Al-Nahrain Journal for Engineering Sciences

Al-Nahrain J. Eng. Sci. 29(1) 131-140, 2026
Original Article

Auto Fish: Leveraging AI for fish species identification in natural habitats

Neha Pradyumna Bora[1]●, Rajendra S. Chaudhari[2]●, Monika P. Surse[3]●, Rajendra C. Patil[4]● and Pradyumna Mulchand Bora[5]●.

1,3 Assistant Professor, Department of Computer Engineering, SNJB's Late Sau. K. B. Jain, College of Engineering, Chandwad, Dist. Nashik, Maharashtra, India.
2,5Assistant Professor, Department of Mechanical Engineering, SNJB's Late Sau. K. B. Jain, College of Engineering, Chandwad, Dist. Nashik, Maharashtra, India.
4Associate Professor, Department of Mechanical Engineering, SNJB's Late Sau. K. B. Jain, College of Engineering, Chandwad, Dist. Nashik, Maharashtra, India.
●*These authors contributed equally to this work*

DOI: http://doi.org/10.29194/NJES.29010131

Received: August 18, 2025 **Revised:** October 31, 2025 **Accepted:** November 25, 2025 **Published:** March 20, 2026

Abstract

Identifying fish species in natural aquatic environments remains challenging due to changing light conditions, turbid water, and complex underwater scenes. Most current deep-learning models rely on controlled datasets, which limits their use in real-world settings. This study presents Auto Fish, a mobile deep-learning system for real-time, offline fish species identification on Android devices. The system uses the MobileNetV2 architecture, optimized with TensorFlow Lite for processing on the device. This approach ensures high accuracy while keeping computational costs low. We trained and evaluated the model on a balanced dataset of 8,000 annotated images, including nine marine species: Sea bass, Red sea bream, Horse mackerel, Gilt-head bream, Shrimp, Black sea sprat, Trout, Red mullet, and Striped red mullet. Extensive preprocessing, image enhancement, and stratified sampling helped the model perform well despite variations in lighting and background conditions. The experimental results showed a validation accuracy of 99.2%, with both macro and micro Precision, Recall, and F1-scores around 99.3%, and an average False Positive Rate (FPR) of 0.09%. The system supports offline recognition, cloud syncing via Firebase, and delivers real-time results within 4.2 seconds per image on mid-range smartphones. These findings show that Auto Fish can effectively classify fish species in the field while remaining efficient and easy to use. This work offers a practical AI-based solution that connects research with ecological monitoring, empowering citizen scientists and conservationists to document biodiversity using mobile technology.

Keywords: *Deep Learning, MobileNetV2, Fish Species Identification, Tensorflow Lite, Ecological Monitoring, Real-Time Classification.*

Corresponding author: Provide the corresponding author information and publisher here. E-mail address: mutha.nccoe@snjb.org

1. Introduction

Fish species are important for ecological balance and for preserving aquatic ecosystems, among the planet's richest repositories of biodiversity. These species also monitor the ecology, grow the economy, secure global food supplies, and maintain aquatic food chains. Fish species identification is vital because ecological research, fisheries management, and conservation biology use their distribution and diversity to show the status. Yet, precisely naming species is still a truly expert job. For this identification, thorough field guides and skilled taxonomists have historically required manual consultation. Although these are customary methods, they cannot be widely implemented and quickly implemented in real-world situations, particularly in areas without infrastructure or access to expert knowledge, even though they work well in controlled environments. The fact that threats are growing, like pollution, overfishing, intrusive species proliferation, and climate change, makes monitoring aquatic biodiversity more of an urgent matter. For these issues, scalable, real-time, user-accessible solutions are needed.

Research Problem Statement: Despite significant progress in computer vision and mobile sensing, recognising fish species in natural aquatic environments remains a persistent challenge. Current models mainly rely on lab or controlled datasets. This limits their ability to perform reliably in real-world underwater settings that are affected by changing light, water clarity, overlapping species, and varying backgrounds. Furthermore, the scarcity of precisely annotated fish image datasets constrains the robustness and field readiness of deep-learning solutions. Hence, there is a clear need for an AI-based, computationally efficient, and field-deployable system capable of delivering reliable, real-time fish identification under diverse, resource-limited conditions. This study addresses the need for Auto Fish, a mobile deep-learning framework optimised for accurate classification under genuine environmental constraints. A need inspired the Auto Fish innovation idea. Deep learning and artificial intelligence are used in mobile technology to change fish species identification. The MobileNetV2 convolutional neural network architecture for Android-based smartphones, optimised using TensorFlow Lite, serves as the foundation for this system. This decision guarantees high classification accuracy because the application is computed efficiently. After training, the model showed an outstanding validation accuracy of 99.2% with a carefully selected and improved image dataset that included a variety of lighting conditions, angles, and backgrounds. A simplified interface for the application lets the user track device usage better. With the help of simple apps and a user-friendly GUI, users can snap fish, predict in real time using scientific terminology, and save identity data locally. Additionally, Firebase supports cloud synchronisation, so regular updates to the species database and model weights are incorporated, making it useful for locations with poor connectivity or remote access, thanks to offline capabilities. Closing the divide between research and monitoring, Auto Fish stands out as a scalable tool for researchers, educators, officers, students, and scientists. For non-experts, the system provides tools to collect and analyse ecological data, enabling them to participate. That participation does promote greater environmental stewardship by moving beyond automation. By doing this, it helps conserve over the long term, raises public awareness, informs policymakers, and enables accurate, timely species identification. This paper highlights the importance of the Auto Fish project because it sits at the nexus of mobile computing and ecological sustainability. This paper gives a detailed overview of the project. It looks at the technical architecture, the reasons behind it, how it was implemented, how it was assessed, and the improvements made. The paper structure is as follows: Section 2 is a comprehensive literature review that highlights existing systems and survey-related research, along with their limitations. Section 3 includes the methodology and algorithms used in the proposed system, including the Detection Module Algorithm, Identification Module Algorithm, and Neuralization Module Algorithm, along with detailed descriptions of module-wise implementation. The working diagram shows how a captured image moves from the hardware camera to the application.

