**Image Classification Project Report**

**Objective**

The goal of this project was to build and deploy an image classification model using deep learning techniques. The model was trained on the CIFAR-10 dataset and deployed via a FastAPI-based REST API, with a Streamlit UI for user interaction.

**Task Completion Checklist**

**✅ Part 1: Data Preprocessing**

✔️ Loaded the CIFAR-10 dataset. ✔️ Performed Exploratory Data Analysis (EDA). ✔️ Resized images to (128,128). ✔️ Normalized pixel values. ✔️ Applied data augmentation (flipping, rotation, brightness adjustments). ✔️ Converted images into PyTorch tensors. ✔️ Split dataset into training, validation, and test sets.

**✅ Part 2: Model Training & Evaluation**

✔️ Implemented ResNet-18 as the CNN model. ✔️ Used PyTorch for training and evaluation. ✔️ Applied early stopping and learning rate scheduling. ✔️ Evaluated model using:

* Accuracy
* Precision & Recall
* Confusion Matrix ✔️ Saved trained model in .pkl and .pth formats.

**✅ Part 3: Model Deployment**

✔️ Built a REST API using FastAPI. ✔️ Created /predict endpoint for image classification. ✔️ Implemented Basic Authentication for security. ✔️ Containerized backend API using Docker. ✔️ Tested API using Postman and CURL.

**✅ Part 4: Frontend**

✔️ Developed a Streamlit UI for easy image classification. ✔️ Allowed users to upload images and get predictions. ✔️ Connected frontend to backend API. ✔️ Dockerized the frontend for deployment.

**✅ Running the Application**

✔️ Successfully ran both backend and frontend separately. ✔️ Integrated and tested both using Docker Compose. ✔️ Verified full functionality of the application.

**✅ Bonus Points Implemented**

✔️ Added basic logging and error handling in FastAPI. ✔️ Developed a user-friendly frontend with Streamlit. ✔️ Containerized the full application for seamless deployment.

**❌ Future Enhancements (Not Implemented)**

⬜ Deploy API on AWS, GCP, Azure, or Render. ⬜ Implement Grad-CAM or SHAP for model explainability. ⬜ Optimize the model for real-time inference. ⬜ Implement JWT authentication for enhanced security.

**Conclusion**

This project successfully demonstrates an end-to-end image classification system, from data preprocessing and model training to API deployment and frontend integration. Future improvements could include deploying the API to cloud platforms and adding explainability features like Grad-CAM.