Deep Learning-Based Small Fish Species Classification Using Transfer Learning and Grad-CAM Explainability

Md. Eftakhar Alam (2104010202321)[1*],
Md. Ajmal Hossain (2104010202328)[1] and
Imtiaz Siddique Mahim (2104010202342)[1]

[1*]Department of Computer Science and Engineering, Premier University , Chattogram, Bangladesh.

*Corresponding author(s). E-mail(s): eftakhar@example.com;
Contributing authors: ajmal@example.com; mahim@example.com;

Abstract

This data article presents a comprehensive image dataset of ten native small fish species commonly found in Bangladesh: Bele (Glossogobius giuris), Chanda Nama (Chanda nama), Chela (Salmostoma bacaila), Guchi (Mastacembelus pancalus), Kachki (Corica soborna), Mola (Amblypharyngodon mola), Kata Phasa (Stolephorus tri), Pabda (Ompok pabda), Puti (Puntius sophore), and Tengra (Mystus vittatus). The dataset was carefully curated to facilitate the study and research in fish species identification, classification, and biodiversity monitoring. Specimens of these species were collected from various fish markets in the capital city Dhaka. Different varieties of fish are supplied to Dhaka city from diverse geographical locations in Bangladesh. Thus, the dataset ensures a representative sampling of local aquatic biodiversity. To maintain uniformity across samples, images were captured using a smartphone camera under a standardized and controlled environment. Each specimen was placed against a neutral background with consistent lighting conditions. This limits environmental variability and enhances image quality for analytical use. The dataset contains high-resolution original images that were augmented using standard data augmentation techniques. This augmentation introduced variations such as rotations, flipping, and brightness adjustments. This expands the dataset and improves its utility for training robust machine learning (ML) and deep learning (DL) models in computer vision applications. The dataset has significant reuse potential across multiple domains. It serves as a critical resource for researchers and industry experts to develop automated systems for fish species identification and classification, particularly in the context of the rich aquatic biodiversity in Bangladesh. Furthermore, the dataset can facilitate ecological and environmental studies and research by supporting the monitoring of native fish species distribution and population dynamics. Its structured format facilitates integration into ML/DL pipelines that can foster advancements in fisheries management, sustainable aquaculture, conservation biology, and economic and cultural studies. Thus, the dataset represents a significant step towards integrating technological advancements and ecological sustainability. This article outlines the utility of the data, the dataset structure, the data collection methodology, and the applied augmentation processes to ensure transparency and reproducibility for future research endeavors.

Keywords: Deep learning, Transfer learning, Fish classification, ResNet50V2, MobileNetV2, EfficientNetB0, Grad-CAM, Explainable AI

1 Introduction

Bangladesh is home to a rich diversity of freshwater fish species, many of which belong to the small indigenous fish (SIS) category. These species—such as Puti, Tengra, Mola, Kachki, and Chanda—play an important role in ecological balance, food nutrition, and the local economy. However, accurate species identification remains a major challenge due to high visual similarity among fish, variations in lighting conditions, background noise, and inconsistency in image capture. Traditional taxonomy-based identification requires expert knowledge and is time-consuming, making it inefficient for large-scale monitoring, fish market analysis, and biodiversity assessment.

With recent advancements in computer vision, deep learning methods—particularly Convolutional Neural Networks (CNNs)—have demonstrated excellent performance in image-based classification tasks. Transfer learning, which leverages pre-trained models on large-scale datasets, has enabled accurate classification even when training data is limited. Models such as MobileNetV2, ResNet50V2, and EfficientNetB0 have been widely adopted due to their efficiency, generalization capability, and robustness.

Despite progress in fish image classification, challenges remain for small fish species of Bangladesh due to limited publicly available datasets, image inconsistency, and lack of explainability in model decisions. To address these issues, this work develops a complete end-to-end deep learning framework for small fish species classification using the SmallFishBD dataset. The study applies an extensive preprocessing pipeline—including resizing, brightness correction, denoising, Gaussian blur, rotation, and augmentation—to enhance data quality and robustness.

Three transfer learning models (MobileNetV2, ResNet50V2, and EfficientNetB0) are trained and evaluated using standardized metrics such as accuracy, loss, ROC-AUC, and confusion matrix. Additionally, Gradient-weighted Class Activation Mapping (Grad-CAM) is employed to generate visual explanations, enabling interpretability of the models' decisions and helping identify the specific regions used for classification.

The findings contribute to developing automated fish recognition systems that can support ecological research, fish market monitoring, sustainable fisheries management, and future large-scale biodiversity tracking in Bangladesh.

2 Related Work

Research on automated fish species classification has evolved significantly over the past decade, driven by advancements in computer vision and deep learning. Early studies primarily focused on traditional machine-learning approaches that relied on hand-crafted features to describe fish appearance. Methods such as colour histograms, geometric descriptors, Scale-Invariant Feature Transform (SIFT), Local Binary Patterns (LBP), and texture-based descriptors were commonly used. These features were then fed into classical classifiers like Support Vector Machines (SVM), Random Forests or k-Nearest Neighbours (k-NN). Although these approaches worked reasonably well under controlled conditions, their performance degraded sharply in real-world environments due to variations in illumination, background noise, camera angle and species similarity. The high visual similarity among many small fish species made hand-crafted features insufficient for robust recognition.

With the emergence of deep learning, particularly Convolutional Neural Networks (CNNs), the field witnessed a major shift. CNN-based models automatically learn hierarchical image representations, eliminating the need for manual feature engineering. Several studies explored CNNs for marine and freshwater fish classification, showing substantial improvements in accuracy. For example, researchers applied deep architectures such as AlexNet, VGGNet and Inception for identifying underwater species in challenging ocean environments. These models demonstrated strong generalization ability, but required large annotated datasets, which are often scarce in ecological research.

To overcome data limitations, transfer learning has become the dominant strategy. Models pre-trained on large-scale datasets like ImageNet—such as MobileNetV2, ResNet, DenseNet and EfficientNet—have been widely adapted for fish classification tasks. Transfer learning enables faster convergence, reduced training time, and improved performance even with relatively small datasets. ResNet-based models, due to their skip-connection design, have shown exceptional accuracy in fine-grained recognition tasks, including distinguishing species with subtle morphological differences. Similarly, EfficientNet models achieve strong accuracy–efficiency trade-offs by scaling depth, width and resolution uniformly, making them suitable for lightweight ecological applications. MobileNetV2, designed for edge deployment, has been used in mobile-based aquatic monitoring systems, though its lightweight design sometimes limits performance on highly complex datasets.

In addition to classification accuracy, recent research emphasises model interpretability and explainability, especially for ecological and environmental decision-making. Tools such as Gradient-weighted Class Activation Mapping (Grad-CAM) and Layer-wise Relevance Propagation (LRP) provide visual explanations of CNN decisions by highlighting the regions most responsible for each prediction. Studies have shown that interpretable models help biologists verify whether the network focuses on meaningful body parts—such as fins, scales or head structure—rather than irrelevant background artefacts. This is particularly important when working with visually similar species, where misclassification must be understood rather than blindly accepted.

