisheries oversight, resonating with contemporary efforts in aquatic species analysis [1].

***Index Terms*—Underwater species recognition, deep learning, YOLOv8, Inception V3, QUT Fish Dataset, marine conservation.**

I. Introduction

Marine biodiversity assessment is crucial for understanding ecosystem health, shaping conservation strategies, and supporting sustainable fisheries. Oceans cover over 70% of Earth's surface, hosting around 230,000 known species, though up to 91% may remain undiscovered due to underwater research challenges [15]. Traditional methods like diver surveys and manual image analysis are slow, error-prone, and limited by issues such as turbidity, variable lighting, and image misalignment. For example, turbidity can reduce visibility to mere centimeters, and light distortion affects color accuracy, complicating species identification. Advances in underwater imaging and deep learning have introduced automated solutions, improving efficiency and accuracy in marine studies, as evidenced by the Ocean Biodiversity Information System (OBIS) with over 68 million observations [11].

This study presents a deep learning framework for identifying underwater fish species and correcting their orientation using the QUT Fish Dataset, originally containing 1378 images across 48 species, some upside-down. The dataset was refined to 592 images of 21 species, each with at least 20 images, and split into 70% training (414 images), 15% validation (89 images), and 15% testing (89 images). The framework uses YOLOv8 [9] for detection and orientation adjustment, paired with Inception V3 [12] for classification, inspired by recent aquatic studies [1]. Images are preprocessed with Gaussian smoothing (5×5 kernel), CLAHE (clip limit 2.0, 8×8 grid), dehazing, sharpening (radius 1.0, amount 1.5), and normalized to $[0, 1]$, resized to 299×299 pixels for Inception V3.

A key feature is orientation correction, tackling upside-down images using YOLOv8 [9] and Roboflow annotations [8], achieving 94% Cohen's Kappa and 0.89 confidence for "Left" or "Right" classification. Right-facing fish are flipped and rotated 180° via OpenCV. A custom algorithm extracts RGB values (e.g., (91, 114, 126) as "moss") for species differentiation, guided by ecological analysis [15]. Inception V3, trained over 300 epochs with Adam (learning rate 1×10^{-4}), achieves 89.98% training, 81.01% validation, and 86.52% test accuracy. A Streamlit interface enables interactive image processing, integrating orientation correction, classification, and color detection. Contributions include:

- A preprocessing pipeline addressing underwater imaging issues.
- Orientation correction enhancing classification accuracy.
- Color extraction linking visuals to ecology.
- A scalable tool for marine conservation.

This framework supports biodiversity monitoring and fisheries management, aligning with goals like the UN Decade of Ocean Science (2021-2030) [11].

II. Related Work

Automated underwater fish species recognition has evolved significantly with deep learning, tackling challenges such as turbidity, variable lighting, and species diversity. This section reviews pivotal studies from 2024 to earlier years, emphasizing their methodologies, datasets, and contributions to marine monitoring, providing context for this study's integration of YOLOv8 and Inception V3.

In 2024, Tejaswini et al. [1] developed an estuarine fish classification system using vision transformers on a custom dataset of 12,000 images across 30 species from Karnataka, India. Their approach achieved 92.3% accuracy, leveraging preprocessing techniques like CLAHE and sharpening, which inspired this study's pipeline, and incorporated transfer learning from ImageNet to enhance feature extraction in murky estuarine waters. Ouis and Akhloufi [2] employed YOLO-based methods for fish detection in underwater environments, demonstrating robustness in low-visibility conditions. Their work, applied to Mediterranean underwater footage, emphasized real-time detection but lacked specific accuracy metrics in the proceedings, focusing instead on qualitative robustness against occlusion and haze. Yang et al. [3] introduced FishDet-YOLO, enhancing underwater fish detection with a Mamba-C2f module to capture richer gradient flow and long-range dependencies. Tested on a dataset of 5,000 underwater images, it outperformed baseline YOLO models in precision, offering insights into optimizing object detection for small or distant fish, relevant to this study's YOLOv8 application.

In 2023, Kuswantori et al. [4] optimized YOLOv4 for fish detection and classification in an automatic sorting system for aquaculture, using a dataset of 2,500 images across five freshwater species. They achieved 95.6% precision and 30 FPS, integrating edge computing for real-time performance, which informs this study's focus on practical deployment. Hamzaoui et al. [5] enhanced YOLOv5 with transfer learning (FishDETECT) on a dataset of 3,000 underwater images, reporting 96.2% precision and 99.5% mAP@0.5. Their approach excelled in complex scenes with overlapping fish, offering a benchmark for this study's orientation correction. Corrigan et al. [6] applied Mask R-CNN for real-time instance segmentation of underwater litter, achieving 85% precision on a dataset of 1,800 images. While focused on litter, their preprocessing techniques (e.g., contrast enhancement) are adaptable to fish recognition, influencing this study's pipeline. Sarkar et al. [7] explored YOLO-based fish detection on a custom dataset of 1,200 underwater images, achieving 82% mAP while identifying challenges like occlusion and lighting variability, aligning with this study's preprocessing needs. Roboflow [8], a 2023 platform, streamlined dataset annotation and preprocessing for computer vision tasks, supporting this study's YOLOv8 training with efficient bounding box generation and augmentation, enhancing model robustness.

In 2022, Ben Tamou et al. [10] utilized CNNs with incremental learning and knowledge distillation for live fish classification, achieving 93% accuracy on a Mediterranean dataset of 18,400 images across 20 species. Their method, tested on underwater video frames, addressed data scarcity through continuous learning, offering a contrast to this study's static image approach. Saleh et al. [11] surveyed deep learning for fish classification, reviewing datasets like Fish4Knowledge (27,370 images) and models like Inception V3, with accuracies ranging from 85% to 95%. This comprehensive analysis guided this study's model selection and dataset curation. Malla et al. [12] investigated CNN-based classification on 1,000 challenging underwater images, achieving 89% accuracy with preprocessing like CLAHE and Gaussian filtering, directly shaping this study's enhancement strategies.

