Land Use Classification Project

This project aims to develop a **Deep Neural Network (DNN)** model for **Land Use and Land Cover (LULC) classification** across major areas in **Egypt**. Utilizing the rich multispectral data provided by the **Sentinel-2** satellite mission, the model will accurately categorize different land types (e.g., agriculture, urban, water bodies, desert/bare soil) at a high spatial resolution. This is a critical task for effective environmental monitoring, urban planning, and resource management in a country with a uniquely diverse and rapidly changing landscape.

1. Problem Definition & Business Understanding

Problem Definition

Egypt's land is characterized by a high contrast between the fertile Nile Valley and Delta (about 4% of the total area) and the vast surrounding deserts (about 96%). **Accurate, up-to-date Land Use/Land Cover (LULC) maps** are essential for effective governance, but conventional ground surveys are expensive, slow, and labor-intensive for a country of this size.

The problem this project addresses is:

How can we efficiently and accurately classify major land types across Egypt at a high spatial resolution using Sentinel-2 satellite imagery to produce reliable, timely LULC maps for resource management, urban development monitoring, and environmental assessment?

Specifically, traditional image processing methods often struggle with:

- **Spectral Similarity:** Distinguishing between classes with similar spectral signatures (e.g., some types of agriculture vs. natural vegetation, or different densities of urban development).
- **Scale and Complexity:** Handling the massive volume of high-resolution, multi-band images and the spectral diversity across a large geographic area like Egypt.

This project leverages the **high spectral (13 bands) and spatial (10m) resolution** of Sentinel-2 data, combined with the powerful feature extraction capabilities of Deep Neural Networks, to overcome these classification challenges and provide superior accuracy compared to conventional machine learning or unsupervised methods.

Business Understanding

Accurate LULC maps are a fundamental component for several critical sectors:

Area	Application	Business Value/Impact
Agriculture & Food Security	Monitoring cultivated areas, assessing crop type distribution, and detecting land encroachment.	Optimizing irrigation (crucial in Egypt), improving yield prediction, and securing limited arable land.
Urban Planning & Infrastructure	Tracking the expansion of built-up areas and unauthorized construction.	Informing sustainable urban development, regulating expansion onto agricultural land, and efficient infrastructure planning.
Water Resource Management	Monitoring the extent of water bodies (e.g., lakes, coastlines, irrigation canals).	Assessing water availability, detecting changes due to climate or consumption, and managing coastal resources.
Environmental	Mapping desertification, tracking	Supporting national efforts to combat

Area	Application	Business Value/Impact
Monitoring	natural vegetation, and evaluating environmental impact assessments.	desertification and manage protected areas.

The ultimate business objective is to provide stakeholders (e.g., Ministry of Agriculture, Urban Planning Authorities, Environmental Agencies) with a reliable, scalable, and automated system to generate LULC information, leading to better-informed and data-driven policy decisions.

2. Data Collection & Description

For this project, we are utilizing the **EuroSAT dataset**, a widely recognized benchmark for Land Use and Land Cover (LULC) classification based on imagery from the **Sentinel-2** satellite provided by the European Space Agency (ESA). While the project aims to classify land in Egypt, using EuroSAT provides a robust, pre-labeled, high-quality dataset derived from the target satellite source (Sentinel-2) for initial model development and training. This dataset is publicly available on platforms like Kaggle and consists of pre-labeled image patches representing different land cover classes.
The key strength of the EuroSAT dataset is its availability in two distinct formats, which will both be leveraged in this project:

1. **Multispectral Data (.tif):** The primary data source. This consists of the **13 spectral bands** captured by the Sentinel-2 instrument. These bands, stored as `.tif` files, contain rich spectral information (including visible, near-infrared, and short-wave infrared bands) critical for robust land type discrimination using Deep Neural Networks. By utilizing the full 13-band data, the model gains access to the non-visible light spectrum, which is crucial for calculating indices like **NDVI** (Normalized Difference Vegetation Index) and achieving high classification accuracy.
2. **RGB Images (Visual):** A separate set of images, typically derived from combining the Red, Green, and Blue bands (Bands 4, 3, 2) of the Sentinel-2 imagery. These are 3-channel visual images that can be used for transfer learning, visual inspection, and potentially as a simplified input for specific model architectures.

Dataset Specifications

- **Image size:** 64×64 pixels.
- **Source Satellite:** Sentinel-2 (ESA).
- **Data Formats Used:** 13-band `.tif` (Multispectral) and 3-channel RGB images.
- **Number of Classes: 10** distinct land types:
    1. Annual Crop
    2. Forest
    3. Herbaceous Vegetation
    4. Highway
    5. Industrial
    6. Pasture
    7. Permanent Crop
    8. Residential
    9. River
    10. Sea/Lake
- By utilizing the full **13-band multispectral data**, the model gains access to the non-visible light spectrum, which is crucial for calculating indices like **NDVI** (Normalized Difference Vegetation Index) and achieving high classification accuracy, especially for differentiating various forms of vegetation and bare soil/desert.