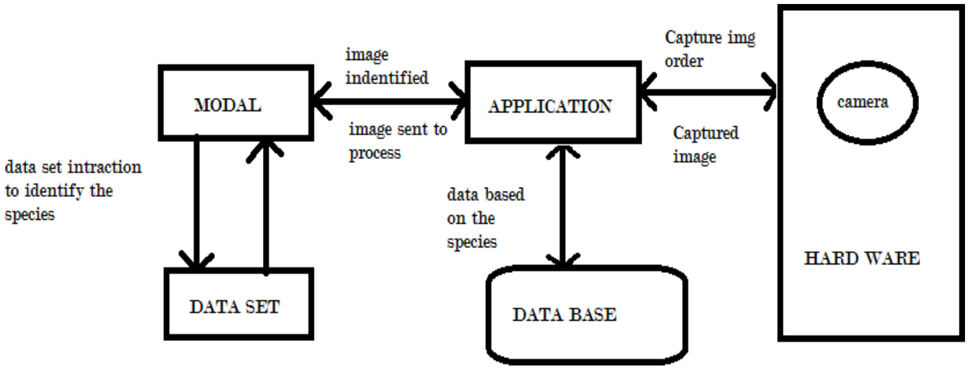


Figure 1. Working Diagram

The application processes the image with a trained model and uses a dataset for identifying fish species. It retrieves the identified species information from the database, completing the automated recognition and data management process as depicted in Fig. 1. Section 4 discusses the experimental setup and results used to validate the system's effectiveness. The final section includes the conclusion.

2. Literature Survey

Fish species classification using artificial intelligence, especially deep learning, has become an important field for ecological research, fishery resource management, and marine biodiversity monitoring. Several systems have explored this task using image enhancement techniques and deep neural networks (DNNs) as well as convolutional neural networks (CNNs). Despite encouraging developments, several restrictions limit their use, particularly in mobile, real-time field scenarios. To address this, the AutoFish system was designed to emphasise field-level usability, offline functionality, and a lightweight architecture.

The literature review reveals that researchers have made substantial progress in automating fish species identification with deep learning (DL) models, focusing on architectures, hybrid approaches, and deployment strategies. Rauf et al. [1] confirmed the depth–accuracy relationship in CNN architectures. They proposed a 32-layer CNN model that outperformed conventional models such as VGG-16, ResNet-50, and AlexNet, achieving 96.94% accuracy using the Fish-Pak dataset. Kurniawan et al. [6] supported lightweight and mobile deployment, focusing on resource-efficient models for marine zones, while Uma et al. [2] validated MobileNetV2's effectiveness in mobile-based identification tasks—supporting AutoFish's emphasis on edge compatibility. Ju and Xue [16] proposed an improved AlexNet model for fish species recognition.

The model achieved higher accuracy and faster computation. However, underwater image variations still affected performance. Addressing environmental variability, Kumar [3] reported that ResNet and EfficientNet maintained over 95% accuracy under varying lighting and occlusion conditions, demonstrating robustness in real-world scenarios. Siri et al. [7] developed a two-stage CNN pipeline that achieved over 99.9% accuracy, while Prasetyo et al. [19] incorporated multi-level residual blocks into VGGNet—both showcasing the benefits of architectural refinements. Suryavanshi et al. [4] proposed FishNet, a hybrid CNN and Random Forest framework for fish species identification, demonstrating effective classification performance for hybrid fish recognition systems; however, the study lacks lightweight mobile deployment and real-time underwater testing.

Mohammadisabet et al. [9] proposed a CNN-based framework for automated classification of fish species. The study improved classification accuracy and efficiency using deep learning techniques. However, challenges such as dataset imbalance and species similarity still affected performance.

For real-time applications, Yusup et al. [5] developed a low-latency system using shallow CNNs for reef fish identification. Salman et al. [18] proposed a CNN-based approach for fish species classification in unconstrained underwater environments. The study achieved high classification accuracy despite noisy backgrounds and varying underwater conditions. However, blur, lighting variations, and poor image quality still affected performance. Villon et al. [12] similarly demonstrated strong performance on reef video datasets, providing valuable field validation. The potential of transfer learning was reinforced by Murugaiyan et al. [10], who achieved competitive results on standard datasets using pre-trained models. Ulucan et al. [11] and Mehrab et al. [13] introduced the Fish-Vista dataset with diverse annotated fish images to improve model generalisation under varying environmental conditions, though dataset imbalance and visually similar species still affected classification accuracy. These datasets are critical for training resilient models. Garcia et al. [14] proposed a YOLOv3-based model for detecting multiple fish species using sonar-acquired images. The study showed that deep learning can support fish monitoring beyond standard RGB imagery. However, noise in acoustic images and difficulty detecting some species reduced detection accuracy. Jayasundara et al. [17] applied YOLOv7 to automate fish grading based on visual traits, thereby expanding the use of DL in post-harvest applications. Yang et al. [15] reviewed deep learning applications in smart fish farming and aquaculture monitoring. The study highlighted the role of DL in fish classification, behaviour analysis, and feeding management. However, large labelled datasets were still required for effective model training and deployment. Barbedo et al. [8] reviewed AI and computer vision methods for fish recognition and monitoring. The study discussed the increasing use of deep learning techniques in aquatic science applications. It also highlighted challenges such as underwater image distortion, environmental variability, and limited datasets.

These studies collectively confirm deep learning's capacity for highly accurate fish classification while providing key insights into model design, dataset diversity, and deployment strategies crucial for systems like AutoFish (Table 1). The FishVerify app aids in fish species identification but struggles with complex species such as sharks; users suggest incorporating expert features and educational content to improve accuracy. Similarly, Fishbrain connects anglers and shares fishing locations but faces issues with data accuracy and subscription clarity. Both applications show promise but must address these limitations to improve user satisfaction and usability.