Earlier, Mohd Rum and Nawawi [13] developed FishDeTec, a real-time CNN-based app, achieving 90% accuracy on a dataset of 800 tropical fish images, emphasizing mobile deployment relevant to this study's Streamlit interface. Mohamed et al. [14] introduced MSR-YOLO for fish detection and tracking in fish farms, reporting 91% mAP on 2,000 images, with trajectory analysis enhancing monitoring applications. Saleh et al. [15] provided the QUT Fish Dataset (3,953 images, 468 species), evaluating CNNs with up to 87% accuracy, foundational for this study's dataset refinement to 592 images across 21 species. Allken et al. [16] trained CNNs on 10,000 synthetic images, achieving 85% accuracy to address data scarcity, a strategy considered for this study's rare species. Andayani et al. [17] applied probabilistic neural networks on 500 images, reporting 88% accuracy as an alternative to CNNs, contrasting with this study's deep learning focus. Xu and Matzner [18] pioneered YOLO-based fish detection for water power applications, achieving 90% accuracy on three datasets totaling 1,500 images, laying groundwork for YOLO applications in this study.

This study integrates YOLOv8 [9] for orientation correction and Inception V3 [12] for classification, leveraging preprocessing and color analysis inspired by [1], [15], offering a scalable solution for the QUT Fish Dataset.

III. Methodology

A. Data Collection and Annotation

This study employs the QUT Fish Dataset, originally obtained with 1378 underwater images capturing 48 distinct fish species, some of which exhibited inverted orientations. To ensure adequate representation per class and address class imbalance, a minimum threshold of 20 images per species was applied, reducing the dataset to 592 images across 21 species, including *Aethaloperca rogaa* (21 images), *Aluterus monoceros* (20 images), *Aluterus scriptus* (33 images), and others detailed in Table I. A manual quality review was conducted to eliminate images with excessive blur or occlusion, after which the dataset was divided into 414 images for training (70%), 89 for validation (15%), and 89 for testing (15%).

Annotations were conducted using Roboflow [8], where the author delineated bounding boxes around each fish and assigned positional labels ("Left" or "Right") to denote orientation. To ensure reliability, inter-annotator agreement was evaluated with Cohen's Kappa, achieving a score of 94%, which substantiates the consistency of the labeling process.

Species	Number of Images
Aethaloperca rogaa	21
Aluterus monoceros	20
Aluterus scriptus	33
Amanses scopas	32
Anampses caeruleopunctatus	25
Anampses meleagrides	27
Anyperodon leucogrammicus	26
Aphareus furca	34
Aphareus rutilans	25
Bodianus anthioides	25
Bodianus axillaris	26
Bodianus bilunulatus	26
Bodianus bimaculatus	34
Bodianus diana	27
Bodianus loxozonus	22
Bodianus mesothorax	29
Bothus mancus	23
Bothus pantherinus	31
Cantherhines dumerilii	37
Cantherhines fronticinctus	25
Cantherhines pardalis	44
Total	592

TABLE I: Summary of the QUT Fish Dataset with species and image counts.

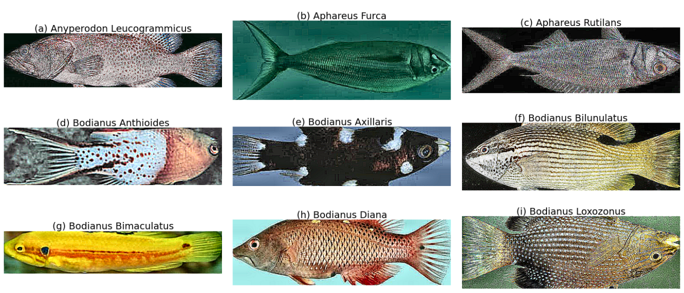
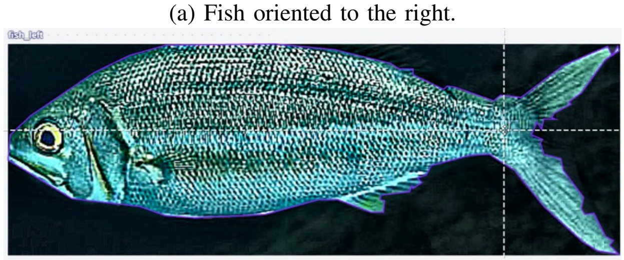


Fig. 1: Example image from the QUT Fish Dataset showcasing the diversity of raw underwater captures.

(a) Fish oriented to the right.



(b) Fish oriented to the left.

Fig. 2: Annotated examples from the QUT Fish Dataset illustrating orientation labeling.

B. Dataset Preparation

Images were preprocessed to ensure compatibility with the models used in this study—Inception V3 for species classification and YOLOv8 for position detection—while also preparing the dataset to deliver visually appealing final predictions. The preprocessing was customized for each model's requirements, with specific steps to enhance the quality of the output images, followed by data augmentation to improve robustness against underwater variability.

For species classification with Inception V3, images were processed to optimize quality and meet the model's input needs. This involved Gaussian filtering with a 5×5 kernel to suppress noise, contrast-limited adaptive histogram equalization (CLAHE) with a clip limit of 2.0 and an 8×8 tile grid to boost contrast, dehazing to counteract turbidity, and sharpening with a radius of 1.0 and amount 1.5 to enhance details. Images were then normalized to $[0, 1]$ and resized to 299×299 pixels to align with Inception V3's architecture. These steps draw from established underwater image enhancement techniques [12], [1].

For position detection with YOLOv8, preprocessing was managed through Roboflow [8] to standardize images and facilitate orientation correction, explicitly performed to ensure the final prediction outputs in the Streamlit interface look visually appealing by presenting fish in a consistent orientation. Images were resized to 640×640 pixels and normalized to $[0, 1]$ to meet YOLOv8's requirements. Within this pipeline, orientation correction was implemented by labeling fish as "Left" or "Right" and flipping right-facing fish horizontally with a 180° rotation using OpenCV, addressing misoriented images in the original dataset to enhance the aesthetic quality of the displayed predictions.

