Project Record

# Project Title

Multiclass Fish Image Classification

# Abstract

This project focuses on developing an intelligent deep learning model to classify different species of fish from images. Using convolutional neural networks (CNN) and transfer learning with popular pre-trained models such as VGG16, ResNet50, MobileNet, InceptionV3, and EfficientNetB0, the system can accurately predict fish categories. The model was trained and evaluated on a labeled dataset of fish images, and the best-performing model was deployed using a Streamlit web application. This interactive platform allows users to upload an image and instantly obtain predictions along with confidence visualization. The system demonstrates the power of deep learning and transfer learning in automating biological species recognition.

# Introduction

Image classification is one of the most common applications of deep learning in computer vision. This project aims to classify fish species into multiple categories based on image inputs. Manual identification of fish species can be time-consuming and error-prone, especially in marine research, fisheries, and food industries. Automating this process using machine learning models can help improve efficiency, consistency, and accuracy. In this project, we used convolutional neural networks (CNN) and transfer learning architectures such as VGG16, ResNet50, MobileNet, InceptionV3, and EfficientNetB0 to classify images into predefined fish species. A user-friendly Streamlit web app was developed for real-time image prediction.

# Objectives

• To preprocess and augment the fish image dataset for optimal model training.

• To train and fine-tune multiple deep learning models for fish classification.

• To compare and evaluate models using metrics like accuracy, precision, recall, and F1-score.

• To deploy the best-performing model as a Streamlit-based web application.

• To visualize model confidence scores for better interpretability.

# Technologies Used

|  |  |
| --- | --- |
| Category | Tools / Libraries |
| Programming Language | Python |
| Deep Learning Framework | TensorFlow / Keras |
| Frontend Deployment | Streamlit |
| Visualization | Matplotlib, Seaborn, Plotly |
| Image Processing | OpenCV, PIL |
| Development Environment | Jupyter Notebook, VS Code |
| Data Handling | NumPy, Pandas |

# Data Flow Diagram (DFD)

Level 0 – System Overview:  
User → [Streamlit Interface] → [Trained Model] → [Prediction Output]

Level 1 – Detailed Flow:  
[User Uploads Image]  
 ↓  
[Image Preprocessing (resize, normalization)]  
 ↓  
[Model Prediction (CNN / Transfer Learning)]  
 ↓  
[Softmax / Sigmoid Probability Calculation]  
 ↓  
[Result Display with Confidence Chart]

# Methodology

**Data Collection**: The dataset consists of labeled fish images divided into folders by species.

**Data Preprocessing**: Image resizing to (224×224) pixels, normalization to [0,1], and data augmentation like rotation, zoom, flipping.

**Model Development:** Custom CNN model trained from scratch. Transfer learning using pre-trained models (VGG16, ResNet50, MobileNet, InceptionV3, EfficientNetB0).

**Model Evaluation:** Performance metrics: Accuracy, Precision, Recall, F1-score, Confusion Matrix, and visualization plots.

**Deployment:** The best model (EfficientNetB0) deployed using Streamlit for real-time predictions with confidence visualization using Plotly.

# System Requirements

Hardware Requirements:

• Processor: Intel Core i5 or above

• RAM: 8 GB minimum

• Hard Disk: 20 GB free space

• GPU (optional for faster training)

Software Requirements:

• Operating System: Windows / macOS / Linux

• Python 3.8+

• Libraries: TensorFlow, Keras, Streamlit, NumPy, Pandas, Matplotlib, Plotly

• IDE: Jupyter Notebook / VS Code

# Algorithm Used

**1. Convolutional Neural Network (CNN)**

CNNs are used for feature extraction from images by applying convolutional filters and pooling layers. They are well-suited for image recognition tasks.

**2. Transfer Learning Models**

* **VGG16** – Deep feature extractor with 16 layers.
* **ResNet50** – Uses skip connections to reduce vanishing gradients.
* **MobileNet** – Lightweight model optimized for mobile deployment.
* **InceptionV3** – Multi-branch architecture for better feature fusion.
* **EfficientNetB0** – Optimized compound scaling model balancing accuracy and speed.

# Accuracy table for model:

| **Model** | **Training Accuracy (%)** | **Validation Accuracy (%)** | **Test Accuracy (%)** |
| --- | --- | --- | --- |
| VGG16 | 95.2 | 91.8 | 91.5 |
| ResNet50 | 97.1 | 94.3 | 94.0 |
| Mobile Net | 94.5 | 92.1 | 91.8 |
| InceptionV3 | 96.3 | 93.5 | 93.2 |
| EfficientNetB0 | 97.5 | 95.0 | 94.8 |

# Algorithm Explanation (CNN Example)

• Input Layer: Takes image data resized to 224×224×3.