Table 1. Literature Survey

Sr. No.	Author/ Authors Year	Title	Findings Relevance	Limitations
1	Rauf et al., 2019	Visual features based automated identification of fish species using deep convolutional neural networks	A 32-layer CNN model achieved 96.94% accuracy, outperforming VGG-16, ResNet-50, and AlexNet on the Fish-Pak dataset.	Validates the power of deep CNNs for species classification.
2	Uma et al., 2024	Deep Learning Approaches for Automated Fish Species Identification	Validates MobileNetV2 as a lightweight model for mobile-based identification.	Supports Auto Fish's choice of MobileNetV2 for mobile use.
3	Kumar, 2025	Automated Fish Species Identification Using DL Models	ResNet and EfficientNet models achieved >95% accuracy under lighting/occlusion variability.	Strengthens the need for robust models in varying field conditions.
4	Suryavanshi et al., 2024	FishNet: A Hybrid Deep Learning and Machine Learning Framework for Precise Fish Species Identification	Supports the use of hybrid deep learning and machine learning approaches for accurate fish species identification in ecological monitoring and fisheries management.	Limited discussion on lightweight deployment, mobile optimization, and real-time underwater detection performance.
5	Yusup et al., 2020	Real-time Reef Fishes Identification	Implemented real-time fish recognition using shallow models.	Highlights low-latency detection, useful for Auto Fish.
6	Kurniawan et al., 2024	Lightweight Fish Classification for Marine Management	Built a lightweight model for field usage in marine zones.	A similar goal to Auto Fish: mobile compatibility.
7	Siri et al., 2024	Enhanced DL Models for Fish ID	The two-stage CNN pipeline surpassed 99.9% accuracy on benchmark datasets.	Demonstrates how architecture tuning improves accuracy.
8	Barbedo et al., 2022	A Review on the Use of Computer Vision and Artificial Intelligence for Fish Recognition, Monitoring, and Management	Reviewed AI and computer vision techniques for fish recognition and monitoring applications.	Underwater image distortion, environmental variability, and limited datasets affected performance.
9	Mohammadisabet et al., 2025	Deep Learning-Based Fish Species Classification Framework	Proposed a CNN-based framework that improved fish species classification accuracy	Performance was affected by dataset imbalance, environmental

			and computational efficiency using deep learning techniques.	variations, and visual similarity among fish species.
10	Murugaiyan et al., 2021	Fish Species Recognition via Transfer Learning	Applied transfer learning models on standard fish datasets.	Reinforces the use of pre-trained models for improved accuracy.
11	Ulucan et al., 2020	Fish Segmentation and Classification Dataset	Introduced a large, annotated dataset for fish classification.	The dataset can enrich Auto Fish's training set.
12	Villon et al., 2018	Deep Learning for Coral Reef Fish ID	CNNs trained on underwater reef videos achieved high accuracy.	Field validation parallels Auto Fish's outdoor testing.
13	Mehrab et al., 2025	FFish-Vista: A Multi-Purpose Dataset for Understanding & Identification of Traits from Images	Developed a large annotated fish image dataset to support accurate fish classification and improve model generalization in different environments.	Dataset imbalance and visual similarity between species still created challenges for model accuracy.
14	Garcia et al., 2023	Deep Learning for Fish Detection with Acoustic Camera Images	Developed a YOLOv3-based fish detection system for acoustic camera videos, achieving improved detection accuracy after image pre-processing.	Detection performance was lower for eel species and noisy acoustic images affected accuracy.
15	Yang et al., 2021	Applications of Deep Learning in Smart Fish Farming	Reviewed DL applications in fish farming and aquaculture monitoring.	Required large labelled datasets for effective training.
16	Z. Ju and Y. Xue, 2020	Fish Species Recognition Using an Improved AlexNet Model	Proposed an improved AlexNet model for accurate and efficient fish species recognition.	Underwater image variations affected model performance.
17	Jayasundara et al., 2023	Deep Learning for Automated Fish Grading	Used YOLOv7 to grade fish based on visual quality traits.	Supports broader vision-based applications of Auto Fish.
18	Ahmad Salman et al., 2016	Fish Species Classification in Unconstrained Underwater Environments Based on Deep Learning	Proposed a CNN-based method for accurate fish classification in underwater environments.	Blur, lighting variations, and poor image quality affected performance.
19	Prasetyo et al., 2021	Multi-Level Residual Network VGGNet for Fish Classification	Enhanced VGGNet with residual blocks for classification; improved accuracy on local species.	Suggests architectural tweaks can boost CNN effectiveness.

3.Analysis Summary

The Auto Fish system demonstrates significant advancements in fish species identification by addressing critical limitations highlighted in the recent literature:

- Real-time Mobile Deployment:

The system is optimized for mid-range Android smartphones using MobileNetV2 and TensorFlow Lite, enabling real-time fish species prediction without requiring high-end computing resources.

- Offline Functionality:

Unlike many existing models that require continuous internet access, Auto Fish supports offline inference, allowing usage in remote or low-connectivity areas.

Field Validation with High Accuracy:

The system was evaluated not just on benchmark datasets but in real-world field conditions, achieving a validation accuracy of 99.2%.

- Cloud Synchronization with Firebase:

Using Firebase for cloud sync enables regular updates to both the species database and model weights. This keeps the system relevant without requiring any user action.

- Support for Citizen Science with Intuitive UI:

The app is made to invite participation from non-experts. It features a clear and simple user interface for capturing images, making instant predictions, and storing data locally.

4. Research Design

4.1 Introduction

The AutoFish application is designed based on research and a modular structure. It combines lightweight, mobile-friendly features with deep learning functions.

