🐟 Fish Species Image Classification – Deep Learning Project

# 📌 Overview

This project builds an image classification system to identify fish species from photographs using Convolutional Neural Networks (CNN) and Transfer Learning (MobileNetV2). It supports model training, evaluation, and deployment via a Streamlit web app.

# 📂 Dataset Description

- \*\*Classes:\*\* 11 fish species  
- \*\*Structure:\*\*

dataset/  
├── train/  
├── val/  
└── test/

Each folder contains 11 subfolders (one for each fish species), with image files.

# 🔍 Problem Statement

Classify a given fish image into one of 11 categories with high accuracy and robustness, despite variations in lighting, orientation, and pose.

# 🧠 Deep Learning Approach

- \*\*Base Model:\*\* MobileNetV2 (pre-trained on ImageNet)  
- \*\*Additions:\*\*  
 - Global Average Pooling  
 - Dense Layer (128 units, ReLU)  
 - Dropout (0.3)  
 - Output Layer (Softmax for 11 classes)

- \*\*Input Size:\*\* 160 × 160 × 3  
- \*\*Optimizer:\*\* Adam  
- \*\*Loss Function:\*\* Categorical Crossentropy  
- \*\*Batch Size:\*\* 64  
- \*\*Epochs:\*\* 10  
- \*\*Early Stopping:\*\* Enabled (patience = 2)

- \*\*Data Augmentation:\*\*  
 - Rotation (±20°)  
 - Width/height shift (±10%)  
 - Zoom (up to 15%)  
 - Horizontal flip  
 - Rescaling (1./255)

# 📊 Model Evaluation

- \*\*Metrics Reported:\*\* Accuracy, Precision, Recall, F1-score (per class), Confusion Matrix

- \*\*Evaluation Code:\*\*  
```python  
from sklearn.metrics import classification\_report, confusion\_matrix  
Y\_pred = model.predict(test\_generator)  
y\_pred = np.argmax(Y\_pred, axis=1)  
y\_true = test\_generator.classes  
print(classification\_report(y\_true, y\_pred, target\_names=test\_generator.class\_indices.keys()))  
```

- \*\*Confusion Matrix Plot:\*\*  
```python  
from sklearn.metrics import ConfusionMatrixDisplay  
cm = confusion\_matrix(y\_true, y\_pred)  
disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=test\_generator.class\_indices)  
disp.plot()  
```

- \*\*Training Accuracy Plot:\*\*  
```python  
plt.plot(history.history['accuracy'], label='Train Accuracy')  
plt.plot(history.history['val\_accuracy'], label='Val Accuracy')  
plt.title('Accuracy over Epochs')  
plt.xlabel('Epoch')  
plt.ylabel('Accuracy')  
plt.legend()  
```

# 📦 Streamlit App Deployment

- Upload an image of a fish  
- Predict the species  
- Show confidence scores for all 11 classes

- \*\*Sample Code:\*\*  
```python  
model = tf.keras.models.load\_model('fish\_classifier\_fast.h5')  
image = Image.open(uploaded\_file).resize((160, 160)).convert("RGB")  
img\_array = np.expand\_dims(np.array(image) / 255.0, axis=0)  
predictions = model.predict(img\_array)[0]  
predicted\_class = class\_names[np.argmax(predictions)]  
```

- \*\*Run Locally:\*\*  
```bash  
streamlit run app/app.py  
```

# 💾 Directory Structure

```  
fish-classification/  
├── app/  
│ └── app.py  
├── model/  
│ └── fish\_classifier\_fast.h5  
├── dataset/  
├── notebooks/  
│ └── training\_and\_evaluation.ipynb  
├── images/  
│ ├── accuracy\_plot.png  
│ └── confusion\_matrix.png  
├── requirements.txt  
└── README.md  
```

# 🎓 Credits

- Model: MobileNetV2  
- Frameworks: TensorFlow, Keras, Streamlit  
- Data: Provided fish image dataset