Project Report: Fighter Jet Image Classification using Deep Learning

Project Name: Fighter Jet Image Classification **Technique:** Convolutional Neural Networks (CNN) & Transfer Learning **Deployment:** Streamlit Web Application (Hosted on Render)

---

1. Problem Statement

The objective of this project was to develop a Deep Learning model capable of accurately identifying and classifying various types of fighter jets based on visual imagery.

Challenges Identified:

- **Dataset Limitations:** The dataset consisted of approximately 800 images across 20 distinct classes (e.g., F22, B52, Rafale).
- **Data Scarcity:** With only 40 images per class, the risk of model overfitting was significant.
- **Fine-Grained Classification:** Distinguishing between different fighter jets is a complex visual task due to similarities in aerodynamic design, color, and background environments (sky/tarmac).

Goal: To achieve a validation accuracy between **85% and 93%** and deploy the solution as a functional web application for real-time user predictions.

---

2. Approach and Methodology

To address the challenges of the small dataset and complex classification task, a phased approach was adopted:

A. Data Preprocessing & Augmentation

- **Inspection:** The dataset was analyzed and found to be perfectly balanced with 40 images per class, eliminating the need for class imbalance handling techniques.
- **Augmentation:** To artificially expand the training set and improve robustness, aggressive Data Augmentation was applied using ImageDataGenerator. Techniques included random rotations (30°), width/height shifts, zooming, and horizontal flips.

- **Preprocessing:** Images were resized to **224x224 pixels** to match the input requirements of standard pre-trained architectures.

B. Model Architecture Evolution

Three distinct modeling strategies were tested to maximize performance:

1. **Baseline Custom CNN:** A standard Convolutional Neural Network was built from scratch.
   - *Result:* Failed to generalize, achieving only ~15% accuracy.
   - *Insight:* The dataset was too small to learn complex features from scratch.
2. **Transfer Learning (MobileNetV2):** A pre-trained lightweight model.
   - *Result:* Improved accuracy to ~50% but plateaued.
   - *Insight:* The model required more capacity to understand specific aircraft features.
3. **Transfer Learning (EfficientNetB0) - *Final Selection*:**
   - **Architecture:** We utilized **EfficientNetB0**, a state-of-the-art model pre-trained on ImageNet.
   - **Strategy:** The model was "unfrozen" to allow fine-tuning of all layers. This enabled the network to adjust its pre-learned filters specifically for the textures and shapes of fighter jets.
   - **Optimization:** The Adam optimizer with a low learning rate (0.0001) was used to refine weights without destroying pre-learned knowledge.

C. Deployment

The final model was deployed as a web application to demonstrate real-world utility.

- **Framework: Streamlit** was chosen for the frontend to create a user-friendly interface.
- **Hosting:** The application was deployed on **Render** cloud services, linked via a GitHub repository for Continuous Deployment (CD).

---

3. Insights and Results

- **Model Performance:** The EfficientNetB0 model successfully met the project target, achieving a validation accuracy within the required **85-93% range**.

- **Impact of Transfer Learning:** Switching from a custom CNN to Transfer Learning was the decisive factor. It allowed the model to leverage millions of pre-learned patterns, compensating for the small size of our specific dataset.
- **Fine-Tuning:** Unfreezing the base model layers was critical. It allowed the model to transition from recognizing generic objects (like cars or birds) to understanding the subtle aerodynamic differences between an *F-22 Raptor* and an *F-35 Lightning*.

---

4. Recommendations & Future Scope

Based on the project outcomes, the following recommendations are made for future improvements:

1. **Dataset Expansion:** While data augmentation helped, collecting more real-world images (increasing to 100+ per class) would significantly further improve robustness.
2. **Ensemble Methods:** Combining predictions from multiple models (e.g., EfficientNet + ResNet) could push accuracy beyond 95%.
3. **Explainability (Grad-CAM):** Future iterations could include "Heatmaps" in the web app to show users exactly *which part* of the jet the model is looking at to make its decision.

Project Links

- **GitHub Repository:** https://github.com/iamkarthikfromba/Newton_School_Assignment_Fire_Fighter_Planes
- **Live Web Application:** https://newton-school-assignment-fire-fighter.onrender.com/