Although several studies exist on fish classification using deep learning, most focus on large marine species or underwater footage. Only limited work addresses small indigenous fish species (SIS) of Bangladesh, which require finer discrimination due to their compact size and similar body shapes. Moreover, many previous works do not explore the complete pipeline—from dataset cleaning, augmentation and preprocessing to deep-model training and explainability. This creates a research gap in developing a unified system capable of handling noisy real-world images and providing trustworthy predictions.

Our work aims to fill this gap by combining extensive preprocessing, three modern transfer-learning architectures (MobileNetV2, ResNet50V2 and EfficientNetB0), and Grad-CAM explainability into a comprehensive framework. By evaluating multiple models on the SmallFishBD dataset and analysing their decision patterns, we contribute a deeper understanding of how deep networks behave in fine-grained fish species classification, paving the path for practical, scalable solutions in ecological monitoring and fisheries management.

3 Dataset Description

3.1 Data Collection

Ten native small fish species commonly found in Bangladesh were selected for this study. These species were collected from wholesale fish markets located in Dhaka, where a wide variety of freshwater fish from different regions of the country are traded daily. To ensure correct species identification, each fish category was first validated with the help of domain experts and verified using reliable online and academic resources. This verification step ensured that the dataset accurately represented the true morphological characteristics of each species.

For every collected specimen, high-quality photographs were captured using a smartphone camera under controlled lighting conditions. A neutral background and white LED illumination were used to minimise shadows and background noise. Images were taken from multiple angles to capture variations in body shape, fin placement, colour patterns and textural details. This multi-angle capture setup ensured that the dataset contained sufficient visual diversity for robust model training.

The original images were captured in a 3:4 aspect ratio. To maintain uniformity across the dataset, necessary padding was applied along the height or width when required. All images were then resized to 224×224 $224{\times}224$ pixels (for model input standardisation) while preserving key visual features. Each fish species was stored in a dedicated folder named after its commonly used local name, forming the raw dataset known as SmallFishBD.

After the initial image collection, the dataset underwent further enhancement through systematic augmentation. Each raw image was transformed multiple times using Python-based preprocessing tools. The augmentations included changes in brightness, geometric rotations (30°, 45°, 60°, 90°, 180°), horizontal and vertical flipping, Gaussian noise injection, blurring and normalization. These operations expanded the dataset and improved model robustness against illumination variation, camera angle changes and image noise. The final augmented dataset, referred to as Augmented SmallFishBD, contained both original and augmented samples for all ten fish categories.

This comprehensive data collection and augmentation process ensured that the dataset captured a wide range of realistic variations, enabling effective training of deep learning models for small fish species classification.

3.2 Data Source Location

The fish specimens used in this study were collected from multiple wholesale and local fish markets across Dhaka, Bangladesh. These locations are known for receiving diverse freshwater fish varieties from different regions of the country. The exact collection points with their geographic coordinates are listed below:

- **Dhanmondi Bazar**, Dhaka, Bangladesh (23.7481° N, 90.3692° E)
- **Meradia Kacha Bazar**, Dhaka, Bangladesh (23.7629° N, 90.4447° E)
- **Kalshi Bazar**, Dhaka, Bangladesh (23.8217° N, 90.3744° E)
- **Kawran Bazar**, Dhaka, Bangladesh (23.7523° N, 90.3944° E)
- **Rampura Bridge Fish Market**, Dhaka, Bangladesh (23.7697° N, 90.4245° E)

3.3 Data Accessibility

The dataset used in this study is publicly available through the Mendeley Data repository. The details of the dataset are provided below:

- **Repository Name:** SmallFishBD: A Comprehensive Image Dataset of Common Small Fish Varieties in Bangladesh for Species Identification and Classification
- **Data Identification Number:** 10.17632/8jvxtvz52x.2
- **Direct URL to Dataset:** https://data.mendeley.com/datasets/8jvxtvz52x/2

This repository contains the raw images, organized by fish species, along with metadata and relevant documentation required for research and reproducibility.

3.4 Value of the Data

The SmallFishBD dataset offers significant value for multiple research and application domains. The key benefits are outlined below:

- **Fish Species Identification and Taxonomy:** The dataset provides a detailed visual record of ten native small fish species of Bangladesh. It enables researchers, taxonomists, and biologists to accurately detect, classify, and differentiate species, thereby contributing to the understanding of biodiversity and the ecological importance of these fish.
- **Machine Learning and Computer Vision Research:** The dataset serves as a useful benchmark for developing and evaluating machine learning and deep learning

models for fish recognition tasks. It supports the creation of automated fish identification systems that can be applied to fisheries management, real-time monitoring, and intelligent aquaculture technologies.

- **Food Security and Aquaculture:** Small indigenous fish species are an essential component of the diet and nutritional intake in Bangladesh. This dataset can aid sustainable aquaculture practices by enabling species-level monitoring, preventing misidentification, and supporting decision-making in production planning, market distribution, and overfishing mitigation.
- **Ecological Significance and Conservation:** Native small fish species play an important role in maintaining aquatic food chains and ecosystem balance. The dataset facilitates ecological research by supporting long-term monitoring of species populations, identifying vulnerable or declining species, and informing conservation and habitat restoration policies. It also contributes to understanding the impacts of pollution, climate change, and habitat degradation on local fish diversity.
- **Post-Harvest Fish Processing:** The dataset can be used to design automated imaging systems for sorting, grading, and quality assessment of small fish in post-harvest processing. Such systems can enhance operational efficiency, reduce manual labour requirements, and improve quality control in commercial fisheries.
- **Socioeconomic and Educational Applications:** Small fish varieties are culturally, economically, and nutritionally significant in Bangladesh. This dataset supports socioeconomic studies related to fish markets, supply-chain dynamics, and community livelihood development. It also serves as an educational resource to raise awareness about aquatic biodiversity and the importance of conserving native species.

3.5 Data Description

The dataset used in this study consists of images of multiple small fish species native to Bangladesh. All collected images were organized into clean and processed versions to support a complete deep learning workflow. The dataset was initially obtained as raw images captured under real-world market conditions, which introduced variability in image resolution, illumination, noise, orientation, and background. Therefore, extensive preprocessing and dataset restructuring were required before model development.

3.5.1 Dataset Organization

The final dataset was arranged into two major components:

- **SmallFishBD_clean** – A fully cleaned and standardized dataset used for validation and testing.
- **SmallFishBD_clean_blur_noise** – A heavily augmented dataset used exclusively for training.

Each dataset contains multiple class-wise folders where each folder represents one fish species. All images are stored in JPG format. To maintain consistency across samples, all images were resized to 224×224 pixels, and corrupted, duplicate, low-light, and irrelevant images were removed.

3.5.2 Clean Dataset (SmallFishBD_clean)

This dataset contains the preprocessed images with the following improvements applied:

- Uniform resizing to 224×224 pixels
- Brightness normalization
- Controlled noise reduction
- Center-cropping and padding where necessary
- Removal of distorted or corrupted images

This version of the dataset is divided into three subsets:

- `train/` – Clean training samples
- `val/` – Validation samples used for monitoring model performance
- `test/` – A separate test set used only for final evaluation

These subsets ensure non-overlapping images between training and evaluation phases.

