

Project Title	Multiclass Fish Image Classification
Skills take away From This Project	Deep Learning, Python, TensorFlow/Keras, Streamlit, Data Preprocessing, Transfer Learning, Model Evaluation, Visualization, and Model Deployment.
Domain	Image Classification

Problem Statement:

This project focuses on classifying fish images into multiple categories using deep learning models. The task involves training a CNN from scratch and leveraging transfer learning with pre-trained models to enhance performance. The project also includes saving models for later use and deploying a Streamlit application to predict fish categories from user-uploaded images.

Business Use Cases:

1. Enhanced Accuracy: Determine the best model architecture for fish image classification.
 2. Deployment Ready: Create a user-friendly web application for real-time predictions.
 3. Model Comparison: Evaluate and compare metrics across models to select the most suitable approach for the task.
-

Approach:

- Data Preprocessing and Augmentation
 - Rescale images to [0, 1] range.
 - Apply data augmentation techniques like rotation, zoom, and flipping to enhance model robustness.
- Model Training
 - Train a CNN model from scratch.
 - Experiment with five pre-trained models (e.g., VGG16, ResNet50, MobileNet, InceptionV3, EfficientNetB0).
 - Fine-tune the pre-trained models on the fish dataset.
 - Save the trained model (max accuracy model) in .h5 or .pkl format for future use.
- Model Evaluation
 - Compare metrics such as accuracy, precision, recall, F1-score, and confusion matrix across all models.
 - Visualize training history (accuracy and loss) for each model.
- Deployment
 - Build a Streamlit application to:
 - Allow users to upload fish images.
 - Predict and display the fish category.
 - Provide model confidence scores.
- Documentation and Deliverables
 - Provide comprehensive documentation of the approach, code, and evaluation.
 - Create a GitHub repository with a detailed README.

Dataset

- The dataset consists of images of fish, categorized into folders by species. The dataset is loaded using TensorFlow's ImageDataGenerator for efficient processing.
- Dataset: [Data as Zip file](#)

Project Deliverables

1. Trained Models: CNN and pre-trained models saved in .h5 or .pkl format.

2. Streamlit Application: Interactive web app for real-time predictions.
 3. Python Scripts: For training, evaluation, and deployment.
 4. Comparison Report: Metrics and insights from all models.
 5. GitHub Repository: Well-documented codebase.
-

Project Guidelines:

- **Follow coding standards:** Consistent naming conventions, modular code.
- **Data validation:** Ensure all data is accurate and complete.