21033990	احمد سامر السيد
21088622	اسلالام محمد رجب
21122808	مصطفف عبدالرحمن رمضا ن
21033945	عبدالرحمن إيهاب صلالاح
21015239	يوسف محمد عبد التواب

Land Type Classification Using Sentinel-2 Satellite Images

AI & Data Science Track

1.Project Proposal

1.1 Project Overview:

The Land Type Classification using Sentinel-2 Satellite Images project aims to develop a deep learning-based system capable of accurately classifying different land types across Egypt using satellite imagery. The system will utilize high-resolution Sentinel-2 images and apply advanced image classification techniques to identify major land categories such as agricultural land, water bodies, urban areas, deserts, roads, and trees.

This project integrates geospatial analysis with artificial intelligence to build a robust classification model that can support real-world environmental and urban planning applications. By leveraging publicly available datasets and custom-collected satellite imagery, the project demonstrates the practical implementation of deep learning in remote sensing.

1.2 Project Objectives:

The main objectives of this project are:

To collect and preprocess Sentinel-2 satellite imagery covering different regions in Egypt.

To explore and analyze the spatial distribution of land types within the dataset.

To design and implement a deep neural network model for land classification.

To optimize and evaluate the model using appropriate performance metrics.

To deploy the trained model in a real-time environment for user interaction.

To visualize classification results through an interactive dashboard.

1.3 Project Scope:

The scope of this project includes the complete lifecycle of a deep learning-based land classification system. This involves data collection, preprocessing, model development, performance optimization, deployment, and visualization.

The project focuses specifically on land type classification within Egypt using Sentinel-2 imagery. It does not include large-scale satellite data processing infrastructure or national-scale continuous monitoring systems. Instead, it aims to develop a functional prototype capable of accurate classification and real-time predictions on selected satellite images.

The system will classify six primary land categories:

- Agricultural land
- Water bodies
- Urban areas
- Desert
- Roads
- Trees

2. Project Plan

2.1 Project Timeline:

Milestone	Duration	Key Activities
Milestone 1: Data Collection & Preprocessing	**Week 1–2**	**Collect Sentinel-2 images, perform EDA, preprocess data**
Milestone 2: Model Development & Training	**Week 3–4**	**Build CNN model, train and validate**
Milestone 3: Model Optimization	**Week 5**	**Hyperparameter tuning and refinement**

Milestone 4: Deployment & Visualization	**Week 6**	**Deploy model and build UI dashboard**
Milestone 5: Documentation & Presentation	**Week 7**	**Final report preparation and demo**

2.2 Milestones & Deliverables:

Milestone 1: Data Collection & Preprocessing

Deliverables:

- Cleaned and preprocessed dataset
- Data augmentation pipeline
- Dataset exploration report

Milestone 2: Model Development & Training

Deliverables:

- Trained deep learning model
- Validation performance metrics
- Initial evaluation report

Milestone 3: Model Optimization

Deliverables:

- Optimized model
- Hyperparameter tuning results
- Final evaluation metrics

Milestone 4: Deployment & Visualization

Deliverables:

- Deployed web-based classification system
- Interactive dashboard
- User interface for image upload

Milestone 5: Documentation & Presentation

Deliverables:

- Final technical report
- Final presentation slides
- Live system demonstration

2.3 Resource Allocation:

The project will require both technical and software resources to ensure successful implementation.

Human Resources:

AI/Deep Learning Developer (Model development and optimization)
Data Analyst (Data preprocessing and exploration)
Backend Developer (Model deployment)
UI Developer (User interface design)

Software & Tools:

Python (TensorFlow / PyTorch)
QGIS for satellite image collection
Sentinel Hub
Flask or FastAPI for deployment
Matplotlib / Plotly for visualization
GitHub for version control

3. Team Structure

3.1 Roles and Responsibilities

Data Team (2 Members)

Member 1 – Data Collection & GIS Specialist

Collect Sentinel-2 images using QGIS and Sentinel Hub.
Organize and label dataset.
Perform exploratory data analysis (EDA).

Member 2 – Data Preprocessing Engineer

Resize and normalize images.
Split dataset into training, validation, and test sets.
Apply data augmentation techniques.
Prepare data in tensor format for model training.

Model Team (2 Members)

Member 3 – Model Development Engineer

Design and implement CNN / Transfer Learning model.
Select loss function and optimizer.
Train and validate the model.

Member 4 – Model Optimization & Evaluation Specialist

Perform hyperparameter tuning.
Apply regularization techniques.

Evaluate performance using accuracy, precision, recall, F1-score, and confusion matrix.

Deployment Team (2 Members)

Member 5 – Deployment Engineer

Develop backend using Flask orFastAPI.
Integrate trained model into thesystem.
Ensure API functionality and performance.

Member 6 – UI & Visualization Engineer (Team Leader)

Technical Role:

Design userinterface for image upload.
Develop visualization dashboard for classification results.
Assist in system integration and testing.

Leadership Role:

Coordinate between Data, Model, and Deployment teams.
Monitor milestone progress.
Ensure timely submission of deliverables.
Oversee final documentation and presentation.

4. Risk Assessment & Mitigation Plan

4.1 Risk Identification:

Several potential risks may affect the project execution:

- Limited availability of high-quality satellite images.
- Imbalanced dataset across land classes.
- Overfitting during model training.
- High computational requirements.
- Deployment compatibility issues.

4.2 Risk Mitigation Strategies

To minimize project risks, the following mitigation strategies will be implemented:

Use multiple open-source datasets to increase data diversity.
Apply data augmentation techniques to balance classes.
Use regularization techniques such as dropout and batch normalization.
Implement early stopping to prevent overfitting.
Utilize cloud-based GPU resources if local hardware is insufficient.
Perform thorough testing before deployment.

5. Key Performance Indicators (KPIs)

5.1 Model Performance KPIs:

The success of the model will be evaluated using the following performance metrics:

Accuracy (Target: ≥ 90%)
Precision per class
Recall per class
F1-Score
Confusion Matrix analysis
Cross-validation consistency

5.2 System Performance KPIs

The deployed system will also be evaluated based on operational metrics:

Response time per image prediction (Target: < 3 seconds)
System uptime (Target: ≥ 95%)
Successful prediction rate
User interaction success rate