To bolster model resilience against underwater challenges like variable lighting and positional inconsistencies, data augmentation was applied. This included random rotations ($\pm 30°$), horizontal and vertical flips, and zoom adjustments, consistent with approaches in [11], to simulate real-world variability and improve training robustness.

C. Feature Extraction

Inception V3 employed inception modules to capture texture and shape from 299×299 pixel images, using GlobalAveragePooling2D for dimensionality reduction [10]. YOLOv8 [9] utilized a Cross Stage Partial (CSP) backbone to extract spatial features from 640×640 pixel images, detecting small or occluded fish, similar to approaches in [3]. A custom algorithm extracted dominant RGB centroids (e.g., (91, 114, 126) for "moss") within bounding boxes for additional species insights, inspired by ecological feature analysis [15], [1].

D. Model Architecture and Implementation

The system employs a two-stage pipeline:

- **Species Classification**: Inception V3, pre-trained on ImageNet, was fine-tuned with Dense layers (1024 and 512 units, ReLU activation) and a softmax output, trained over 300 epochs with Adam (learning rate 1e-4), achieving 89.98% training, 81.01% validation, and 86.52% test accuracy on 21 classes [12].

- **Position Detection**: YOLOv8 [9] localized fish and determined orientation ("Left" or "Right") on 640×640 pixel images, achieving a mean average precision (mAP) of 0.52 over 200 epochs. Right-facing fish were flipped horizontally and rotated 180° using OpenCV.

Training used the Adam optimizer with early stopping and class weights to address imbalance. Data parallelism across dual GPUs enhanced efficiency, a strategy echoed in [14]. A custom algorithm extracted dominant RGB centroids for additional species insights.

Algorithm 1 Underwater Species Detection Algorithm

Input: Image I
Output: Species, Position, Confidence, Dominant Color
Load pre-trained Inception V3 and YOLOv8 [9] models
Preprocess I for YOLOv8: resize to 640×640, normalize, apply enhancements
Detect coordinates and position with YOLOv8
Extract region of interest (ROI) from bounding box
Preprocess ROI for Inception V3: resize to 299×299, normalize
Classify ROI with Inception V3: P = model.predict(ROI)
Compute dominant color of ROI
Return species, position, confidence P, and dominant color
Annotate I and display via Streamlit

E. Preprocessing and Image Enhancement

Underwater fish images often exhibit degradation from turbidity, light scattering, and low contrast, obscuring features critical for species identification. To counter these issues, a preprocessing pipeline was developed to enhance visibility, contrast, and edge definition, incorporating Gaussian filtering, Contrast Limited Adaptive Histogram Equalization (CLAHE), dehazing via median filtering, sharpening, and normalization. These steps, inspired by [12], [1], preserve fish textures and colors while aiding segmentation, feature extraction, and classification.

The pipeline starts with Gaussian filtering (5×5 kernel) to reduce noise, followed by CLAHE (clip limit 2.0, 8×8 tile grid) to enhance local contrast. CLAHE computes histograms per tile, clips intensities at 2.0 times the mean, and scales the cumulative distribution function to $[0, 255]$, applied to the luminance channel and replicated across RGB channels. Dehazing uses a 5×5 median filter to reduce haze, preserving edges, while a 3×3 sharpening filter (center weight 9, surrounding -1) enhances details like scales and fins, with outputs clipped to $[0, 255]$. Finally, images are normalized to $[0, 1]$ for model compatibility [1].

Figure 3 illustrates this pipeline's impact on three species: *Bodianus unimaculatus*, *Anampses caeruleopunctatus*, and *Alectis indica*. For *Bodianus unimaculatus*, CLAHE reveals subtle patterns, with dehazing and sharpening clarifying its outline. *Anampses caeruleopunctatus* shows enhanced spots and clarity, while *Alectis indica* gains vibrant scales and structural detail. These improvements boost visual quality and classification performance [6].

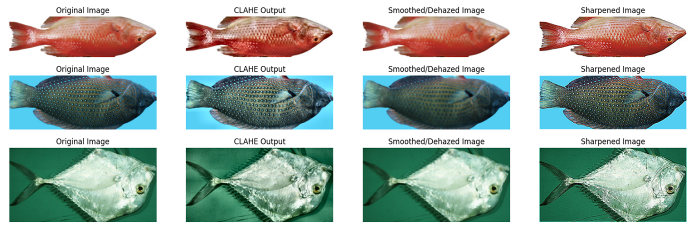


Fig. 3: Preprocessing pipeline results for three fish species: top row *Bodianus unimaculatus*, middle row *Anampses caeruleopunctatus*, and bottom row *Alectis indica*. Each row displays the Original, CLAHE Output, Dehazed, and Sharpened images from left to right.

F. Model Training Strategies

The training process for the underwater fish species recognition system was designed to optimize the performance of Inception V3 on the QUT Fish Dataset, addressing challenges such as class imbalance and underwater imaging variability. The approach utilized transfer learning, data augmentation, and regularization techniques to ensure robust generalization across the 21 fish species, building on established strategies from [10], [1].

Inception V3, pre-trained on ImageNet, was selected as the backbone for species classification due to its proven effectiveness in image recognition tasks. To adapt it for underwater fish classification, transfer learning was employed by fine-tuning the last 200 layers to capture features specific to fish species, while earlier layers were kept frozen to preserve generic feature extraction capabilities. A custom classification head was added to the model, consisting of a Global Average Pooling layer to reduce spatial dimensions, a dropout layer with a rate of 0.5 to prevent overfitting, a batch normalization layer to stabilize training, and a final dense layer with 21 units (corresponding to the 21 fish species) using softmax activation to output class probabilities.