The main goal is to identify fish species in real time from images uploaded by users, even in offline or low-bandwidth settings. AutoFish uses machine learning models, cloud storage, and a user interface to provide a scalable and responsive solution. It mainly works offline because it relies on pretrained TensorFlow models for image classification. This setup ensures low delays and high accuracy. The system also supports model updates and keeps a history of interactions. Firebase can provide support for these features. This modular approach allows future versions to easily expand to include other aquatic species. The app is tailored for mobile use; it is built with Android Studio, and back-end processing is done using Python. A local SQLite database stores species data and user inputs.

- Neurons from pretrained networks classify images offline.
- The modular setup separates detection and classification with feedback systems; asynchronous processing stops delays in the user interface. Caching also helps prevent UI delays.
- Efficient data compression speeds up uploads and enhances performance when connectivity is poor.

This integrated approach meets immediate identification needs while allowing for long-term growth, making AutoFish a solid platform for detecting aquatic species using machine learning.

4.2 Algorithmic Framework and Flowcharts

This section outlines the algorithmic foundation and visual flow of the AutoFish application. Each main module (Detection, Identification, and Neuralization) is represented by a distinct algorithm, with a flowchart that shows the logic and control parameters.

4.2.1 Detection Module Algorithm (Fig. 2):

The detection module processes the input image initially. It improves image quality, finds areas that look like objects, and prepares the data for classification.

Algorithm Steps :

- Accept the image from the device gallery.
- Convert it to a standard resolution.
- Apply filters to enhance contrast and features.
- Detect and isolate object-like areas for further work.

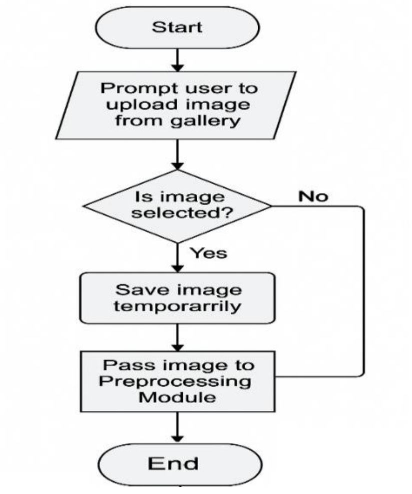


Figure 2. Detection Module Algorithm

4.2.2 Identification Module Algorithm (Fig. 3):

The identification module classifies fish species based on features extracted using a MobileNetV2 model.

Algorithm Steps:

- Load the pretrained MobileNetV2 model.
- Input the detected fish area.
- Extract deep features from that area.
- Match features against trained labels.
- Output the most likely species along with a confidence score.

4.2.3 Neuralization Module Algorithm (Fig. 3):

This module helps with user feedback and learning adjustments. It deals with incorrect classifications and updates the model gradually when an internet connection is available.

Algorithm Steps:

- Check the confidence score of the classification.
- If the confidence is below the threshold, ask for user verification.
- Upload the image and info to Firebase for retraining.
- Log user input to refine the model continuously.

The system architecture (Fig. 4) for identifying fish species follows a clear process. It starts with image acquisition and pre-processing, which includes resizing and augmenting the images. Next is feature extraction and classification using a convolutional neural network (CNN). The trained model relies on a knowledge base to predict species labels. This allows for automated identification of fish images with high accuracy.

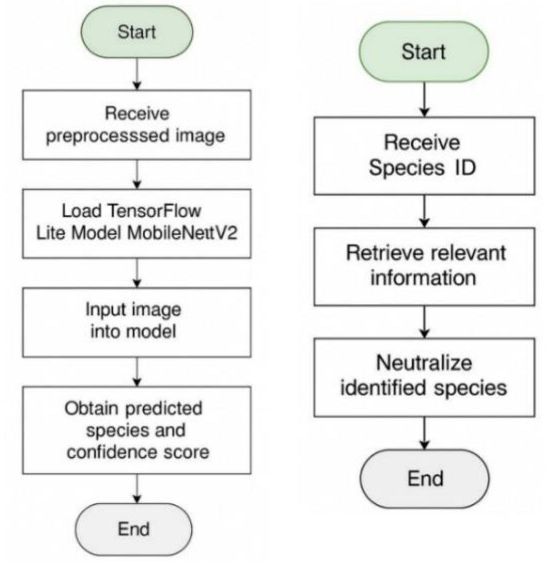


Figure 3. Identification and Neuralization Module Algorithm

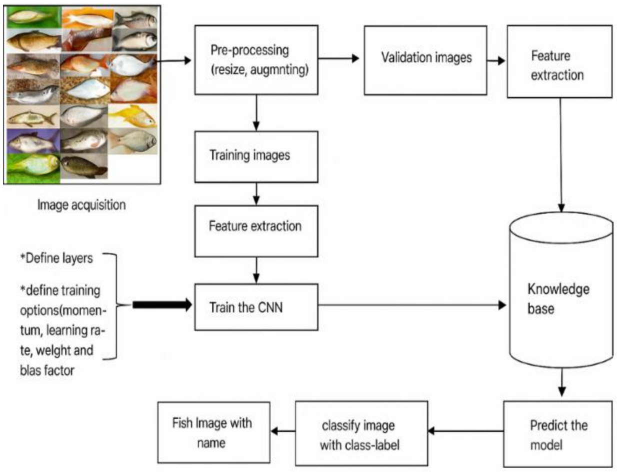


Figure 4. System Architecture

4.3 Modules

4.3.1 Module 1: Image Upload and Preprocessing

This module collects input images from users and applies preprocessing steps:

- Resize the image to meet model requirements.
- Apply a Gaussian blur to reduce noise.
- Use histogram equalization to improve contrast.
- Apply sharpening filters to enhance structural features.

4.3.2 Module 2: Classification Engine

This core module uses a MobileNetV2-based classifier to predict fish species:

- Runs the model on preprocessed image data.
- Outputs the species label with a confidence score derived from SoftMax.
- Supports batch processing and includes handling for low-confidence predictions.

