**Land Type Classification using EuroSAT Dataset**

**Documentation Report**

**1. Project Planning**

**1.1 Overview**

This project focuses on classifying different land types—such as agricultural areas, forests, urban regions, water bodies, and roads—using the **EuroSAT dataset**, which consists of Sentinel-2 satellite imagery. The project leverages deep learning techniques implemented in Python to build a scalable model capable of identifying land cover classes from multispectral data.

**1.2 Objectives**

* Utilize the EuroSAT dataset to train and evaluate land cover classification models.
* Extract and analyze spectral and spatial patterns among EuroSAT land types.
* Develop a deep neural network (CNN-based) classifier for multi-class prediction.
* Integrate classification visualization within a web-based dashboard for interactive exploration.

**1.3 Scope**

The model targets **10 distinct land cover categories** defined in the EuroSAT dataset. While the project begins with static image classification, it is extendable to geospatial mapping and temporal monitoring in future phases.

**1.4 Project Timeline**

| **Phase** | **Description** | **Duration** |
| --- | --- | --- |
| Phase 1 | Dataset exploration and preprocessing | 2 weeks |
| Phase 2 | Model selection and training | 3 weeks |
| Phase 3 | Evaluation and tuning | 2 weeks |
| Phase 4 | Web dashboard development | 2 weeks |
| Phase 5 | Testing and documentation | 1 week |

**Total Duration:** 10 weeks

**1.5 Deliverables**

* Trained classification model (TensorFlow/Keras).
* Preprocessed and labeled EuroSAT dataset subsets.
* Interactive dashboard visualizing model predictions and accuracy.
* Final report and performance metrics.

**2. Stakeholder Analysis**

**2.1 Stakeholder Identification**

| **Stakeholder** | **Role** | **Interest** | **Influence** |
| --- | --- | --- | --- |
| Project Sponsor | Provides direction and evaluation criteria | High | High |
| Data Scientist | Model training, analysis, and performance optimization | High | High |
| Web Developer | Implements visualization dashboard | Medium | High |
| End Users (Researchers, Planners) | Use results for spatial analysis and decision-making | High | Medium |
| Supervisor | Monitors progress and methodology | Medium | High |

**2.2 Communication Plan**

* Weekly updates on training progress and accuracy metrics.
* Sprint reviews for dashboard development.
* Final presentation including performance summary and demo.

**3. Dataset and Database Design**

**3.1 Dataset Overview**

* **Source:** [EuroSAT Dataset (Kaggle)](https://www.kaggle.com/datasets/apollo2506/eurosat-dataset)
* **Origin:** Sentinel-2 imagery captured by ESA.
* **Image Format:** RGB (JPEG) and optionally 13-band TIFF.
* **Resolution:** 10–60 m (resampled to 64×64 pixels).
* **Classes (10):**
  1. Annual Crop
  2. Forest
  3. Herbaceous Vegetation
  4. Highway
  5. Industrial
  6. Pasture
  7. Permanent Crop
  8. Residential
  9. River
  10. Sea/Lake

**3.2 Local Dataset Design**

The EuroSAT dataset is organized by folders per class:

/dataset/

├── AnnualCrop/

├── Forest/

├── HerbaceousVegetation/

├── Highway/

├── Industrial/

├── Pasture/

├── PermanentCrop/

├── Residential/

├── River/

└── SeaLake/

Each folder contains 2,000–3,000 labeled 64×64 images.

**3.3 Logical Data Model**

A lightweight local metadata file (SQLite or CSV) can store image paths and labels:

**Schema Example:**

TABLE Images (

id INTEGER PRIMARY KEY,

path TEXT,

label TEXT,

width INTEGER,

height INTEGER

);

**4. Model Design and Methodology**

**4.1 Data Preprocessing**

* Load RGB or multispectral bands.
* Normalize pixel intensities (0–1).
* Augment dataset (flip, rotate, zoom) to improve generalization.
* Split dataset (70% train / 15% validation / 15% test).

**4.2 Model Architecture**

A Convolutional Neural Network (CNN) is applied for image classification:

* **Input:** 64×64×3 (RGB) or 64×64×13 (multispectral)
* **Layers:** Convolution → ReLU → MaxPooling → Dropout → Dense
* **Output:** 10-way Softmax layer

Alternative: Transfer learning with **ResNet50**, **EfficientNetB0**, or **MobileNetV2** for improved accuracy.

**4.3 Evaluation Metrics**

* Accuracy, Precision, Recall, F1-Score
* Confusion Matrix
* Per-class performance

**5. UI / UX Design**

**5.1 Objective**

Provide a user-friendly web dashboard that visualizes EuroSAT classification results and evaluation metrics interactively.

**5.2 User Flow**

1. **Home Page:** Overview and model information.
2. **Image Upload or Selection:** User uploads an image or selects a random EuroSAT sample.
3. **Classification Output:** Displays predicted class with probability.
4. **Visualization Panel:** Shows confusion matrix and class distribution.
5. **Export Section:** Save prediction summary or plots.

**5.3 Design Overview**

* **Frontend:** Dash (Plotly) or Flask + HTML/CSS.
* **Core Components:**
  + File upload widget.
  + Image preview.
  + Class prediction and confidence bar.
  + Interactive charts (accuracy trends, confusion matrix).

**5.4 Color Legend**

| **Land Type** | **Color** |
| --- | --- |
| Water / Sea-Lake | Blue |
| Forest | Green |
| Annual Crop | Yellow |
| Residential / Industrial | Red |
| Highway | Gray |
| Pasture / Vegetation | Light Green |

**6. Technology Stack**

| **Component** | **Technology** |
| --- | --- |
| Programming Language | Python |
| Libraries | TensorFlow / Keras, Scikit-learn, NumPy, Matplotlib |
| Data Handling | Pandas, OpenCV |
| Web Framework | Flask / Dash (Plotly) |
| Storage | Local dataset directories + optional SQLite metadata |
| Deployment | Docker / Render / Localhost |

**7. Risk and Mitigation**

| **Risk** | **Impact** | **Mitigation** |
| --- | --- | --- |
| Overfitting due to small classes | Medium | Apply augmentation and dropout |
| Class imbalance | High | Use weighted loss or resampling |
| Large dataset size | Medium | Train in batches or use generators |
| Browser performance | Low | Optimize dashboard rendering |

**8. Conclusion**

The **Land Type Classification using the EuroSAT Dataset** successfully demonstrates the integration of deep learning with remote sensing data. By combining CNN models and an interactive web dashboard, this project provides a powerful tool for understanding land cover distribution and supporting environmental analysis.