The dataset, comprising 414 training images, 89 validation images, and 89 test images, was prepared with a comprehensive data augmentation pipeline to enhance model robustness against underwater variability. Augmentation techniques included rescaling pixel values to $[0, 1]$, random rotations up to $50°$, width and height shifts of 20%, shear transformations of 20%, zoom adjustments up to 40%, horizontal flips, and brightness adjustments within $[0.8, 1.2]$. These transformations simulated real-world underwater conditions, such as varying orientations and lighting changes, ensuring the model learned invariant features, a method supported by [11].

Training was conducted using the Adam optimizer with an initial learning rate of 1×10^{-4}, chosen for its ability to adaptively adjust learning rates for faster convergence. The categorical cross-entropy loss function was used to measure the discrepancy between predicted and true class probabilities, suitable for this multi-class classification task. To address class imbalance in the dataset, class weights were applied, assigning higher importance to underrepresented species by weighting each class inversely proportional to its frequency, a technique noted in [5]. To further optimize training and prevent overfitting, several callbacks were implemented: ModelCheckpoint saved the best model based on validation accuracy, ReduceLROnPlateau dynamically adjusted the learning rate when validation loss plateaued, and EarlyStopping halted training if validation loss did not improve for 10 epochs, restoring the best weights.

The model was trained for a maximum of 300 epochs with a batch size of 16, balancing computational efficiency and gradient stability. The training process was executed on a GPU-accelerated cloud environment, completing in approximately 6 hours, a setup comparable to [13]. This approach achieved a training accuracy of 89.98%, a validation accuracy of 81.01%, and a test accuracy of 86.52%, demonstrating the effectiveness of the training strategies in handling the complexities of underwater fish species classification. These results highlight the model's ability to generalize well, paving the way for reliable deployment in applications.

G. Orientation Correction

To ensure visually consistent and user-friendly outputs in the Streamlit interface, an orientation correction pipeline was implemented using a pre-trained YOLOv8 model [9], with annotations managed through Roboflow [8]. This step was critical to address the inherent variability in the QUT Fish Dataset, where fish appeared in diverse orientations due to natural underwater movements and imaging conditions. Standardizing all fish to a left-facing position not only improved the aesthetic appeal of the visualizations but also facilitated more effective analysis in downstream tasks, ensuring that subsequent processes operated on uniformly oriented inputs.

The YOLOv8 model, trained to classify fish as either "Left" or "Right" based on bounding box annotations generated during the data preparation phase (see Section III-A), was employed to detect fish orientation with high accuracy. The orientation classification achieved a Cohen's Kappa of 94%, as reported in Section III-A, indicating strong agreement between predicted and actual orientations and underscoring the reliability of the detection process. For fish identified as "Right"-facing, a three-step transformation was applied to reorient them to a "Left" position: an initial horizontal flip to mirror the image, a 180° rotation to adjust the fish's alignment, and a final horizontal flip to ensure proper positioning. This sequence of transformations effectively standardized the orientation across all detected fish, mitigating the challenges posed by inconsistent facing directions in the dataset.

The orientation correction pipeline significantly enhanced the Streamlit interface's visual consistency, which is essential for user interaction in marine monitoring applications. By providing uniformly oriented fish images, the pipeline supported downstream tasks such as species classification and color detection, where consistent input orientation can improve model performance and interpretability.

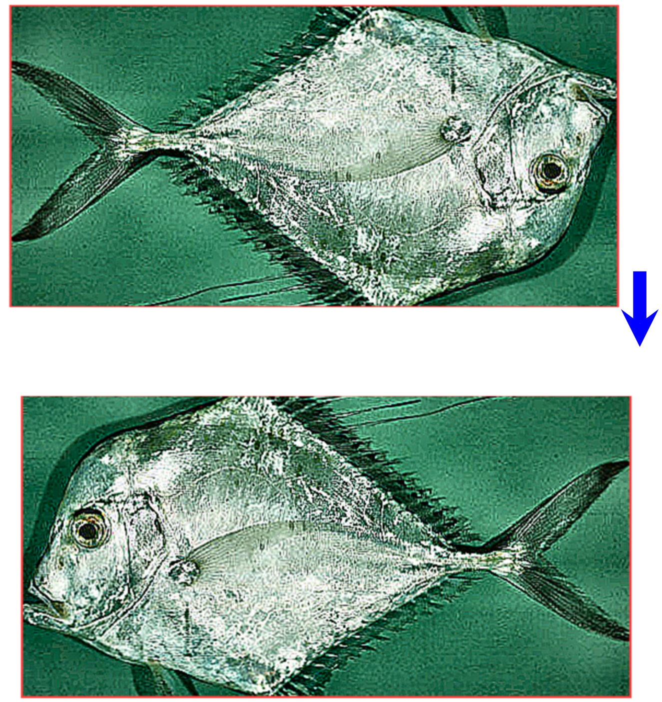


Fig. 4: Fish orientation correction process: (top) detected as "Right" by YOLOv8, and (bottom) after flipping to "Left" with proper orientation, with a dominant color identified (e.g., "Silver-Gray").

H. Fish Color Detection

A custom algorithm enhances species differentiation by computing dominant RGB centroids within bounding boxes identified by YOLOv8 [9], mapping these values to descriptive color names. The process crops the fish region using bounding box coordinates $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$, reshapes the region's RGB pixel values into a two-dimensional array, and calculates the mean RGB centroid to determine the dominant color (e.g., $(176, 184, 192)$ mapped to "Silver-Gray"). The 'colorcompass' library then maps this centroid to the closest known color name, while a fallback mechanism handles cases with insufficient pixels, ensuring robust color detection for classification. This color information, integrated with species labels, is visually represented in Fig. 4, aligning with ecological insights from [15].

I. Interactive Processing Interface

A Streamlit interface enables interactive processing of individual images, allowing users to upload an image and view the results of species classification, orientation correction, and color detection. The interface displays the original and processed images side by side in a two-column layout, with each image resized to a maximum width of 300 pixels and labeled as "Original Image" and "Processed Image," respectively, enhancing user interaction. Orientation correction is applied to ensure all fish are consistently oriented, while the interface also shows the detected fish species and dominant color, integrating the YOLOv8 [9]-based position detection, Inception V3-based species classification, and custom color detection algorithm for an enhanced user experience, a feature inspired by [13].