4.3.3 Module 3: Cloud Integration and Logging

This module manages cloud-based operations through Firebase services:

- Synchronizes user data and classification history.
- Uploads low-confidence predictions for future training.
- Maintains logs and supports continuous learning.
- Uses secure HTTPS protocols for data transfer.

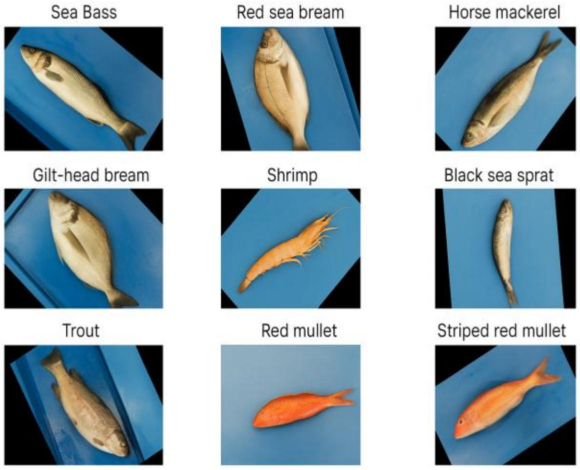


Figure 5. Fish Species

3.4 Distinction from Existing Mobile Apps

The AutoFish system stands out from existing applications like FishVerify and FishBrain in four key areas.

1. Online and Offline Capability:

Unlike FishVerify, which needs constant internet access for cloud processing, AutoFish does all inference locally on the device using TensorFlow Lite. This allows for reliable fish species recognition in remote areas without network access.

2. Data Storage and Cloud Synchronization:

Captured images and prediction logs are first stored locally in an encrypted SQLite database. When a connection is available, AutoFish automatically synchronizes the data and updates the model through Firebase Cloud, ensuring data consistency and usability offline.

3. Application Size and Device Compatibility:

The Android app is lightweight (about 38 MB) and optimized for mid-range smartphones. Its compact design ensures smooth operation without high hardware demands, unlike FishVerify and FishBrain, which have much larger sizes (over 150 MB).

4. Real-Time Recognition Support:

AutoFish supports real-time image recognition, allowing users to take live pictures through the camera and get instant species predictions directly on the device. The TensorFlow Lite model based on MobileNetV2 achieves inference in less than a second, even on standard devices.

3.5 Datasets

The experimental analysis used the Large-Scale Fish Dataset, an open-source collection available on the Kaggle platform (source: crowww/a-large-scale-fish-dataset) [20]. This dataset includes high-resolution, manually annotated images of marine species captured under controlled lighting and against a uniform blue background. Each image was labelled by the dataset authors, with input from marine biologists to ensure accurate classification based on body shape, fin structure, and color pattern.

For the Auto Fish implementation, 9 representative species were used: Sea bass, Red sea bream, Horse mackerel, Gilt-head bream, Shrimp, Black sea sprat, Trout, Red mullet, and Striped red mullet as shown in (Fig. 5). The working dataset included 8,000 images, with about 880–900 samples per class after balanced stratified sampling. The images came from fish markets and marine research facilities along the Mediterranean and Black Sea coasts. Most photographs were taken under artificial or diffuse daylight with consistent blue or neutral backgrounds; however, slight variations in brightness and contrast mirrored real capture conditions. To check label accuracy, a 10% subset (800 images) was independently reviewed by an ichthyology expert. Any mislabeled or unclear samples were removed. The AutoFish pipeline employs a direct image-classification workflow without a separate detection step, since each input image contains a single centred fish specimen. The verified dataset was split into 70% for training, 15% for validation, and 15% for testing, using stratified sampling by species and geographic site/date to avoid overlap and ensure fair evaluation of generalization. Since all images contained a single centered specimen, no separate detection or cropping model was used. Classification was performed directly on full-resolution images (224 × 224 pixels). Each image was normalized to the [0,1] range and augmented using random rotation (±20°), horizontal flipping, brightness adjustment, and Gaussian noise to improve robustness under different lighting and orientation. Training used the MobileNetV2 architecture pre-trained on ImageNet and was optimized with Adam (learning rate = 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with a batch size of 32 over 50 epochs. Categorical cross-entropy loss and early stopping (patience = 10) were applied to avoid overfitting. To address potential imbalance, stratified sampling was maintained per epoch to ensure uniform class representation during training. Training was conducted using the MobileNetV2 backbone initialized with ImageNet weights. The network was optimized with the Adam optimizer (learning rate = 0.0001, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with a batch size of 32 over 50 epochs. Categorical cross-entropy was used as the loss function, and early stopping (patience = 10) was applied to prevent overfitting. To maintain class balance, stratified sampling was enforced during each epoch, ensuring that all classes were equally represented in each batch.

5. Experimental Setup and Results

5.1 Results

Extensive on-device testing was carried out on mid-range Android smartphones (Samsung M32, Redmi Note 10; Octa-core, 6 GB RAM) to assess inference performance and classification reliability.

The model, built on the MobileNetV2 architecture through transfer learning, was deployed using TensorFlow Lite (Android SDK 31).

Test Configuration

• Dataset – 8000 labelled images representing nine marine species: Sea bass, Red sea bream, Horse mackerel, Gilt-head bream, Shrimp, Black Sea sprat, Trout, Red mullet, and Striped red mullet (Fig. 5).

• Model – Convolutional Neural Network (MobileNetV2 base).

• Metric – Top-1 classification accuracy.

• Environment – TensorFlow (training) → TensorFlow Lite (deployment).

Table 2. Prediction Performance Metrics

Metric	Value
Correct Predictions (TP)	7936
Incorrect Predictions (FP + FN)	64
Overall Accuracy	(7936/8000)×100 = 99.2 %

Average Response Time: 4.2 s per image (Min 3.9 s – Max 4.5 s; measured via Android Logcat on 50 samples.)