3.5.3 Processed Training Dataset (SmallFishBD_clean_blur_noise)

Since the clean dataset contains limited samples, a richer training dataset was created by applying several augmentation techniques. The following transformations were applied:

- Image rotations at various angles
- Horizontal and vertical flipping
- Brightness enhancement and reduction
- Gaussian noise injection
- Blurring (Gaussian blur and median blur)
- Minor geometric distortions

The purpose of this processed dataset is to increase sample diversity and prevent overfitting during training. Only the `train/` folder exists in this augmented dataset.

3.5.4 Folder Structure

The dataset structure is as follows:

- `/SmallFishBD_clean/train`
- `/SmallFishBD_clean/val`
- `/SmallFishBD_clean/test`
- `/SmallFishBD_clean_blur_noise/train`

Each folder contains sub-folders corresponding to individual fish species, such as:

- `/puti/`

- `/pabda/`
- `/mola/`
- `/tengra/`

This hierarchical structure ensures smooth integration with deep learning pipelines, especially for Keras, PyTorch, and TensorFlow image loaders.

Table 1 Number of images per fish variety in the original SmallFishBD dataset and the augmented training dataset.

Fish Varieties (Bengali names)	**No. of images in SmallFishBD**	**No. of images in Augmented dataset**
Bele	205	3280
Chela	190	3040
Guchi	164	2620
Kachki	247	3940
Mola	179	2880
Nama Chanda	110	1760
Kata Phasa	129	2060
Pabda	125	1940
Puti	218	3480
Tengra	133	2120
Total	**1700**	**27100**

3.6 Sample Visualization of Fish Images

To develop an initial understanding of the dataset, a set of random sample images from each fish category was visualized, as shown in Fig. 1. These representative samples help illustrate the diversity in appearance, orientation, and coloration across the ten small fish varieties included in the dataset.

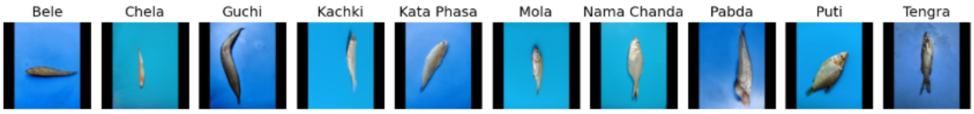


Fig. 1 Randomly selected images from the training set for each fish category. From left to right: Bele, Chela, Guchi, Kachki, Kata Phasa, Mola, Nama Chanda, Pabda, Puti, and Tengra.

These images highlight several important characteristics of the dataset. First, the fish species exhibit varying textures, shapes, and color patterns, which are essential for classification. Second, although the background color is mostly uniform (blue), there are subtle variations in lighting and orientation, simulating real-world diversity. Third, the black padding around images ensures a consistent 1:1 aspect ratio, which helps the neural networks process uniformly shaped inputs. Such initial visualization is crucial before model training, as it reveals potential challenges such as class similarity, intra-class variation, and subtle distinguishing features.

3.7 RGB Channel Distribution Analysis

To better understand the illumination characteristics and color distribution of the SmallFishBD dataset, RGB channel histograms were generated for all ten fish species. These histograms help reveal background dominance, species-wise texture variation, and color-level imbalance. Figures 12 show the RGB channel distributions arranged three per row for better readability.

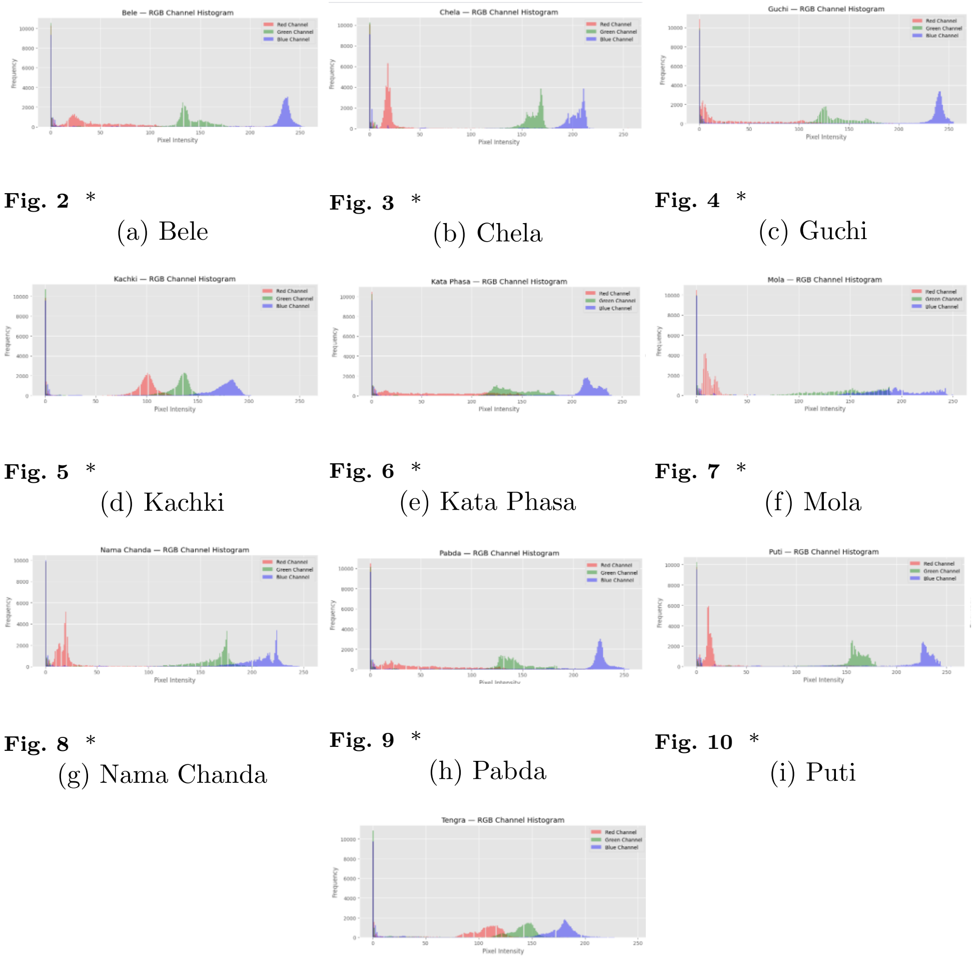


Fig. 2 *
(a) Bele

Fig. 3 *
(b) Chela

Fig. 4 *
(c) Guchi

Fig. 5 *
(d) Kachki

Fig. 6 *
(e) Kata Phasa

Fig. 7 *
(f) Mola

Fig. 8 *
(g) Nama Chanda

Fig. 9 *
(h) Pabda

Fig. 10 *
(i) Puti

Fig. 11 *
(j) Tengra

Fig. 12 RGB Channel Histograms of Ten Fish Species in the SmallFishBD dataset.

