Multiclass Fish Image Classification

An Intelligent AI System for Identifying Fish Species

BY

PRAVEEN.....

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.

INTRODUCTION OF THE PROJECT

This project presents a complete end-to-end AI workflow, including:

- Preprocessing and augmenting fish images.
- Training 5 different models (CNN + Transfer Learning).
- Comparing metrics to identify the best model.
- Deploying the selected model using Streamlit.

Workflow (End-to-End Project Pipeline)

1. **Data Collection & Organization:**
Images are sorted species-wise into folders.
2. **Data Preprocessing & Augmentation:**
Resize → Rescale → Rotation → Zoom → Flip
3. **Model Training:**
Train 5 models:
CNN (baseline), VGG16, ResNet50, InceptionV3, EfficientNetB0, MobileNetV2
4. **Evaluation & Comparison:**
Accuracy curve, loss curve, confusion matrix, F1-score
5. **Best Model Selection:**
6. **Deployment:**
Streamlit-based prediction system
7. **User Interaction:**
Upload → Predict → Confidence output

CODE REASONING (WHY EACH CODE EXISTS):

Data loading: Organizes classes & labels automatically
ImageDataGenerator: Normalizes and augments data
Transfer Learning import: Access pretrained feature extractors
Freezing layers: Preserve pretrained knowledge
Custom layers: Flatten + Dense + Dropout = learns fish-specific patterns
Compile: Adam optimizer + categorical loss for multiclass problem
Training: Learns patterns from dataset
Saving model: Export best model for deployment
Streamlit code: Handles UI, image preprocessing, prediction, confidence display

MODEL TRAINING PIPELINE:

[Input Images]
↓
[Resize 224x224]
↓
[Normalize 0–1]
↓
[Data Augmentation]
↓
[Transfer Learning Models]
(VGG16, ResNet50, InceptionV3, EfficientNetB0, MobileNetV2)
↓
[Feature Extraction]
↓
[Custom Layers]
(Dense + Dropout + Softmax)
↓
[Backpropagation]
↓
[Trained Model]

EXPLANATION:

This diagram shows how raw fish images are transformed step-by-step into a fully trained deep learning model. Transfer learning accelerates the training and improves performance.

TRAINING (MODEL DEVELOPMENT PHASE):

1.CNN from Scratch (Baseline)
2.VGG16
3.ResNet50
4.InceptionV3
5.EfficientNetB0
6.MobileNetV2 (Best model)
Each model was trained using augmented data to improve generalization.
MobileNetV2 delivered the highest validation accuracy and fastest inference.

Why MobileNetV2 was selected:

- Lightweight architecture
- Fast inference
- 98%+ validation accuracy
- Best generalization among tested models

TRAINING RESULTS (MOBILENETV2)

Training Accuracy: 98.19%
Validation Accuracy: 98.35%
Training Loss: 0.0604
Validation Loss: 0.0514
Observations:
- Fast convergence
- Very stable validation curve
- Good generalization
- No overfitting

TEST ACCURACY:

A separate held-out test dataset was used to measure true model performance.
✓ Confirms model generalization
✓ Proves the model works on unseen images
✓ Validates the final selected AI model
MobileNetV2 achieved **high test accuracy**, making it deployment-ready.

PREDICTIONS:

Prediction flow:
1. Upload image
2. Resize → 224×224
3. Convert to array
4. Normalize (0–1)
5. Predict using MobileNetV2
6. Show:
    1. Species name
    2. Confidence score
    3. Probability for all classes

CONFUSION MATRIX:

The confusion matrix shows:
- Correct predictions (diagonal)
- Misclassifications (off-diagonal)
- Class-wise strengths & weaknesses

Helps identify which species need better augmentation or more data.

CONFUSION MATRIX INTERPRETATION DIAGRAM LAYOUT:

```
                PREDICTED →
             Class1   Class2   Class3
ACTUAL ↓
Class1      [ TP ]   [ FP ]   [ FP ]
Class2      [ FN ]   [ TP ]   [ FP ]
Class3      [ FN ]   [ FN ]   [ TP ]
```

A confusion matrix visually compares actual vs predicted classes.
Diagonal values = correct predictions.
Off-diagonal values = misclassifications.

CLASSIFICATION REPORT:

Includes:

- **Precision → Correctness of predictions**
- **Recall → Ability to detect actual positives**
- **F1-score → Balance between precision & recall**
- **Support → Number of samples per class**

Essential for multiclass evaluation.

JSON MAPPING (CLASS LABELS):

During training:

```
0 → Bass
1 → Salmon
2 → Tuna
...
```

JSON file stores class names for prediction stage.

During deployment:

- **Model outputs label numbers**
- **JSON maps them back into fish species names**

Essential for correct UI display.

DEPLOYMENT (STREAMLIT APPLICATION):

Streamlit features:

✓ Upload any fish image
✓ Automatic preprocessing
✓ Real-time fish species prediction
✓ Confidence progress bar
✓ JSON probability output

Converts the trained model into a usable application.

BUSINESS USE CASE 1: ENHANCED ACCURACY:-

Determine the best model architecture for fish image classification?

Solution Provided by Our Project:

To achieve enhanced accuracy, we trained **multiple models**:

- CNN from scratch
- VGG16
- ResNet50
- InceptionV3
- EfficientNetB0
- MobileNetV2

Each model was evaluated based on:

✓ Validation accuracy
✓ Precision & F1-score
✓ Loss curve
✓ Confusion matrix

Finally, **MobileNetV2 achieved the highest accuracy (98%)**, making it the most reliable architecture.

➞ **This satisfies the business requirement of Enhanced Accuracy.**

BUSINESS USE CASE 2: DEPLOYMENT READY

Create a user-friendly web application for real-time predictions?

Solution Provided by Our Project:

To meet this requirement, we created a **Streamlit web application**:

✓ Simple and user-friendly UI
✓ Drag-and-drop image upload
✓ Real-time fish species prediction
✓ Confidence meter
✓ Probability distribution for all classes

The backend uses the **final selected MobileNetV2 model**, while the frontend is built using Streamlit.

➞ **This makes the model deployment-ready and easily accessible.**

BUSINESS USE CASE 3: MODEL COMPARISON

Evaluate and compare metrics across models to select the most suitable approach for the task?

Solution Provided by Our Project:

Our project compares all trained models using:

✓ Accuracy
✓ Precision
✓ Recall
✓ F1-score
✓ Confusion Matrix
✓ Training & Validation curves
✓ Inference speed

This comparison ensures the final selected model is not based on guesswork but **data-driven analysis.**

MobileNetV2 was chosen because:

•Highest validation accuracy
•Lowest validation loss
•Best generalization
•Fastest prediction time

➞ **This fulfills the business need for proper Model Compariso**

CONCLUSION & FUTURE WORK:

Conclusion:

MobileNetV2-based fish classifier achieved high accuracy and performed best among all trained models. The system is fully deployed using Streamlit for real-time usage, demonstrating a complete AI pipeline.

Future Work:

- Add more diverse fish datasets
- Improve augmentation (brightness, contrast)
- Deploy model on mobile using TFLite
- Add real-time video detection
- Explore Vision Transformers (ViT)

THANK YOU...