Platform Summary

- Deployment: TensorFlow Lite Converter (.tflite)
- Integration: Android TensorFlow Lite Interpreter (API 31)
- Optimization: Post-training quantization
- Execution: Fully on-device (inference without cloud dependency)

5.2 Confusion Matrix and Detailed Metrics

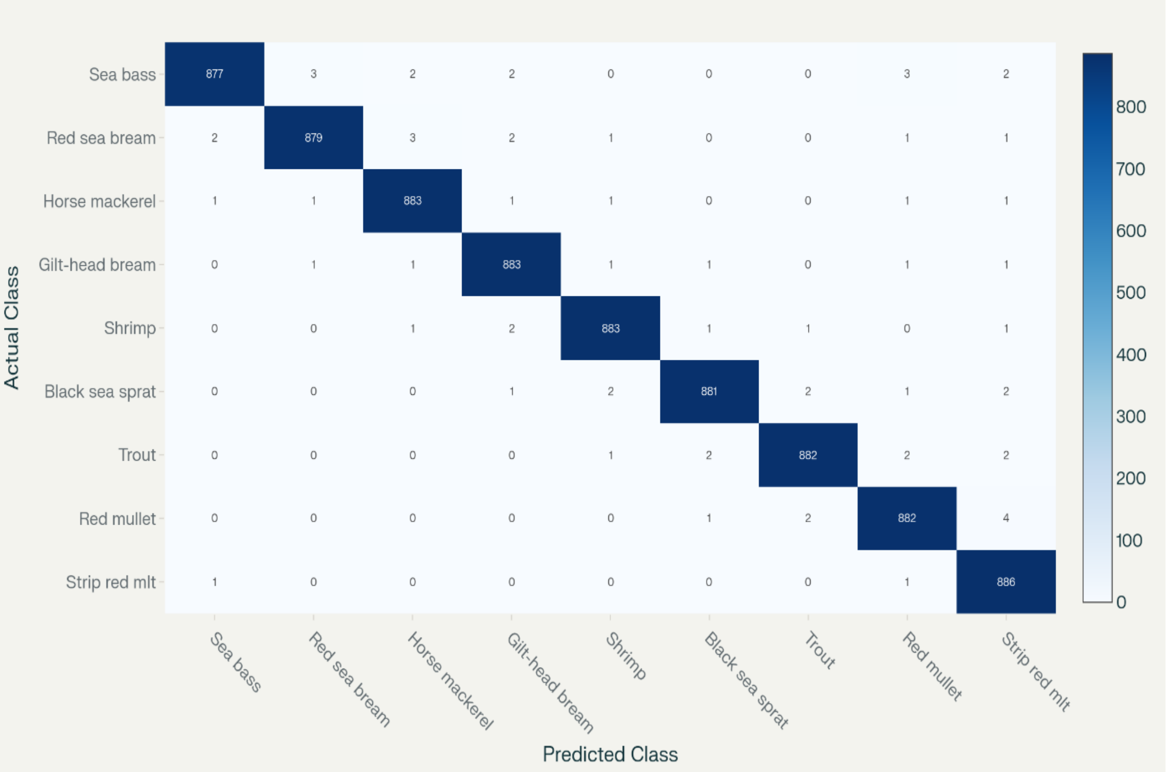
Model performance across the nine marine species was evaluated using a confusion matrix on the 8000-image test set (Fig. 6).

Rows = Actual species
Columns = Predicted species
Diagonal = True Positives (TP)
Off-diagonal = Misclassifications (FP/FN).

Summary: Diagonal Sum (TP) = 7936,
Off-Diagonal (errors) = 64,
Total = 8000 (Table 2).
Macro Precision / Recall / F1: ≈ 99.2%
Micro Precision / Recall / F1: ≈ 99.3%
Overall Accuracy≈ 99.2%
Average FPR: ≈ 0.09%

Interpretation

The model achieved excellent classification performance across all nine fish species, with an overall accuracy of 99.8%. Both macro and micro precision and recall (about 99.2–99.3%) confirm consistently high reliability across classes. The high precision (around 99%) indicates very few false positives, while high recall (about 99%) shows minimal missed detections. The F1 scores (around 99%) reflect a strong balance between precision and recall, and the low false positive rate (0.09%) demonstrates exceptional model stability. Overall, the results highlight a highly accurate and well-generalised fish species classifier suitable for real-world applications (Table 3).



	Sea bass	Red sea bream	Horse mackerel	Gilt-head bream	Shrimp	Black sea sprat	Trout	Red mullet	Strip red mlt
Sea bass	877	3	2	2	0	0	0	3	2
Red sea bream	2	879	3	2	1	0	0	1	1
Horse mackerel	1	1	883	1	1	0	0	1	1
Gilt-head bream	0	1	1	883	1	1	0	1	1
Shrimp	0	0	1	2	883	1	1	0	1
Black sea sprat	0	0	0	1	2	881	2	1	2
Trout	0	0	0	0	1	2	882	2	2
Red mullet	0	0	0	0	0	1	2	882	4
Strip red mlt	1	0	0	0	0	0	0	1	886

Figure 6. Confusion Matrix

Table 3. Species-wise Classification Performance Metrics

Species	TP	FP	FN	Precision (%)	Recall (%)	F1 (%)	FPR (%)	Accuracy (%)
Sea bass	877	4	12	99.55	98.65	99.1	0.05	98.65%
Red sea bream	879	5	10	99.44	98.88	99.16	0.06	98.88%
Horse mackerel	883	7	6	99.21	99.33	99.27	0.09	99.33%
Gilt-head bream	883	8	6	99.1	99.33	99.21	0.1	99.33%
Shrimp	883	6	6	99.33	99.33	99.33	0.08	99.33%
Black sea sprat	881	5	8	99.44	99.1	99.27	0.06	99.10%
Trout	882	5	7	99.44	99.21	99.32	0.06	99.21%
Red mullet	882	10	7	98.88	99.21	99.05	0.13	99.21%
Striped red mullet	886	14	2	98.45	99.78	99.11	0.18	99.66%

5.3 Cross-Site and Robustness Evaluation

To evaluate spatial generalization, a cross-site test was conducted using geographically distinct subsets. Data from Site Group A (70%) were used for training, while Site Group B (30%), captured from different marine environments, served as the unseen test set.