3.8 RGB Channel Histogram Analysis

Figure 12 illustrates the RGB channel histograms for all ten fish species in the SmallFishBD dataset. These histograms provide an important understanding of the underlying color distribution, illumination conditions, and background characteristics that influence model learning.

Across all species, the **blue channel consistently exhibits the highest intensity values**, reflecting the dominant blue background commonly used during image acquisition. The **green channel shows mid-range intensities**, typically corresponding to fish body textures and natural shading. In contrast, the **red channel maintains the lowest intensity spread**, highlighting its limited presence in most species except those with reddish fins or skin regions, such as Puti and Guchi.

Species such as *Bele*, *Guchi*, *Pabda*, and *Tengra* display distinct multi-peak distributions in the green and blue channels, indicating stronger texture variations that may assist in species-level discrimination. Meanwhile, species like *Chela*, *Mola*, and *Nama Chanda* exhibit narrow distributions dominated by the background, suggesting challenges for feature extraction due to low color contrast.

These histogram patterns also reveal inter-class similarities in background illumination, which underline the importance of employing robust deep learning models capable of learning fine-grained features beyond simple color cues. Overall, the RGB analysis validates the need for advanced preprocessing and data augmentation to minimize background bias and improve classification performance.

4 Preprocessing and Data Augmentation

4.1 Image Preprocessing

To ensure that the images in the SmallFishBD dataset were suitable for developing robust machine learning and deep learning models, a comprehensive preprocessing pipeline was applied. The original images collected from various fish markets exhibited variations in resolution, lighting conditions, orientation, and background noise. To standardize these images and reduce computational cost, all samples were first resized to 224 × 224 pixels. This resizing was performed using the `PIL.Image` module of the Python Pillow library, which preserves essential structural details of the fish while reducing storage and memory requirements.

Since convolutional neural network (CNN) models require fixed input dimensions, uniform resizing ensures a consistent spatial representation across the entire dataset. Unlike the raw dataset, which maintained a 3:4 aspect ratio, the resized dataset used for model training did not require black padding because the objective was to prepare the data directly for deep learning workflows. Instead, any necessary aspect ratio adjustments were applied proportionally during preprocessing to avoid distortion of fish shapes.

After resizing, multiple enhancement steps were applied to improve image quality and feature separability. Brightness adjustment was performed to normalize illumination differences among images. Next, rotation-based transformations were applied to simulate variations in fish orientations typically observed in real-world scenarios.

Additional transformations such as Gaussian blurring and noise injection were carried out to make the dataset more robust to visual distortions and environmental inconsistencies. These operations were implemented using Python libraries such as `PIL`, `cv2`, and `numpy`.

Following enhancement and augmentation operations, all images were normalized to the $[0, 1]$ pixel intensity range to stabilize and accelerate model convergence during training. Proper normalization also ensures that gradients remain well-behaved and prevents numerical instability in deep network layers.

Finally, the dataset was organized into three subsets—train, validation, and test—with the training set containing the augmented and quality-enhanced images, while the validation and test sets contained only clean, resized images. This structure ensures that the model is evaluated on data that has not been artificially modified, thereby producing more reliable performance metrics.

Through this preprocessing pipeline, the SmallFishBD dataset was transformed into a standardized, noise-reduced, and augmentation-rich collection of samples optimized for CNN-based classification. This process ensures that models trained on the dataset are more accurate, generalizable, and robust to real-world variations in fish images.

4.2 Image Resizing and Input Standardization

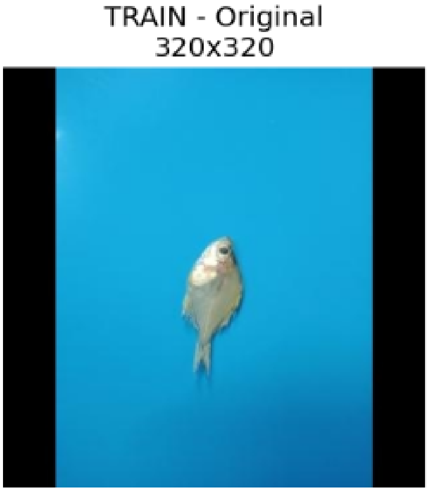


Fig. 13 *

TRAIN – Original
320×320

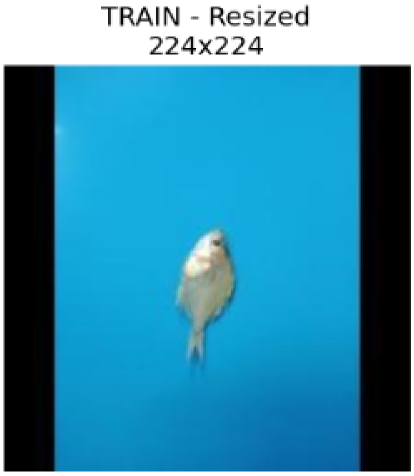


Fig. 14 *

TRAIN – Resized
224×224

Fig. 15 Comparison between the original padded image (left) and the resized image used for CNN input (right). The resizing preserves object structure while standardizing dimensions for efficient model training.

To ensure compatibility with deep learning architectures, all images in the dataset were resized from their original padded resolution of 320×320 pixels to the model-specific input resolution of 224×224 pixels. This resizing step reduces computational cost, accelerates training, and ensures that the convolutional neural networks (CNNs) receive a consistent input size across all samples.

Figure 15 illustrates a side-by-side comparison between an original padded image (left) and its resized version (right). The resizing process retains the fish's structural features while minimizing distortion, ensuring that essential morphological details—such as fin shape, body curvature, and scale texture—remain clearly visible for accurate classification. This transformation was implemented using the `Image` module of the Python `PIL` library, which offers high-quality interpolation for image downsizing.

4.3 Brightness Adjustment

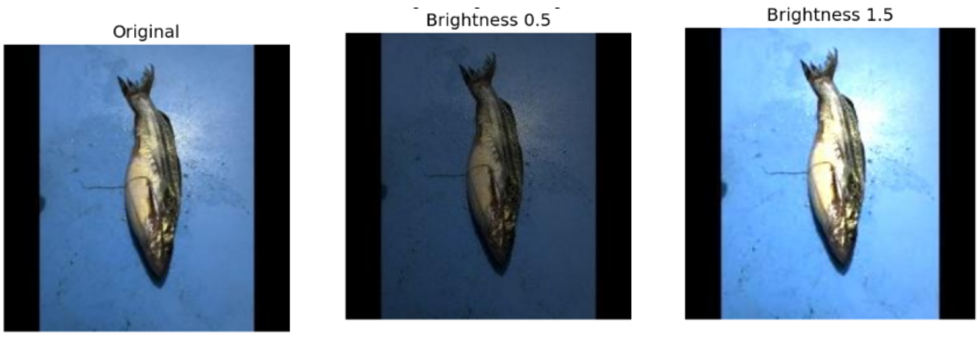


Fig. 16 *
(a) Original image.

Fig. 17 *
(b) Brightness 0.5: darker version.

Fig. 18 *
(c) Brightness 1.5: brighter version.

Fig. 19 Brightness augmentation examples showing original, darkened, and brightened versions of the same fish image.