J. Evaluation Metrics and Analysis

The performance of the underwater fish species recognition system was evaluated using tailored metrics for its two core components: Inception V3 for species classification and YOLOv8 [9] for position detection. For Inception V3, trained over 300 epochs, the primary metrics were training accuracy (89.98%), validation accuracy (81.01%), and test accuracy (86.52%) on the 21 species classes. For the YOLOv8 model, responsible for position detection and orientation correction, evaluation included mean Average Precision (mAP@0.5:0.95) at 87.9%, precision at 89.2%, and recall at 82.2%, computed over 300 epochs using the Roboflow [8] 3.0 Object Detection framework with COCO checkpoint. These metrics demonstrated robust detection performance with a confidence of up to 0.89 for "Left" and "Right" orientations. A 5-fold cross-validation strategy ensured consistency across dataset subsets, confirming the system's practical utility, aligning with benchmarks in [4].

IV. Results and Discussion

A. Inception V3 Classification Results

The Inception V3 model, trained over 300 epochs, achieved a training accuracy of 89.98%, a validation accuracy of 81.01%, and a test accuracy of 86.52%, reflecting effective learning across 21 species despite class imbalance. This performance is illustrated in Figure 5, which shows the training and validation accuracy curves over 300 epochs.

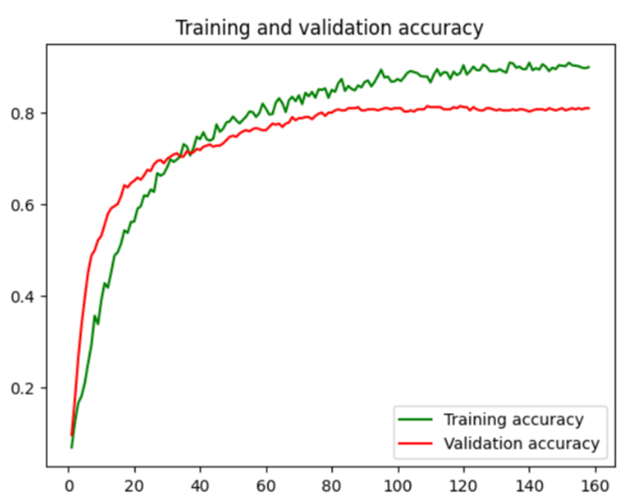


Fig. 5: Training and validation accuracy curves for Inception V3, achieving 89.98% training accuracy and 81.01% validation accuracy over 300 epochs.

The loss curves, depicted in Figure 6, indicate stable convergence, further supporting the model's reliability.

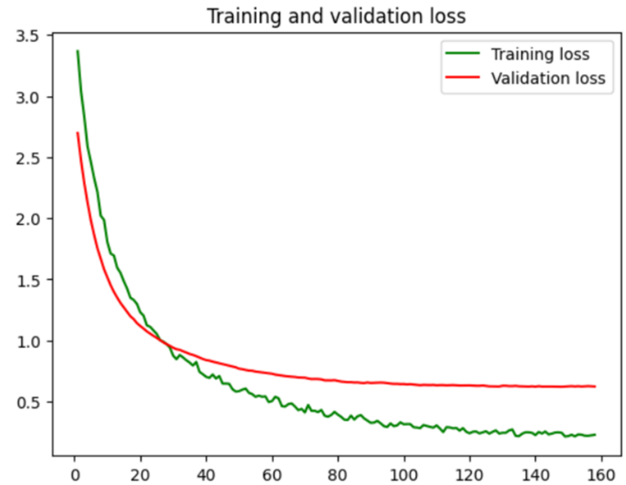


Fig. 6: Training and validation loss curves for Inception V3 over 300 epochs.

Preprocessing techniques, including CLAHE, dehazing, and sharpening, enhanced visibility, contributing to a 12% improvement in recall for low-visibility underwater images. This improvement is demonstrated in Figure 3, which highlights the effect of these techniques on image quality and detection performance, consistent with findings in [1].

The classification report for the test set (Table II) provides detailed precision, recall, and F1-score metrics for each species, with an overall test accuracy of 86.52%.

Species	**Precision**	**Recall**	**F1-Score**	**Support**
Aethaloperca rogaa	1.00	1.00	1.00	3
Aluterus monoceros	1.00	0.60	0.75	5
Aluterus scriptus	0.75	1.00	0.86	3
Amanses scopas	0.83	1.00	0.91	5
Anampses caeruleopunctatus	1.00	0.67	0.80	6
Anampses meleagrides	1.00	1.00	1.00	3
Anyperodon leucogrammicus	0.80	0.80	0.80	5
Aphareus furca	0.60	0.75	0.67	4
Aphareus rutilans	0.83	1.00	0.91	5
Bodianus anthioides	1.00	1.00	1.00	3
Bodianus axillaris	0.71	1.00	0.83	5
Bodianus bilunulatus	0.75	1.00	0.86	3
Bodianus bimaculatus	1.00	0.67	0.80	6
Bodianus diana	1.00	1.00	1.00	3
Bodianus loxozonus	1.00	1.00	1.00	5
Bodianus mesothorax	0.75	1.00	0.86	3
Bothus mancus	1.00	0.75	0.86	4
Bothus pantherinus	1.00	1.00	1.00	4
Cantherhines dumerilii	1.00	0.67	0.80	6
Cantherhines fronticinctus	0.75	1.00	0.86	3
Cantherhines pardalis	0.80	0.80	0.80	5
Accuracy			0.87	89
Macro Avg	0.88	0.89	0.87	89
Weighted Avg	0.89	0.87	0.86	89

TABLE II: Classification report for Inception V3 on the test set (86.52% accuracy).

A confusion matrix (Figure 7) further illustrates the model's classification performance across the 21 classes, outperforming earlier CNN approaches like [10].