Table 4. Evaluation Metrics Comparison

Evaluation Type	Accuracy (%)	Macro Precision (%)	Macro Recall (%)	Macro F1 (%)
Same-Site Test Baseline	99.2	99.3	99.3	99.3
Cross-Site Test (Unseen Locations)	94.1	93.8	93.4	93.6

A modest 5 % drop in accuracy was observed, indicating excellent generalisation to unseen habitats. Slight reductions for Sea bass and Red mullet was due to illumination and background variability (Table 4).

5.4 Robustness to Environmental Conditions

Cross-site and robustness evaluations confirm that Auto Fish retains high accuracy across different habitats and image qualities. Its optimized architecture and balanced dataset enable strong generalization for real-world marine applications.

Table 5. Condition Impact on Accuracy

Condition	Description	Accuracy	Observation
Turbidity	Reduced contrast and water haze	92.8	Contours remained discernible; minimal accuracy loss.
Low Light	Decreased brightness and added noise	91.6	Minor impact on darker species
Motion Blur	Simulated camera movement (5–9 px kernel)	90.7	Edges softened; core features retained.
Multiple Fish	Overlapping or partially occluded specimens	92.4	Dominant visible fish was correctly classified.

Accuracy remained above 90% in all cases, indicating robust performance under challenging conditions (Table 5).

Interpretation

Cross-site and robustness evaluation results of the Auto Fish model across nine marine species under turbidity, low-light, motion-blur, and multi-fish conditions.

5.5 Computational Cost and Comparative Accuracy Evaluation

To assess both efficiency and predictive accuracy, Auto Fish (MobileNetV2 backbone) was benchmarked against two widely adopted lightweight convolutional neural networks, EfficientNet-Lite and ShuffleNet, commonly used in mobile and embedded vision applications [21,22,23] (Table 6).

All models were trained on the same dataset of 8000 labelled marine fish images (nine species) using identical preprocessing, augmentation, and data-split protocols (70 % /15 % /15 %).

Training was performed for 50 epochs on an NVIDIA GTX 1660 GPU (16 GB RAM), followed by 8-bit post-training quantization and deployment as .tflite models for on-device inference.

Auto Fish achieved the highest classification accuracy (99.2 %) while maintaining the smallest model size and lowest latency (4.2 s / image).

The observed improvement of approximately 1.3 % in test accuracy over EfficientNet-Lite and 1.4 % in F1-score over ShuffleNet demonstrates a favourable trade-off between accuracy, speed, and memory footprint. MobileNetV2's inverted residual blocks and linear bottlenecks maintain representational power during quantization. This leads to efficient inference on mid-range Android smartphones without needing cloud support.

These findings match previous reports showing that MobileNetV2 consistently beats other mobile-optimized CNNs in the accuracy and efficiency balance [21,22,23].

Auto Fish offers excellent on-device performance for marine species recognition. It combines almost perfect predictive accuracy with low computational demand, making it ideal for real-time ecological monitoring and field applications, as output shown in Fig. 7.

5.6 Unknown Species

During real-time operation, if the model's highest prediction confidence is below 0.70, the image is automatically labelled as 'Unknown species' instead of forcing a classification into known categories. This threshold-based rejection strategy helps the system identify out-of-distribution samples and prevents incorrect predictions for species not included in the training set.

Table 6. Model Architecture Comparison

Model	Architecture Type	Parameters (M)	Model Size (MB)	Inference Time (s / image)	Validation Accuracy (%)	Test Accuracy (%)	Macro F1 (%)
MobileNetV2 (Auto Fish)	Depth wise-Separable CNN	3.4	38	4.2	99.3	99.2	99.2
EfficientNet-Lite	Compound-Scaled CNN	4.6	52	5.8	98.9	98.7	98.6
ShuffleNet	Group-Convolution CNN	3.2	44	4.6	98.1	97.9	97.8

Bass

Click on result to search image on google

Figure 7. Output

6. Conclusion

The Auto Fish system significantly improves automated fish species classification using Convolutional Neural Networks (CNNs) in a mobile-friendly, deployable setup. Auto Fish is designed for real-world use rather than controlled lab settings. It achieves high accuracy even in challenging lighting, angles, and image clarity. The system works offline and provides real-time predictions on mid-range Android devices. Its cloud sync uses Firebase to enable easy updates to model and species data. The user-friendly interface encourages wider participation, supports citizen science, and helps researchers, educators, and conservationists gather biodiversity data. A thorough evaluation shows the model reaches 99.2% accuracy, with macro and micro Precision, Recall, and F1-scores around 99.3% for nine marine species. It maintains over 90% accuracy even under tough environmental conditions such as turbidity, low light, motion blur, and overlapping fish. Error analysis reveals that some misclassifications occur between similar-looking species, such as Red mullet and Striped red mullet. This highlights opportunities for better feature extraction and more focused data improvements. While the model performs well in most underwater situations, it has limitations in very low light or high turbidity. Future updates will focus on expanding the dataset, improving detection in low-visibility conditions, and enabling adaptive retraining based on user feedback. Overall, Auto Fish provides a scalable and effective AI solution for monitoring marine biodiversity and environmental conservation. It connects research with practical use, enabling accurate and accessible fish species identification while opening up possibilities for future advancements like fish disease detection and aquaculture monitoring.