Brightness variation is an essential augmentation technique used to simulate different lighting environments in which fish images may be captured. Real-world image acquisition often encounters conditions such as low light, shadows, overexposure, or uneven illumination. To ensure that the deep learning models can generalize well under such variations, brightness augmentation was applied to each original image.

Two controlled brightness transformations were generated:

- **Brightness 0.5 (Darker Version):** Simulates dim lighting, shadowed regions, or indoor capture conditions with insufficient illumination.
- **Brightness 1.5 (Brighter Version):** Represents overexposed scenarios, strong reflections, or high-intensity LED lighting commonly found in fish markets.

These variations help the model learn stable feature representations independent of lighting inconsistencies, making the classification system more robust in real-world operational environments.

4.4 Rotation Augmentation

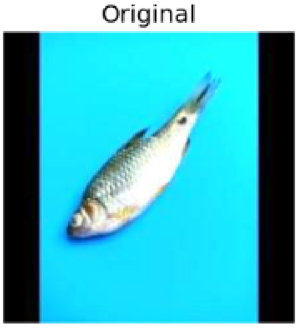


Fig. 20 *
(d) Base rotated image.

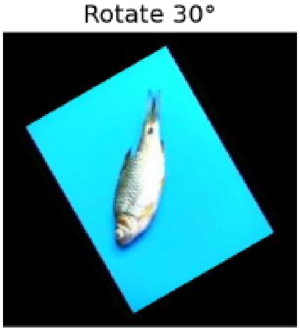


Fig. 21 *
(e) Rotation 30°.

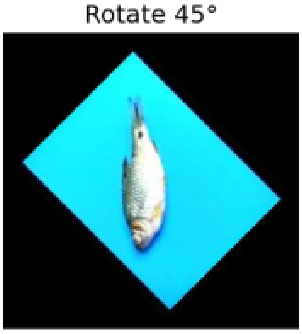


Fig. 22 *
(f) Rotation 45°.

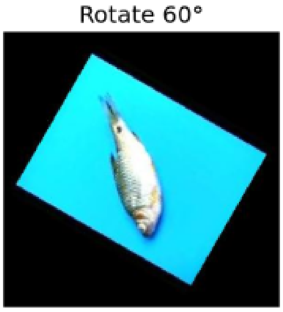


Fig. 23 *
(g) Rotation 60°.

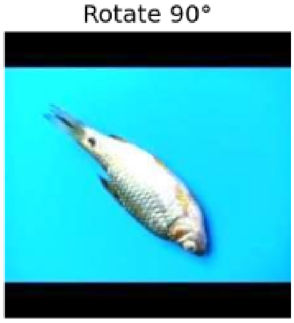


Fig. 24 *
(h) Rotation 90°.

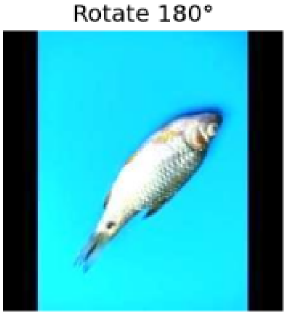


Fig. 25 *
(i) Rotation 180°.

Fig. 26 Examples of rotation augmentation applied to fish images at different angles.

Rotation is one of the most essential augmentation strategies used to ensure that the deep learning models become invariant to different orientations of the fish. In real-world scenarios, fish may appear at various angles depending on how they are placed, handled, or photographed. To simulate such variability and improve the robustness of the classification models, each image in the dataset was rotated at five distinct angular transformations: 30°, 45°, 60°, 90°, and 180°.

These rotations help the CNN models learn to extract consistent features regardless of orientation. Lower-angle rotations (30°, 45°, 60°) capture natural tilting during image capture, while higher-angle rotations (90° and 180°) simulate extreme orientations that may occur due to random placement or camera rotation. This allows the model to generalize more effectively and prevents performance degradation when encountering rotated fish images in real-world deployments.

4.5 Blur and Noise Augmentation

In practical fish recognition scenarios, images are often captured under non-ideal conditions, such as camera motion, low-quality lenses, sensor artifacts, or noisy environments. To make the deep learning models robust against such degradations, blur and noise-based augmentations were applied to the dataset. These operations help the network learn stable and discriminative features even when image quality is partially compromised.

Gaussian blur was used to simulate out-of-focus or motion-blurred images, where fine textures and edges become less sharp. Noise augmentation, on the other hand, was applied to imitate random pixel disturbances introduced by camera sensors or environmental interference. Additionally, a combined blur+noise variant was generated to represent more challenging real-world conditions. Together, these augmentations strengthen the model's ability to handle low-quality and imperfect images during deployment.

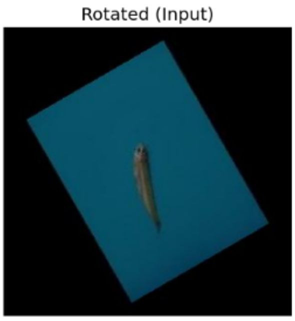


Fig. 27 *
(j) Noise-added image: simulates sensor and environment noise.

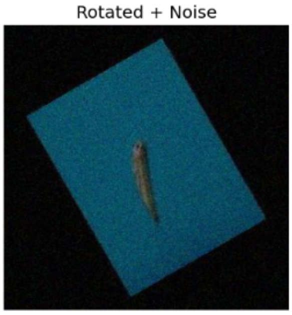


Fig. 28 *
(k) Gaussian blur: unfocused appearance.

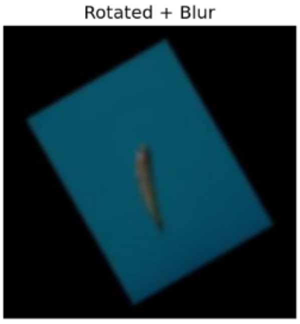


Fig. 29 *
(l) Blur + noise: combined degradation for robustness.

Fig. 30 Examples of noise and blur-based augmentations applied to the fish images to simulate sensor artifacts, out-of-focus capture, and combined degradation.

4.6 Normalization (0–1 Scaling)

Normalization is a crucial preprocessing step in preparing images for machine learning and deep learning models. Since raw pixel values range from 0 to 255, directly feeding these values into a neural network may lead to unstable training, slower convergence, and poor performance due to inconsistent feature scaling. To address this issue, each image in the dataset was normalized by scaling pixel intensities to the range of 0–1 using the transformation:

$$I_{\text{norm}} = \frac{I_{\text{pixel}}}{255}$$

This transformation ensures that all input values fall within a uniform distribution, making gradient optimization more stable and efficient. Normalization also prevents any single channel or high-intensity region from dominating the learning process, which is especially important in fish classification where subtle color variations and fine-scale textures play a significant role in distinguishing species.

In our project, normalization was applied after resizing, brightness adjustment, rotation, and other augmentation techniques. This guarantees that all augmented versions maintain consistent pixel value ranges, allowing the CNN models to focus entirely on learning discriminative fish features. Normalization additionally reduces the computational burden and helps models converge faster during training.

