

# ■ Multiclass Fish Image Classification - Project Documentation

This project focuses on classifying fish images into multiple categories using deep learning models. It includes building a Convolutional Neural Network (CNN) from scratch and leveraging transfer learning with pre-trained models to improve performance. The final solution is deployed as a Streamlit web application that allows users to upload fish images and receive predictions along with confidence scores.

**Skills Acquired:** Deep Learning, Python, TensorFlow/Keras, Data Preprocessing, Transfer Learning, Model Evaluation, Visualization, and Model Deployment with Streamlit.

**Domain:** Image Classification

**Problem Statement:** The objective is to accurately classify fish images into their respective categories using deep learning. The project also focuses on comparing CNN and multiple pre-trained models to determine the most effective approach and deploying the best-performing model.

## **Business Use Cases:**

- 1 Enhanced Accuracy: Identify the best architecture for fish image classification.
- 2 Deployment Ready: User-friendly web application for real-time predictions.
- 3 Model Comparison: Evaluate metrics to choose the most effective model.

## **Approach:**

- 1 Data Preprocessing and Augmentation: Rescale images and apply transformations like rotation, zoom, and flipping.
- 2 Model Training: Train CNN from scratch and fine-tune five pre-trained models (VGG16, ResNet50, MobileNet, InceptionV3, EfficientNetB0).
- 3 Model Evaluation: Compare accuracy, precision, recall, F1-score, and confusion matrix across models.
- 4 Deployment: Build a Streamlit app for image upload and prediction.
- 5 Documentation: Prepare detailed README, PDF documentation, and GitHub repository.

**Dataset:** The dataset contains images of various fish species, organized into folders by category. Loaded using TensorFlow's ImageDataGenerator for efficient preprocessing.

## **Project Deliverables:**

- 1 Trained models in .h5 or .pkl format.
- 2 Streamlit web application for real-time classification.
- 3 Python scripts for training, evaluation, and deployment.
- 4 Comparison report with metrics and insights.
- 5 GitHub repository with detailed documentation.

**Installation & Usage:** 1. Clone the repository. 2. Install dependencies from requirements.txt. 3. Run the Streamlit app using: `streamlit run app.py` 4. Upload a fish image to get predictions.

This project demonstrates the application of deep learning and transfer learning in image classification. By comparing multiple architectures and deploying the best-performing model, it delivers a practical solution for automated fish species identification, which can be extended to other domains.