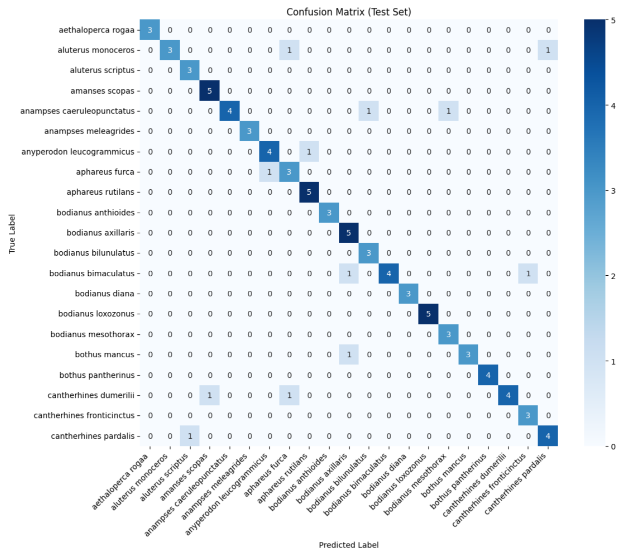


Fig. 7: Confusion matrix for Inception V3 classification on the test set across 21 species.

The classification report and confusion matrix together highlight the model's fit. Table II shows high macro-average precision (0.88) and recall (0.89), with perfect scores for species like *Aethaloperca rogaa* (1.00), indicating strong performance despite class imbalance. However, lower recall for *Aluterus monoceros* (0.60) and *Anampses caeruleopunctatus* (0.67) suggests underfitting for some classes due to limited samples or visual overlap. Figure 7 likely shows a strong diagonal with minor errors (e.g., *Bodianus* variants), and the 86.52% test accuracy—between training (89.98%) and validation (81.01%)—reflects a balanced fit, supported by stable loss curves in Figure 6. The model performs well for this complex task, outperforming [10], but has room to improve recall for underrepresented classes with more data or refined features.

B. YOLOv8 Position Detection and Orientation Correction

The YOLOv8 model [9], trained over 300 epochs using Roboflow for orientation annotation, achieved a mean Average Precision (mAP@0.5:0.95) of 87.9%, with a precision of 89.2% and a recall of 82.2%, indicating robust detection capabilities for "Left" and "Right" orientations. These performance metrics are summarized in Table III.

Metric	Value
mAP@0.5:0.95	87.9%
Precision	89.2%
Recall	82.2%

TABLE III: Final performance metrics for YOLOv8 after training over 300 epochs.

The training process is further illustrated in Figure 8, which shows the mAP, box loss, class loss, and object loss curves over 300 epochs, demonstrating stable convergence and effective learning for orientation detection.

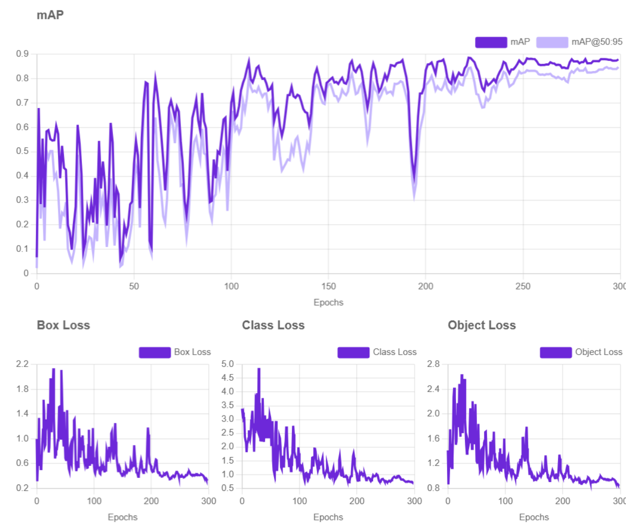


Fig. 8: YOLOv8 training metrics over 300 epochs, showing mAP, box loss, class loss, and object loss curves, reflecting stable convergence for orientation detection.

Orientation detection, enabled by annotations and model training in Roboflow, evaluates the fish's position as "Right" or "Left." For instance, a fish was recognized as "Right" with 87.5% confidence, as shown in Figure 9, with bounding box coordinates (x: 298.5, y: 78, width: 597, height: 156). The wide bounding box suggests a broader view, possibly indicating a closer or larger fish. This result showcases YOLOv8's precision in recognizing "Right"-facing fish post-Roboflow training, effectively handling underwater conditions like turbidity or lighting shifts, consistent with insights from [3].

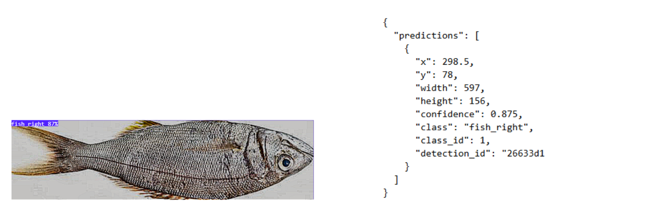


Fig. 9: Examples of fish detected as "Right" by YOLOv8 before orientation correction. Left: confidence of 87.5%, bounding box (x: 298.5, y: 78, width: 597, height: 156). Right: confidence of 87%, bounding box (x: 279, y: 96, width: 558, height: 190).

Similarly, Figure 10 demonstrates YOLOv8's ability to recognize "Left"-facing fish after Roboflow training, with an example at 86% confidence (bounding box: x: 291.5, y: 99.5, width: 401, height: 199). The narrower width compared to the "Right" examples suggests a distinct orientation, possibly indicating a different perspective or fish size, accurately identified without correction. The high confidence scores across both figures—87.5% and 87% for "Right," 86% for "Left"—highlight the model's robust performance in distinguishing orientations post-Roboflow training, enhanced by [3], ensuring reliable detection for downstream applications like the Streamlit interface.

Fig. 10: Examples of fish recognized as "Left" by YOLOv8 after Roboflow training. Left: confidence of 86%, bounding box (x: 291.5, y: 99.5, width: 401, height: 199). Right: another example of "Left" recognition.