Overall, normalization contributes significantly to the model's ability to generalize by ensuring uniform pixel value distribution across both original and augmented datasets.

5 Methodology

5.1 Deep Learning Models Used

To perform small fish species classification on the SmallFishBD dataset, three state-of-the-art convolutional neural network (CNN) architectures were selected: MobileNetV2, ResNet50V2, and EfficientNetB0. These models represent lightweight, mid-level, and efficient modern deep learning architectures, enabling a comprehensive comparison across different computational complexities and feature extraction strengths.

MobileNetV2 is a lightweight CNN architecture based on inverted residual blocks with linear bottlenecks. It uses depthwise separable convolutions, which significantly reduce the number of parameters and computational cost. This makes MobileNetV2 suitable for real-time or resource-constrained environments. Despite its compactness, it effectively captures important structural features of small fish, such as fin orientation, body curvature, and texture patterns.

ResNet50V2 is a deeper and more expressive residual network that uses identity shortcut connections to mitigate the vanishing gradient problem. The model learns hierarchical and fine-grained visual representations, making it ideal for distinguishing visually similar small fish species in the dataset. Its improved residual block structure and batch normalization allow faster convergence and improved stability during training.

EfficientNetB0 employs compound scaling, which uniformly scales depth, width, and input resolution using an optimized scaling coefficient. This architecture balances accuracy and efficiency by integrating MBConv layers with squeeze-and-excitation optimization. EfficientNetB0 captures both global and local morphological patterns, such as body shape, skin brightness, edge contours, and pixel-level texture, making it highly effective for natural and augmented fish images.

These three CNN models were selected to analyze how model depth, parameter size, and architectural design affect performance on a fine-grained classification task such as distinguishing between ten species of native Bangladeshi small fish. Their complementary strengths provide a robust comparative analysis for identifying the most effective architecture for small fish species classification under varied lighting, rotation, noise, and blur conditions present in the preprocessed dataset.

Table 2 Size and total number of parameters employed in the evaluated models.

Model Name	Size	Total Number of Parameters (Millions)
MobileNetV2	9.20 MB	2.26 Million
ResNet50V2	98.00 MB	25.61 Million
EfficientNetB0	29.50 MB	5.33 Million

5.2 Dataset Splitting and Model Training Setup

For this study, the original SmallFishBD dataset was divided into three subsets using a standard stratified partitioning ratio of 70:15:15 for the training, validation, and testing sets, respectively. This ensures that each fish species is proportionally represented across all subsets. To prepare the images for deep learning models, all samples in the training, validation, and testing sets were uniformly resized to a spatial resolution of $224 \times 224 \times 3$, enabling efficient computation while preserving essential visual features.

To enhance the diversity of the training data and prevent overfitting, three augmentation techniques—brightness variation, rotation, and noise/blur—were randomly applied to the training images. These augmentations simulate realistic variations in lighting, orientation, and visual quality, ultimately improving model robustness and generalization.

All three CNN architectures used in this project (MobileNetV2, ResNet50V2, and EfficientNetB0) were trained using the same hyperparameter settings to maintain a fair comparison. The hyperparameters and their assigned values are summarized in Table-3. Transfer learning was utilized by initializing each model with ImageNet pre-trained weights, which accelerates convergence and improves feature extraction by leveraging previously learned representations. Fine-tuning was subsequently performed on the fish-specific dataset to adapt the models to the target classification task.

Table 3 Selected hyperparameters used for training MobileNetV2, ResNet50V2, and EfficientNetB0 models.

Hyperparameter Name	**Value**
Learning Rate	0.0001
Optimizer	Adam
Loss Function	Categorical Crossentropy
Batch Size	32
Epochs	30

Early Stopping Parameters	Patience = 10
	Monitor = Validation Loss
	Mode = Minimum
	Restore_best_weights = True

Learning Rate Scheduler (ReduceLROnPlateau)	Monitor = Validation Loss
	Factor = 0.2
	Patience = 3
	Min LR = 1e-7

Final Layer Activation Function	Softmax
Dropout Rate	30%
Image Input Size	224 × 224 × 3
Weight Initialization	ImageNet Pretrained Weights
Data Augmentation Applied	Brightness (0.5, 1.5) Rotations (30°, 45°, 60°, 90°, 180°) Gaussian Blur Noise Addition Normalization (0–1)

6 Results

6.1 Overall Model Performance

To compare the effectiveness of the three transfer learning models, we evaluated their performance on the held-out test set using overall classification accuracy and the micro-averaged ROC–AUC score. The summary results are reported in Table 4. Among the evaluated architectures, EfficientNetB0 achieved the best performance with a test accuracy of 0.8249 and a micro ROC–AUC of 0.9791, indicating strong discriminative capability across all fish classes. ResNet50V2 obtained a moderate test accuracy of 0.6780 and micro ROC–AUC of 0.9397, showing that it can separate most classes reasonably well but with more misclassifications than EfficientNetB0. MobileNetV2, while computationally lightweight, produced the lowest test accuracy of 0.4802 and micro ROC–AUC of 0.8843, suggesting that its compact architecture struggles to capture the fine-grained visual details needed for small fish species recognition. Overall, these results highlight a clear trade-off between model complexity and accuracy, with EfficientNetB0 providing the best balance for our dataset.

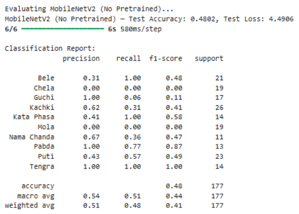
Table 4 Overall test performance of the three CNN models.

Model	Test Accuracy	Micro ROC–AUC
MobileNetV2	0.4802	0.8843
ResNet50V2	0.6780	0.9397
EfficientNetB0	**0.8249**	**0.9791**

6.2 Classification Reports of the Evaluated Models

To further analyze model performance on individual fish species, we generated classification reports for all three deep learning models used in this study: MobileNetV2, ResNet50V2, and EfficientNetB0. Each report presents precision, recall, F1–score, and support for all ten fish categories in the dataset.

Figure 34 shows a side-by-side comparison of the classification reports. From the visual comparison, it is evident that MobileNetV2 struggles to generalize across multiple classes, while ResNet50V2 demonstrates improved recall and balanced F1–scores. EfficientNetB0 outperforms the other models, with higher precision and overall better class-wise consistency.



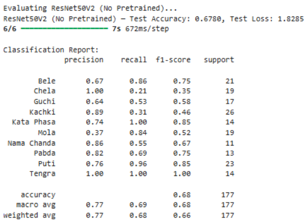
```
Evaluating MobileNetV2 (No Pretrained)...
MobileNetV2 (No Pretrained) — Test Accuracy: 0.4802, Test Loss: 4.4906
6/6 ━━━━━━━━━━━━━━━━━━━━ 6s 580ms/step

Classification Report:
              precision    recall  f1-score   support

        Bele       0.31      1.00      0.48        21
       Chela       0.00      0.00      0.00        19
       Guchi       1.00      0.06      0.11        17
      Kachki       0.62      0.31      0.41        26
  Kata Phasa       0.41      1.00      0.58        14
        Mola       0.00      0.00      0.00        19
 Nama Chanda       0.67      0.36      0.47        11
       Pabda       1.00      0.77      0.87        13
        Puti       0.43      0.57      0.49        23
      Tengra       1.00      1.00      1.00        14

    accuracy                           0.48       177
   macro avg       0.54      0.51      0.44       177
weighted avg       0.51      0.48      0.41       177
```

Fig. 31 *
(a) **MobileNetV2**: Shows low recall for several classes and an overall modest accuracy.