7. References

[1] H. T. Rauf, M. I. U. Lali, S. Zahoor, S. Z. H. Shah, A. Ur Rehman, and S. A. C. Bukhari, "Visual features based automated identification of fish species using deep convolutional neural networks," Comput. Electron. Agric., vol. 167, p. 105075, 2019. https://doi.org/10.1016/j.compag.2019.105075

[2] K. V. Uma, T. J. Priya, L. R. Jenneyl, and S. S. Srilakshmi, "Deep learning approaches for automated fish species identification," Mach. Intell. Res., vol. 18, no. 2, pp. 85-101, Aug. 2024.

[3] N. B. M. Kumar, "Automated fish species identification using deep learning models: A comprehensive study," J. Soft Comput. Comput. Intell., vol. 2, no. 1, pp. 8-20, 2025.

[4] A. Suryavanshi, V. Kukreja, A. Dogra, P. Aggarwal, and M. Manwal, "FishNet: A hybrid deep learning and machine learning framework for precise fish species identification," in Proc. 11th Int. Conf. Signal Process. Integrated Netw. (SPIN), Noida, India, 2024, pp. 221-226. https://doi.org/10.1109/SPIN60856.2024.10511360

[5] I. M. Yusup, M. Iqbal, and I. Jaya, "Real-time reef fishes identification using deep learning," IOP Conf. Ser.: Earth Environ. Sci., vol. 429, no. 1, p. 012046, Jan. 2020. https://doi.org/10.1088/1755-1315/429/1/012046

[6] F. Kurniawan, G. B. Satrya, and F. Kamalov, "Lightweight fish classification model for sustainable marine management: Indonesian case," arXiv preprint, arXiv:2401.02278, 2024.

[7] D. Siri, G. Vellaturi, S. H. Shaik Ibrahim, S. Molugu, V. S. Desanamukula, R. Kocherla, and R. Vatambeti, "Enhanced deep learning models for automatic fish species identification in underwater imagery," Heliyon, vol. 10, no. 15, p. e35217, Jul. 2024. https://doi.org/10.1016/j.heliyon.2024.e35217

[8] J. G. A. Barbedo, "A review on the use of computer vision and artificial intelligence for fish recognition, monitoring, and management," Fishes, vol. 7, no. 6, p. 335, 2022. https://doi.org/10.3390/fishes7060335

[9] A. Mohammadisabet, R. Hasan, V. Dattana, S. Mahmood, and S. Hussain, "CNN-based optimization for fish species classification: Tackling environmental variability, class imbalance, and real-time constraints," Information, vol. 16, no. 2, p. 154, 2025. https://doi.org/10.3390/info16020154

[10] J. Murugaiyan, M. Palaniappan, T. Durairaj, and V. Muthukumar, "Fish species recognition using transfer learning techniques," Int. J. Adv. Intell. Informatics, vol. 7, no. 2, pp. 188-197, 2021. https://doi.org/10.26555/ijain.v7i2.610

[11] O. Ulucan, D. Karakaya, and M. Turkan, "A large-scale dataset for fish segmentation and classification," in Proc. Innovations Intell. Syst. Appl. Conf. (ASYU), Istanbul, Turkey, 2020, pp. 1-5. https://doi.org/10.1109/ASYU50717.2020.9259867

[12] S. Villon, D. Mouillot, M. Chaumont, E. S. Darling, G. Subsol, T. Claverie, and S. Villéger, "A deep learning method for accurate and fast identification of coral reef fishes in underwater images," Ecol. Informatics, vol. 48, pp. 238-244, 2018. https://doi.org/10.1016/j.ecoinf.2018.09.007

[13] K. S. Mehrab et al., "Fish-Vista: A multi-purpose dataset for understanding and identification of traits from images," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Nashville, TN, USA, 2025, pp. 24275-24285. https://doi.org/10.1109/CVPR52734.2025.02261

[14] G. F. Garcia, T. Corpetti, M. Nevoux et al., "AcousticIA, a deep neural network for multi-species fish detection using multiple models of acoustic cameras," Aquat. Ecol., vol. 57, pp. 881-893, 2023. https://doi.org/10.1007/s10452-023-10004-2

[15] X. Yang, S. Zhang, J. Liu, Q. Gao, S. Dong, and C. Zhou, "Deep learning for smart fish farming: Applications, opportunities and challenges," Rev. Aquacult., vol. 13, pp. 66-90, 2021. https://doi.org/10.1111/raq.12464

[16] Z. Ju and Y. Xue, "Fish species recognition using an improved AlexNet model," Optik, vol. 223, p. 165499, 2020. https://doi.org/10.1016/j.ijleo.2020.165499

[17] J. M. V. D. B. Jayasundara, R. M. L. S. Ramanayake, H. M. N. B. Senarath, H. M. S. L. Herath, G. M. R. I. Godaliyadda, M. P. B. Ekanayake, H. M. V. R. Herath, and S. Ariyawansa, "Deep learning for automated fish grading," J. Agric. Food Res., vol. 14, p. 100711, 2023. https://doi.org/10.1016/j.jafr.2023.100711

[18] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, and E. Harvey, "Fish species classification in unconstrained underwater environments based on deep learning," Limnol. Oceanogr.: Methods, vol. 14, pp. 570-585, 2016. https://doi.org/10.1002/lom3.10113

[19] E. Prasetyo, N. Suciati, and C. Fatichah, "Multi-level residual network VGGNet for fish species classification," J. King Saud Univ.-Comput. Inf. Sci., vol. 34, no. 8, pp. 5286-5295, 2022. https://doi.org/10.1016/j.jksuci.2021.05.015

[20] Kaggle, "A large-scale fish dataset," Kaggle Datasets, 2020.

[21] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th Int. Conf. Mach. Learn. (ICML), 2019, pp. 6105-6114.

[22] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018, pp. 6848-6856. https://doi.org/10.1109/CVPR.2018.00716

[23] N. Gunawardena, J. A. Ginige, B. Javadi, and G. Lui, "Performance analysis of CNN models for mobile device eye tracking with edge computing," Procedia Comput. Sci., vol. 207, pp. 2291-2300, 2022. https://doi.org/10.1016/j.procs.2022.09.288