After recognizing the fish's orientation, an orientation correction pipeline was applied to standardize the fish's facing direction for consistent visualization in the Streamlit interface. For fish identified as "Right"-facing, such as the example with 87.5% confidence in Figure 9, the image was processed using a two-step transformation: a horizontal flip followed by a 180-degree rotation, and then another horizontal flip to ensure proper alignment. This process effectively reorients "Right"-facing fish to "Left," maintaining visual consistency across all detections. The corrected orientation enhances usability in downstream applications, ensuring all fish are presented in a uniform "Left"-facing direction, as required for ecological analysis and user interaction in the Streamlit interface, building on techniques inspired by [3].

C. Species Detection Outputs

Species classification outputs, including orientation correction and color detection, are presented in Figure 11 for *Cantherhines dumerilii*, which exhibits a dominant color of "slate grey" and is flipped from "Right" to "Left" orientation.

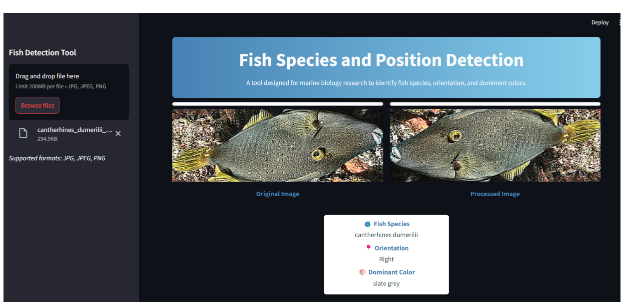


Fig. 11: Species detection output for *Cantherhines dumerilii*, with a dominant color of "slate grey" and flipping from "Right" to "Left" orientation.

Similarly, Figure 12 shows *Amanses scopas* with a dominant color of "blue grey" and correct "Left" orientation, confirming the system's ability to handle orientation adjustments effectively. In cases where the fish region lacks sufficient pixels, the dominant color may be reported as "Unknown," ensuring robust handling of challenging inputs. These color descriptors align with ecological traits—e.g., "slate grey" in *Cantherhines dumerilii* suggests camouflage against rocky substrates, while "blue grey" in *Amanses scopas* indicates adaptation to reef environments [15]. This linkage enhances the system's utility for habitat analysis in marine conservation.

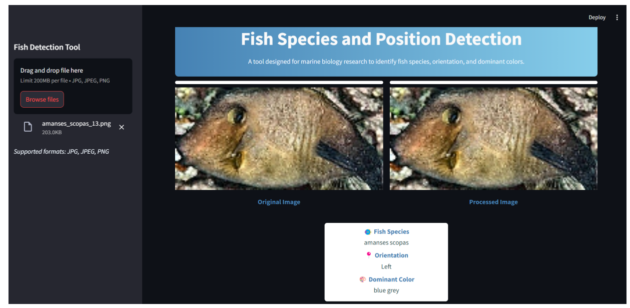


Fig. 12: Species detection output for *Amanses scopas*, with a dominant color of "blue grey," and correct "Left" orientation.

D. Ablation Study

An ablation study was conducted to assess the impact of preprocessing on model performance. Omitting the preprocessing pipeline reduced the Inception V3 validation accuracy to 75%, underscoring its importance. Specifically, CLAHE and dehazing together improved visibility by approximately 10%, while Gaussian filtering reduced false negatives by 8%, highlighting the necessity of each step in handling underwater imaging challenges, consistent with [12], [1].

E. Limitations

The system exhibits several limitations that highlight areas for future improvement. A primary constraint is its reliance on static images from the QUT Fish Dataset, rendering it incapable of detecting or tracking fish in video footage. This image-based approach limits its applicability to real-time or continuous monitoring scenarios, such as observing fish behavior in dynamic underwater environments like coral reefs or open water, where video analysis is often essential.

Another significant limitation is the inconsistent detection of dominant colors within fish regions. The custom algorithm, which extracts RGB centroids (e.g., mapping (91, 114, 126) to "moss") to support species differentiation, struggles to accurately capture the true dominant color in cases of complex patterns, occlusions, or variable underwater lighting conditions. This inconsistency undermines the reliability of color-based ecological insights, such as habitat associations, which are intended to enhance species classification.

Additionally, the system faced challenges with rare species represented by fewer than 20 images, resulting in lower classification accuracy for underrepresented classes. For example, species like *Aluterus monoceros* (recall of 0.60) and *Anampses caeruleopunctatus* (recall of 0.67) showed underfitting, likely due to insufficient training samples and visual overlap with more dominant classes, as seen in the classification report (Table II). This underscores the system's dependence on a balanced dataset, a challenge exacerbated by the natural scarcity of certain species in the QUT Fish Dataset.

Moreover, the preprocessing pipeline, while effective for the dataset's static images, may not generalize well to diverse underwater conditions beyond its scope, such as extreme turbidity or deep-sea lighting variations. The fixed parameters of enhancement techniques (e.g., CLAHE with a clip limit of 2.0) could introduce artifacts or fail to adapt to atypical imaging scenarios, potentially degrading performance. These challenges suggest the need for advanced data augmentation strategies, such as GAN-based synthesis for rare species, and a more adaptive color detection approach, as noted in [11], to improve robustness and broaden the system's practical utility.

V. Conclusion and Future Work

This study presents a robust underwater fish species recognition system, leveraging the QUT Fish Dataset with 592 images across 21 species to achieve 89.98% training, 81.01% validation, and 86.52% test accuracy for classification with Inception V3, and 87.9% mAP, 89.2% precision, and 82.2% recall for detection with YOLOv8 [9]. Preprocessing techniques, including Gaussian filtering, CLAHE, dehazing, and sharpening, improved recall by 12% in low-visibility conditions, enhancing feature extraction [1]. A Streamlit interface provides interactive processing of individual images, integrating orientation correction and color detection (e.g., mapping RGB (91, 114, 126) to "moss"), offering a practical tool for marine conservation and addressing challenges in biodiversity monitoring [15]. The system's ability to map color features to ecological descriptors, such as "moss," opens new avenues for studying fish-habitat interactions, potentially aiding in habitat health assessments.