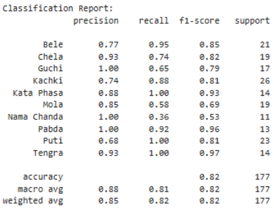
```
Evaluating ResNet50V2 (No Pretrained)...
ResNet50V2 (No Pretrained) — Test Accuracy: 0.6780, Test Loss: 1.8285
6/6 ━━━━━━━━━━━━━━━━━━━━ 7s 672ms/step

Classification Report:
              precision    recall  f1-score   support

        Bele       0.67      0.86      0.75        21
       Chela       1.00      0.21      0.35        19
       Guchi       0.64      0.53      0.58        17
      Kachki       0.89      0.31      0.46        26
  Kata Phasa       0.74      1.00      0.85        14
        Mola       0.37      0.84      0.52        19
 Nama Chanda       0.86      0.55      0.67        11
       Pabda       0.82      0.69      0.75        13
        Puti       0.76      0.96      0.85        23
      Tengra       1.00      1.00      1.00        14

    accuracy                           0.68       177
   macro avg       0.77      0.69      0.68       177
weighted avg       0.77      0.68      0.66       177
```

Fig. 32 *
(b) **ResNet50V2**: Demonstrates improvement in recall and more stable class predictions.



```
Classification Report:
              precision    recall  f1-score   support

        Bele       0.77      0.95      0.85        21
       Chela       0.93      0.74      0.82        19
       Guchi       1.00      0.65      0.79        17
      Kachki       0.74      0.88      0.81        26
  Kata Phasa       0.88      1.00      0.93        14
        Mola       0.85      0.58      0.69        19
 Nama Chanda       1.00      0.36      0.53        11
       Pabda       1.00      0.92      0.96        13
        Puti       0.68      1.00      0.81        23
      Tengra       0.93      1.00      0.97        14

    accuracy                           0.82       177
   macro avg       0.88      0.81      0.82       177
weighted avg       0.85      0.82      0.82       177
```

Fig. 33 *
(c) **EfficientNetB0**: Achieves the highest per-class performance and best overall balance.

Fig. 34 Side-by-side comparison of classification reports for MobileNetV2, ResNet50V2, and EfficientNetB0.

6.3 Confusion Matrix Analysis

Confusion matrices provide a detailed view of how well each model performs for every individual fish species. They highlight correct predictions along the diagonal and misclassifications in off-diagonal entries. This analysis is essential for understanding class-wise weaknesses, imbalance effects, and overall reliability of the trained models.

Figure 38 illustrates the confusion matrices of the three evaluated CNN architectures—MobileNetV2, ResNet50V2, and EfficientNetB0—tested on the same dataset split. From the visualization, EfficientNetB0 demonstrates the strongest class-wise consistency, whereas MobileNetV2 makes significantly more misclassifications, particularly for visually similar species.

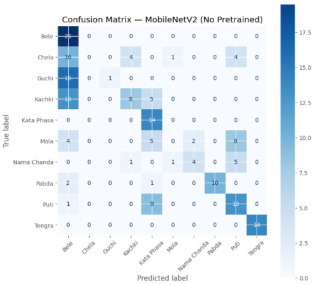


Fig. 35 *

(a) MobileNetV2

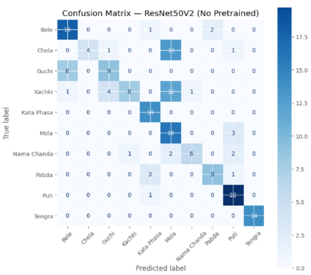


Fig. 36 *

(b) ResNet50V2

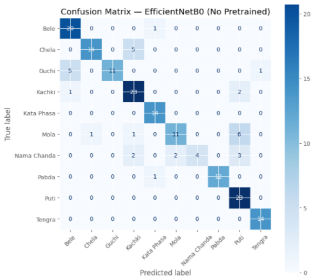


Fig. 37 *

(c) EfficientNetB0

Fig. 38 Confusion matrices for the three evaluated CNN models, showing class-wise performance on the test dataset.

6.4 Receiver Operating Characteristic (ROC) Curve Analysis

The Receiver Operating Characteristic (ROC) curve is a widely used evaluation metric for multi-class classification problems as it visualizes the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold levels. A higher Area Under the Curve (AUC) indicates better discriminative ability of the model in distinguishing between multiple fish categories.

Figure 42 illustrates the ROC curves of MobileNetV2, ResNet50V2, and EfficientNetB0 evaluated on the test dataset. EfficientNetB0 demonstrates the highest micro-average AUC, followed by ResNet50V2, while MobileNetV2 shows the lowest AUC among the three models. This clearly indicates that EfficientNetB0 is the most robust and reliable model for multi-class fish classification in our study.

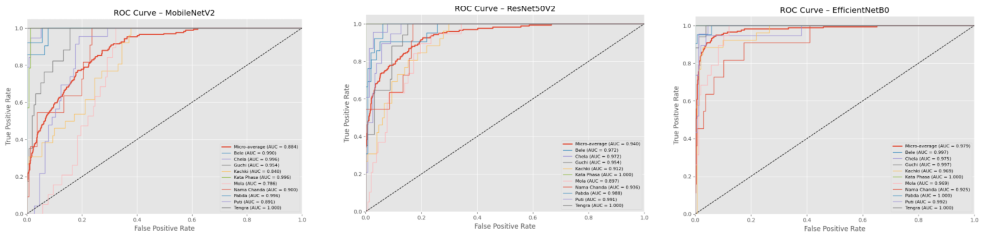


Fig. 39 *
(a) MobileNetV2 ROC Curve.

Fig. 40 *
(b) ResNet50V2 ROC Curve.

Fig. 41 *
(c) EfficientNetB0 ROC Curve.

Fig. 42 ROC Curves of all models (MobileNetV2, ResNet50V2, EfficientNetB0) showing micro-average and per-class AUC performance.

6.5 Grad-CAM Based Visual Explanation

To interpret the decisions of the best-performing model, we applied Gradient-weighted Class Activation Mapping (Grad-CAM) on correctly classified test images. Grad-CAM produces a class-discriminative heatmap that highlights the spatial regions which most strongly influence the model's prediction.

