Project Report — AI Image Classifier

Intern Name: Ruthvik Gurrala
Internship Domain: Machine Learning
Organization: Skillbit Technologies
Internship Duration: 8th October 2025 – 18th November 2025

Project Overview

This project involved the end-to-end development of a full-stack AI image classification application. The system is designed to predict the category of a user-uploaded image by leveraging a deep learning model. The core of the application is a Convolutional Neural Network (CNN) that provides real-time analysis. When a user submits an image through the web interface, the application processes it and returns the top three most likely classifications along with their corresponding confidence scores, providing an intuitive and interactive user experience.

Objective

The primary objective was to design, build, and deploy a real-time image classification system. This entailed training a robust deep learning model on the CIFAR-10 dataset and integrating it seamlessly with a modern, responsive React-based web application. The goal was to create a functional prototype demonstrating a complete MLOps workflow, from data preprocessing and model training to API-based model serving and frontend integration.

Dataset and Model Training

Dataset: The project utilized the CIFAR-10 dataset, a well-established benchmark in computer vision. It consists of 60,000 32x32 color images across 10 distinct classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

Framework: The model was developed using TensorFlow with the Keras high-level API.

Model: A Convolutional Neural Network (CNN) was designed with a standard architecture comprising multiple convolution and pooling layers to extract hierarchical features, followed by dense layers for classification.

Techniques:

- **Normalization:** Pixel values were scaled to a range of [0, 1] to improve training stability and speed.
- **Dropout:** Dropout layers were strategically placed to prevent overfitting by randomly deactivating neurons during training.
- **Fine-Tuning:** The model was initially trained for 20 epochs and then fine-tuned for an additional 10 epochs to further improve its accuracy on the validation set.

Output: The fully trained and optimized model was saved as `cnn_model_finetuned.h5`.

Backend

Technology: The backend was built using FastAPI, a high-performance Python web framework chosen for its speed and ease of use.

Function: The API server is responsible for loading the saved `cnn_model_finetuned.h5` model into memory upon startup. It exposes a primary `/upload` API endpoint that accepts multipart/form-data image files, preprocesses them to match the model's input requirements, and performs inference.

Output: The API returns a JSON object containing the top three class predictions and their associated confidence percentages, which are then consumed by the frontend.

Frontend

Technology: The user interface was developed as a single-page application using React (bootstrapped with Vite) and styled with Tailwind CSS. The react-dropzone library was integrated to create an intuitive drag-and-drop file upload component.

Features: The frontend features a clean, modern, AI-themed UI with a responsive design suitable for both mobile and desktop use. Key features include a drag-and-drop upload box, an image preview panel, and an animated visualization of the model's confidence scores using progress bars.

API Communication: Axios was used to handle asynchronous API calls to the FastAPI backend, facilitating the transfer of the image file and the receipt of prediction data.

Features Implemented

- Real-time image upload and classification via a web interface.
- Confidence-based visualization with animated progress bars.
- Display of the top 3 most likely predictions.
- A fully responsive and mobile-friendly user interface.
- Clear and immediate feedback to the user during the prediction process.

Challenges Faced & Solutions

- **Model Training Time:** Initial model training was time-consuming. This was managed by leveraging a GPU-accelerated runtime environment and employing techniques like data augmentation to achieve better generalization with fewer epochs.
- **Integration Errors (CORS):** Cross-Origin Resource Sharing (CORS) errors initially prevented the React frontend from communicating with the FastAPI backend. This was resolved by implementing and correctly configuring FastAPI's CORSMiddleware to allow requests from the frontend's origin.
- **Tailwind Setup on Windows:** As in a previous project, Tailwind CSS required manual CLI initialization on the Windows development environment. The issue was resolved by following the standard manual setup procedure.
- **Frontend Debugging:** Initially, the prediction data received from the backend was not displaying correctly. This was traced to an issue in how the Axios response was being mapped to the React

state. The problem was solved by adjusting the response handling logic to correctly parse the JSON and update the component's state.

Skills & Tools Gained

This project provided hands-on experience with a wide range of modern technologies and machine learning concepts:

ML Frameworks: TensorFlow, Keras
Backend Development: FastAPI, Uvicorn, Model Serving
Frontend Development: React, Vite, Tailwind CSS, Axios, React-Dropzone
ML Techniques: CNN Architecture, Image Preprocessing, Transfer Learning/Fine-Tuning
Development Tools: Git, GitHub, Jupyter, Visual Studio Code

Learning Outcome

This project was an invaluable learning experience in practical, end-to-end AI application development. It bridged the gap between theoretical knowledge of Convolutional Neural Networks and their real-world implementation. I gained significant hands-on experience in building a CNN from scratch, serving it via a REST API, and integrating it with a dynamic frontend. The project solidified my understanding of the entire MLOps lifecycle, from model training and optimization to deployment readiness and full-stack integration.

Conclusion

I am immensely grateful to Skillbit Technologies for the opportunity to work on this challenging and rewarding project. The AI Image Classifier project significantly strengthened my deep learning knowledge, particularly in computer vision, and provided me with critical full-stack integration skills. The practical experience gained during this internship has been instrumental in preparing me for a career in machine learning and AI development.