Final Project Proposal

Project Title: Land Type Classification using Sentinel-2 Satellite Images

1. Project and Team Information

Property	Details
Track	AI & Data Science Track – Round 3
Program	Digital Egypt Pioneers Initiative
Team Leader	**Tadrous Adel William**
Team Members	Maria Ashraf Haleem, Abd-elaziz Hassan Fouad, Karen Medhat Zaher, Mohamed Essam, Mohamed Kamal
Core Dataset	EuroSat (Sentinel-2 Labeled Dataset)
Key AI Model	**EfficientNet_B0** (Deep Neural Network)
Primary Frameworks	PyTorch, Flask
Final Test Accuracy	**97.12 %**
Deployment Solution	Simple Web Interface using Flask API

2. Executive Summary and Project Goals

This project successfully developed an automated, high-accuracy system for classifying land types using multispectral satellite imagery from the European Space Agency's Sentinel-2 mission. The primary objective was to leverage **Deep Neural Networks (DNNs)** to categorize land into major classes (e.g., permanent crops, industrial areas, rivers, forest, etc.) to aid in critical applications like urban planning, resource management, and environmental monitoring.

We utilized the **EuroSat dataset** and successfully engineered a robust solution based on the **EfficientNet_B0** architecture, a state-of-the-art Convolutional Neural Network (CNN). The model achieved an outstanding **test accuracy of 97.12%** and a validation accuracy of 97%, demonstrating excellent generalization capabilities. The final model was deployed as a simple, functional API using **Flask**, making the predictive power accessible for immediate practical use.

3. Comprehensive Methodology and Technical Execution

The project adhered rigorously to the five-milestone structure, focusing on technical depth and advanced deep learning techniques.

3.1. Milestone 1: Data Collection, Exploration, and Preprocessing

- **Data Acquisition:** The **EuroSat dataset** was selected due to its high quality, public availability, and labeled classification across 10 distinct land types derived from Sentinel-2 images (13 spectral bands).
- **Preprocessing:** The raw image data underwent critical transformations to ensure model compatibility and performance. This included:
    - **Normalization:** Scaling pixel values across the multispectral bands to a consistent range for optimal neural network training.
    - **Resizing:** Ensuring all input images were resized to a consistent dimension required by the EfficientNet_B0 architecture.
    - **Data Augmentation:** Techniques (rotations, flips, scaling) were systematically applied to the training set to increase dataset diversity, thus enhancing the model's ability to generalize and reducing overfitting.

3.2. Milestone 2: Advanced Data Analysis and Model Selection

- **Spectral Band Analysis:** Exploratory analysis confirmed the crucial role of specific non-visible bands (Near-Infrared, Red Edge) for differentiating between land classes (e.g., separating healthy vegetation from bare soil or water bodies). This insight justified using all 13 spectral bands.
- **Model Selection Rationale:** After an initial experiment with a basic Custom CNN (to establish a performance baseline), the team transitioned to a sophisticated transfer learning approach. **EfficientNet_B0** was chosen as the final architecture due to its superior performance-to-efficiency ratio,

leveraging a compound scaling method to optimize model depth, width, and resolution simultaneously.

3.3. Milestone 3: Model Development, Training, and Optimization

- **Implementation:** The model was built and trained using the **PyTorch** deep learning framework, which provided flexibility for custom training loops and advanced model manipulation.
- **Training Strategy:**
  - **Transfer Learning:** A pre-trained version of EfficientNet_B0 (on ImageNet) was used, and the top layers were fine-tuned specifically for the EuroSat classification task.
  - **Hyperparameter Tuning:** A systematic approach was used to optimize the learning rate, batch size, and the **Adam Optimizer** parameters to ensure rapid convergence and optimal performance.
  - **Overfitting Prevention:** Techniques such as **Dropout Layers** and **Early Stopping** based on the validation loss were implemented to ensure the model's performance on the validation set remained high.
- **Evaluation:** The final model achieved a stellar performance:
  - **Test Accuracy: 97.12%**
  - **Validation Accuracy:** 97.00%
  - Performance was also verified across other critical metrics including **Precision, Recall, and F1-Score**, confirming robust and balanced classification across all land classes.

3.4. Milestone 4: Deployment and Monitoring (MLOps)

- **Model Deployment (API):** The final trained EfficientNet_B0 model was serialized and deployed as a RESTful API using the **Flask** web framework. This API is designed to receive a Sentinel-2 image as input and return the predicted land type.
- **User Interface:** A simple, functional web interface was developed using Flask to demonstrate the model's capability, allowing users to upload an image and view the classification result immediately.

- **MLOps Strategy:** A foundational strategy for monitoring was documented, including:
  - **Performance Tracking:** Establishing a baseline for accuracy and latency.
  - **Drift Detection:** Plan for periodically checking **data drift** (changes in incoming satellite imagery patterns) or **model drift** (decay in performance over time).
  - **Retraining Plan:** A clear strategy to trigger retraining with new or more diverse data to maintain the 97% + accuracy threshold.

4. Technical Stack and Core Libraries

Category	Tools and Libraries Used	Rationale
Deep Learning Framework	**PyTorch**	Chosen for its dynamic graph computation, flexibility, and strong community support for advanced research models like EfficientNet_B0.
Core Model Architecture	**EfficientNet_B0**	State-of-the-art model selected for its high accuracy and parameter efficiency through compound scaling.
Deployment	**Flask**	Lightweight and flexible Python web framework used to create a rapid and simple API for the model inference.
Data Handling	NumPy, Pandas	Essential libraries for array manipulation, data structure management, and statistical analysis of spectral data.
Image Processing	Torchvision, PIL, OpenCV	Used for data loading, preprocessing, augmentation, and applying required transformations.
Visualization	Matplotlib, Seaborn	Utilized for generating visual reports, confusion matrices, and analyzing spectral distributions.

5. Team Roles and Contributions

The success of the project was based on a clear division of labor, ensuring expertise in each milestone.

Student Name	Primary Roles and Key Contributions
Tadrous Adel William (Leader)	Overall project management and coordination, leading the **Deployment and MLOps phase** with Flask, and final code integration and testing.
Maria Ashraf Haleem	Extensive **Data Preprocessing** and cleaning pipeline development, implementation of data augmentation techniques, and technical documentation of Milestone 1.
Abd-elaziz Hassan Fouad	Development of the initial **Custom CNN** baseline model, meticulous focus on **Model Evaluation** metrics (F1, Precision, Recall), and interpreting confusion matrices.
Karen Medhat Zaher	Primary responsibility for **Data Exploration (EDA)**, analysis of spectral bands, and providing crucial insights for model selection based on advanced data analysis.
Mohamed Essam	Implementation and training of the **EfficientNet_B0** model using PyTorch, detailed **Hyperparameter Tuning**, and optimization of the training process.
Mohamed Kamal	Responsible for final project **Documentation** and report writing, preparing the comprehensive **Final Presentation** slides, and articulating the project's real-world applications.

6. Conclusion and Future Work

This project has successfully delivered a state-of-the-art land type classification model with an accuracy of **97.12%**, demonstrating proficiency in advanced deep learning and MLOps principles.

Future Improvements:

- **Data Integration:** Incorporate additional satellite data sources (e.g., Landsat or synthetic aperture radar/SAR data) to improve classification robustness in cloudy conditions.
- **Advanced Architectures:** Experiment with **Vision Transformer (ViT)** models for classification to explore potential improvements over CNNs.

- **Scalable Deployment:** Migrate the Flask API to a scalable cloud platform (AWS Lambda or Azure Functions) and containerize the application using Docker for enhanced robustness and scalability.