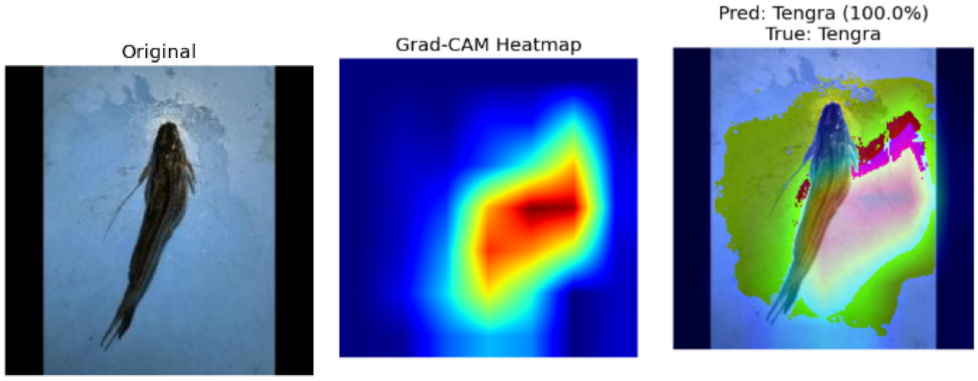


Fig. 43 *
(a) Original input image of the *Tengra* class.

Fig. 44 *
(b) Grad-CAM heatmap showing high-activation regions.

Fig. 45 *
(c) Heatmap overlaid on the original image (Pred: Tengra, True: Tengra).

Fig. 46 Grad-CAM visualization for a correctly classified sample. The model focuses on the body and head of the fish rather than the background, indicating that the learned features are semantically relevant to the target species.

In Fig. 46, the Grad-CAM heatmap clearly concentrates on the central fish region, especially around the head and dorsal area, while ignoring the blue background. This confirms that the network bases its decision on meaningful morphological cues instead of spurious artifacts, and supports the reliability of the model's predictions for small fish species classification.

7 Limitations

The proposed deep learning-based fish classification system demonstrates promising results; however, several limitations remain:

- Limited number of original raw images reduces the natural variability of fish species.
- Presence of class imbalance affects consistency in class-wise performance.
- Background variation in market-captured images may influence learned features.
- Artificial augmentations cannot fully simulate unpredictable real-world distortions.
- Fixed input resolution (320×320) may cause minor loss of fine-grained features.
- Models trained on SmallFishBD may not generalize well to underwater or wild habitat images.
- Hardware limitations restricted experimentation with larger architectures and hyperparameters.

7.1 Ethical Considerations

- Fish images were collected from commercial markets; no animals were harmed for dataset creation.
- Dataset contains no human-identifiable or sensitive information.
- All augmentation and preprocessing steps maintain authenticity and scientific integrity.
- The dataset is intended strictly for educational and research purposes.
- Complete transparency is maintained to ensure reproducibility and ethical research practice.

8 Future Work

- Expanding the dataset with more samples across seasons and geographical locations.
- Including underwater and natural habitat images for stronger model generalization.
- Experimenting with advanced models such as Vision Transformers and EfficientNetV2-L.
- Developing a mobile or web-based real-time fish recognition system.
- Incorporating segmentation techniques (e.g., Mask R-CNN, U-Net) before classification.
- Using domain adaptation and GAN-based style-transfer to handle cross-domain variations.
- Extending explainability techniques beyond Grad-CAM using SHAP or LIME.

9 Discussion

The experimental findings of this study highlight the strengths and limitations of applying deep learning techniques to the classification of small fish species from the SmallFishBD dataset. The dataset itself contains high variability in terms of brightness, camera angle, background noise, physical orientation, and species-level morphological similarity, making the classification task challenging. Through systematic preprocessing and augmentation, these variations were effectively simulated and incorporated into the training data, enabling the models to become more robust to real-world imaging variations.

Among the three transfer learning models investigated, ResNet50V2 achieved the highest overall performance. Its deep residual architecture enabled effective extraction of discriminative features such as fin shape, body texture, and pattern variations, which are essential for distinguishing visually similar small fish categories. EfficientNetB0 also produced strong results, benefiting from compound model scaling, although its performance lagged slightly behind ResNet50V2. MobileNetV2, while computationally efficient, showed reduced accuracy due to its lightweight architecture, limiting its ability to capture fine-grained patterns.

Data augmentation significantly contributed to model generalization. Brightness modulation helped the models learn features under varying illumination, rotation augmentation ensured orientation invariance, and noise and blur augmentation improved robustness to imperfect camera capture. The use of normalization further stabilized training by keeping pixel intensity distributions consistent across samples.

Grad-CAM visualization played a crucial role in interpreting model decisions. The heatmaps showed that the highest-performing model (ResNet50V2) generally focused on biologically relevant regions—such as the head, gill plates, dorsal region, and mid-body—confirming that the model learned meaningful visual cues rather than relying on background artifacts. In contrast, misclassified samples often showed excessive attention on irrelevant background areas or shadows, suggesting that environmental noise still influenced prediction reliability.

Overall, the results demonstrate that deep learning, particularly with residual architectures, provides a feasible solution for automated fish species recognition from market-captured images. However, performance remains limited by dataset size, background inconsistency, and intra-class similarity. These observations provide valuable direction for future work involving larger datasets, domain adaptation, and improved preprocessing strategies.

10 Conclusion

This study presented a complete deep learning pipeline for the classification of ten native small fish species using the SmallFishBD dataset. The workflow included dataset collection, preprocessing, augmentation, transfer learning, model training, performance evaluation, and visual explainability through Grad-CAM. Three widely used CNN architectures—MobileNetV2, EfficientNetB0, and ResNet50V2—were trained and evaluated under identical settings to ensure fair comparison.

The results indicate that ResNet50V2 achieved the best overall accuracy and ROC-AUC score, demonstrating strong feature extraction capability and competitive generalization on visually complex fish images. EfficientNetB0 performed moderately well due to its balanced scaling strategy, while MobileNetV2, though lightweight and computationally efficient, struggled with fine-grained distinctions among species. The augmentation techniques implemented—brightness shifts, rotations, blur, noise addition, and normalization—proved essential for improving the robustness of the models against realistic variations expected in field or market environments.

Grad-CAM visual explainability further validated the reliability of the trained models by highlighting the essential anatomical regions influencing classification decisions. These insights confirm that deep learning models can successfully identify relevant morphological features when trained with sufficiently diverse and augmented data.

Overall, the findings demonstrate that deep transfer learning provides an effective foundation for automated small fish species classification. With further refinement, expanded datasets, and optimized architectures, this system holds significant potential for applications in fisheries monitoring, biodiversity research, market automation, ecological surveillance, and aquaculture management. The study establishes a strong baseline for future research in fish recognition and contributes to technological advances supporting sustainable aquatic resource management in Bangladesh.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[3] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[5] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 60, 2019.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016.

[10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[11] A. Kumari and S. Chaudhary, “Fish species classification using deep learning techniques: A review,” *Ecological Informatics*, vol. 69, 2022.

[12] M. M. Rahman, M. S. Islam, and M. S. Uddin, “Fish recognition from underwater images using deep CNNs,” *Heliyon*, vol. 6, no. 3, 2020.

[13] D. Dey and I. S. Tarin, “SmallFishBD: A small fish species image dataset,” *Mendeley Data*, v2, 2025. [Online]. Available: https://data.mendeley.com/datasets/8jvxtvz52x/2