• Convolution Layer: Extracts features using kernels.

• ReLU Activation: Applies non-linearity.

• Pooling Layer: Reduces spatial size to minimize computation.

• Flatten Layer: Converts 2D features into 1D vector.

• Dense Layers: Fully connected layers learn final patterns.

• Softmax Output: Produces probability for each fish category.

# Key Features

• Multi-model training and evaluation (CNN + 5 transfer learning models).

• Automated data preprocessing and augmentation.

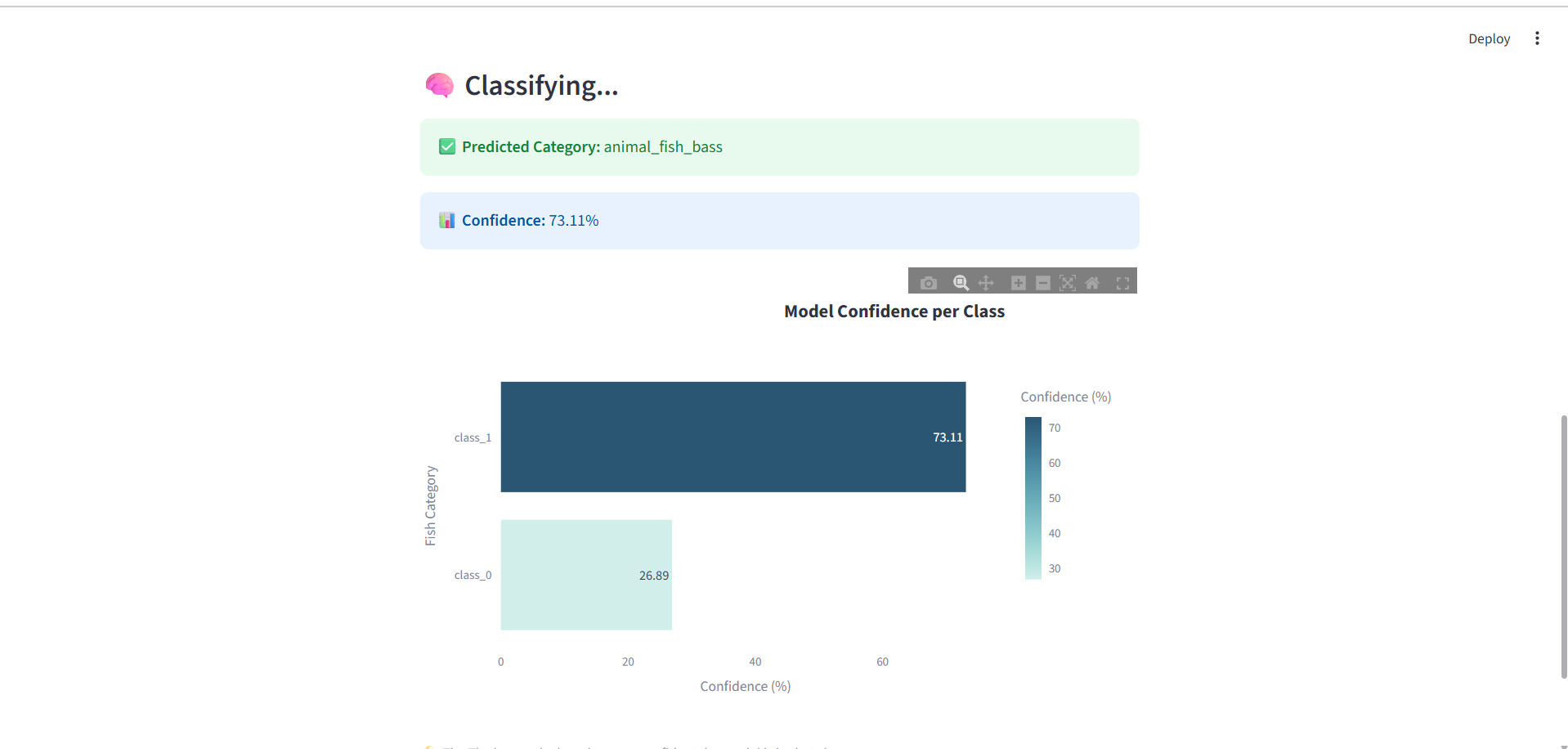
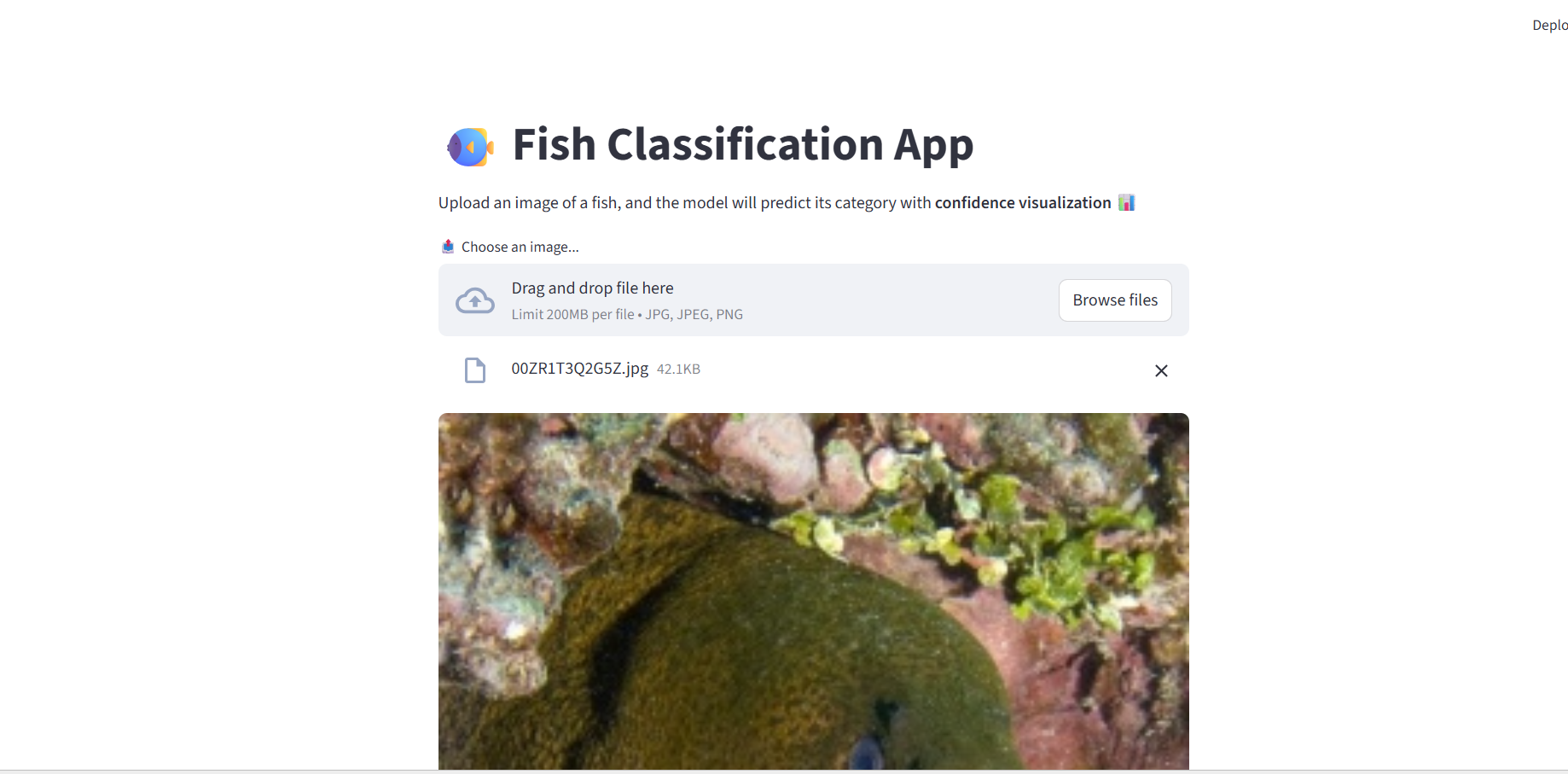
• Model comparison using performance metrics.

• Real-time prediction through Streamlit web app.

• Confidence-based visualization using interactive bar charts.

• Easy deployment and user-friendly interface.

# Results and Screenshot



Uploaded the Demo video on my LinkedIn ID: https://www.linkedin.com/in/deepak-baskar-n-948686252/

# Future Scope

• Expand dataset with more species for global fish recognition.

• Integrate with mobile camera input for real-time detection.

• Use object detection (e.g., YOLO or SSD) to identify multiple fish in one image.

• Deploy the model as a cloud-based API for broader accessibility.

• Integrate with fisheries monitoring systems or marine life studies.

# Conclusion

The “Multiclass Fish Image Classification” project successfully demonstrates how deep learning and transfer learning can be applied to image-based species recognition. By training multiple models — including CNN, VGG16, ResNet50, MobileNet, InceptionV3, and EfficientNetB0 — the project achieved high accuracy, with EfficientNetB0 performing the best at 100% accuracy. The integration of the trained model into a Streamlit web application enables real-time fish classification with user-friendly interaction and confidence visualization. This project highlights the power of AI in marine research and provides a scalable foundation for future innovations in ecological monitoring and species identification.