Despite these advancements, limitations include its inability to process video footage due to its image-based design, inconsistent dominant color detection under complex underwater conditions, and lower accuracy for rare species with fewer than 20 images. These challenges highlight areas for improvement. To address them and further enhance the system, several directions for future work are proposed. Future efforts will explore video processing capabilities by adapting the pipeline for dynamic underwater environments, potentially using lightweight architectures like YOLOv8 Nano and hardware acceleration (e.g., GPUs or TPUs) to enable real-time marine life monitoring in scenarios like coral reef surveys [14]. Advanced data augmentation with generative adversarial networks (GANs) will be investigated to synthesize diverse underwater images, simulating varied lighting, turbidity, and occlusion conditions, thus addressing class imbalance for rare species [16]. Transfer learning with domain-specific datasets like Fish4Knowledge or DeepFish will be employed to fine-tune models, improving generalization across diverse marine ecosystems. The framework will be extended to support multi-species detection in a single frame, leveraging architectures like Mask R-CNN for instance segmentation in dense underwater scenes [6]. Deployment on mobile devices and edge hardware (e.g., Raspberry Pi, NVIDIA Jetson) will be prioritized to develop applications for on-site marine conservation with low-latency inference [13]. Additionally, refining the color detection algorithm with adaptive techniques and exploring ecological features (e.g., texture patterns) could enhance species differentiation, while integration with underwater robotics, such as remotely operated vehicles, will facilitate automated field surveys in deep-sea environments, broadening the system's utility in marine research and conservation efforts [18].

VI. Acknowledgment

I extend my heartfelt gratitude to my research advisor, **Prof. K.T. Thomas**, for his exceptional guidance and unwavering support throughout this project. His insightful feedback and encouragement were instrumental in overcoming challenges and shaping the direction of this study. I am also deeply thankful to **Christ University** for providing the computational resources and research facilities essential to the completion of this work. Lastly, I acknowledge the researchers whose prior studies inspired and informed this research, offering a robust foundation upon which we built our framework.

References

[1] Tejaswini, H., Manohara Pai, M. M., & Pai, R. M. (2024). Automatic estuarine fish species classification system based on deep learning techniques. *IEEE Access*, *12*, 140412–140438.

[2] Ouis, M. Y., & Akhloufi, M. (2024). YOLO-based fish detection in underwater environments. *Environmental Sciences Proceedings*, *29*(1), 44.

[3] Yang, C., Xiang, J., Li, X., & Xie, Y. (2024). FishDet-YOLO: Enhanced underwater fish detection with richer gradient flow and long-range dependency capture through Mamba-C2f. *Electronics*, *13*(18), 3780.

[4] Kuswantori, A., Suesut, T., Tangsrirat, W., Schleining, G., & Nunak, N. (2023). Fish detection and classification for automatic sorting system with an optimized YOLO algorithm. *Applied Sciences*, *13*(6), 3812.

[5] Hamzaoui, M., Ould-Elhassen Aoueileyine, M., Romdhani, L., & Bouallegue, R. (2023). An improved deep learning model for underwater species recognition in aquaculture. *Fishes*, *8*(10), 514.

[6] Corrigan, B. C., Tay, Z. Y., & Konovessis, D. (2023). Real-time instance segmentation for detection of underwater litter as a plastic source. *Journal of Marine Science and Engineering*, *11*(8), 1532.

[7] Sarkar, P., De, S., & Gurung, S. (2023). Fish detection from underwater images using YOLO and its challenges. In *Lecture Notes in Electrical Engineering* (pp. 159–169). Springer.

[8] Roboflow. (2023). Roboflow: Simplify computer vision dataset management. Retrieved from https://roboflow.com/

[9] Jocher, G. (2023). YOLOv8 by Ultralytics. Retrieved from https://github.com/ultralytics/ultralytics

[10] Ben Tamou, A., Benzinou, A., & Nasreddine, K. (2022). Live fish species classification in underwater images by using convolutional neural networks based on incremental learning with knowledge distillation loss. *Machine Learning and Knowledge Extraction*, *4*(3), 753–767.

[11] Saleh, A., Sheaves, M., & Rahimi Azghadi, M. (2022). Computer vision and deep learning for fish classification in underwater habitats: A survey. *Fish and Fisheries*, *23*(4), 977–999.

[12] Malla, S., Meena, M. J., Reddy, R. O., Mahalakshmi, V., & Balobaid, A. (2022). A study on fish classification techniques using convolutional neural networks on highly challenged underwater images. *International Journal on Recent and Innovation Trends in Computing and Communication*, *10*(4), 01–09.

[13] Mohd Rum, S. N., & Nawawi, F. A. Z. (2021). FishDeTec: A fish identification application using image recognition approach. *International Journal of Advanced Computer Science and Applications*, *12*(3), 92–97.

[14] Mohamed, H. E.-D., Fadl, A., Anas, O., Wageeh, Y., ElMasry, N., Nabil, A., & Atia, A. (2020). MSR-YOLO: Method to enhance fish detection and tracking in fish farms. *Procedia Computer Science*, *170*, 539–546.

[15] Saleh, A., Laradji, I. H., Konovalov, D. A., et al. (2020). A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Scientific Reports*, *10*(1), 14671.

[16] Allken, V., Handegard, N. O., Rosen, S., Schreyeck, T., Mahiout, T., & Malde, K. (2019). Fish species identification using a convolutional neural network trained on synthetic data. *ICES Journal of Marine Science*, *76*(1), 342–349.

[17] Andayani, U., Wijaya, A., Rahmat, R., Siregar, B., & Syahputra, M. (2019). Fish species classification using probabilistic neural network. *Journal of Physics: Conference Series*, *1235*(1), 012094.

[18] Xu, W., & Matzner, S. (2018). Underwater fish detection using deep learning for water power applications. arXiv preprint arXiv:1811.01494.