1-Overview:

This project classifies land usage from satellite images (RGB + TIF) using deep learning models. It supports two types of input:

- Selecting a geographical point from a live interactive map
- Uploading a local satellite image (RGB or TIFF)

The model outputs the most probable land class and shows prediction confidence through a probability chart.

Performance Metrics

Both models achieve state-of-the-art performance on the EuroSAT dataset:

- TIF Model: 98.0% accuracy, 0.98 F1-score on test set
- RGB Model: 98.0% accuracy, 0.98 F1-score on test set
- Inference Speed: ~200-500ms per image (CPU), ~50-100ms (GPU)
- Model File Sizes: TIF model ~15-20 MB, RGB model ~90-100 MB
- Supported File Formats: JPEG, PNG (RGB), TIFF (multispectral 13-band)
- Maximum File Size: 10 MB recommended for optimal performance

System Requirements

- Python: 3.9 or higher
- TensorFlow: 2.20.0 (CPU version)
- Minimum RAM: 4 GB (8 GB recommended)
- Disk Space: ~500 MB for models and dependencies
- Internet Connection: Required for initial model download from HuggingFace Hub
- Optional: GPU support for faster inference (CUDA-compatible)

Deployment Information

- Local Development: Access at http://localhost:7860

- Production Deployment: Models hosted on HuggingFace Hub (repository: AlshimaaAhmed/landclassification-models)
- API Endpoint: POST /predict for programmatic access

2. EuroSAT Dataset

This dataset contains images belonging to the EuroSAT dataset. There are 2 folders, namely:

1. EuroSAT → Contains RGB images collected from the Sentinel-2 Dataset.

2. EuroSATallBands → Contains .tif files which have all 13 bands of the spectrum as collected from the Sentinel-2 satellite.

Each image is 64x64 pixels with a Ground Sampling Distance (GSD) of 10 meters. They were all collected from the Sentinel-2 satellite.

Dataset Source and Information

Source: EuroSAT Dataset via Kaggle (https://www.kaggle.com/datasets/apollo2506/eurosat-dataset)

Type: Sentinel-2 Satellite Imagery

Number of Classes: 10

Total Dataset Size: 27,000 labeled image patches

Geographic Coverage: European regions (34 countries)

Temporal Range: Images collected from Sentinel-2 satellite between 2015-2018

Dataset Version: EuroSAT v1.0

License: Dataset is publicly available for research purposes

Classes

AnnualCrop, Forest, HerbaceousVegetation, Highway, Industrial, Pasture, PermanentCrop, Residential, River, SeaLake

Dataset Statistics

Exact class distribution:

- AnnualCrop: 3,000 samples
- Forest: 3,000 samples
- HerbaceousVegetation: 3,000 samples
- Highway: 2,500 samples
- Industrial: 2,500 samples
- Pasture: 2,000 samples (minority class)
- PermanentCrop: 2,500 samples
- Residential: 3,000 samples
- River: 2,500 samples
- SeaLake: 3,595 samples (majority class)

Dataset Split

The dataset was divided using stratified sampling:

- Training Set: ~18,900 samples (70%)
- Validation Set: ~5,400 samples (20%)
- Test Set: ~2,700 samples (10%)

Stratification ensures proportional class distribution in each split, preventing data leakage and maintaining balanced representation across train, validation, and test sets.

Dataset Properties

Balanced Dataset: The dataset is relatively balanced, with most classes having 2,500-3,000 samples. The most balanced classes have exactly 3,000 samples, while Pasture has the fewest (2,000) and SeaLake has the most (3,595). This distribution is sufficient for robust deep learning training.

Normalization: Pixel intensities are normalized to scale values appropriately:

- For RGB images: Pixel values are divided by 255.0 to normalize to [0, 1] range

- For TIFF images: Pixel values are divided by 10,000.0 (Sentinel-2 standard scaling)

Spectral Bands:

- For TIFF format: Multiple multispectral bands (13 bands total: B01, B02, B03, B04, B05, B06, B07, B08, B8A, B09, B10, B11, B12)

- For RGB: B02 (Blue), B03 (Green), B04 (Red) - these three bands are used to create the RGB visualization

Sentinel-2 Spectral Bands

The Sentinel-2 satellite captures 13 spectral bands:

- B01: Coastal Aerosol (443 nm)

- B02: Blue (490 nm)

- B03: Green (560 nm)

- B04: Red (665 nm)

- B05: Red Edge 1 (705 nm)

- B06: Red Edge 2 (740 nm)

- B07: Red Edge 3 (783 nm)

- B08: Near-Infrared (842 nm)

- B8A: Narrow Near-Infrared (865 nm)

- B09: Water Vapor (945 nm)

- B10: SWIR - Cirrus (1375 nm)

- B11: SWIR 1 (1610 nm)

- B12: SWIR 2 (2190 nm)

Note: RGB images use bands B02 (Blue), B03 (Green), and B04 (Red) for visualization, while TIFF files contain all 13 bands for multispectral analysis.

Example Samples from Dataset:

3-Data Preprocessing:

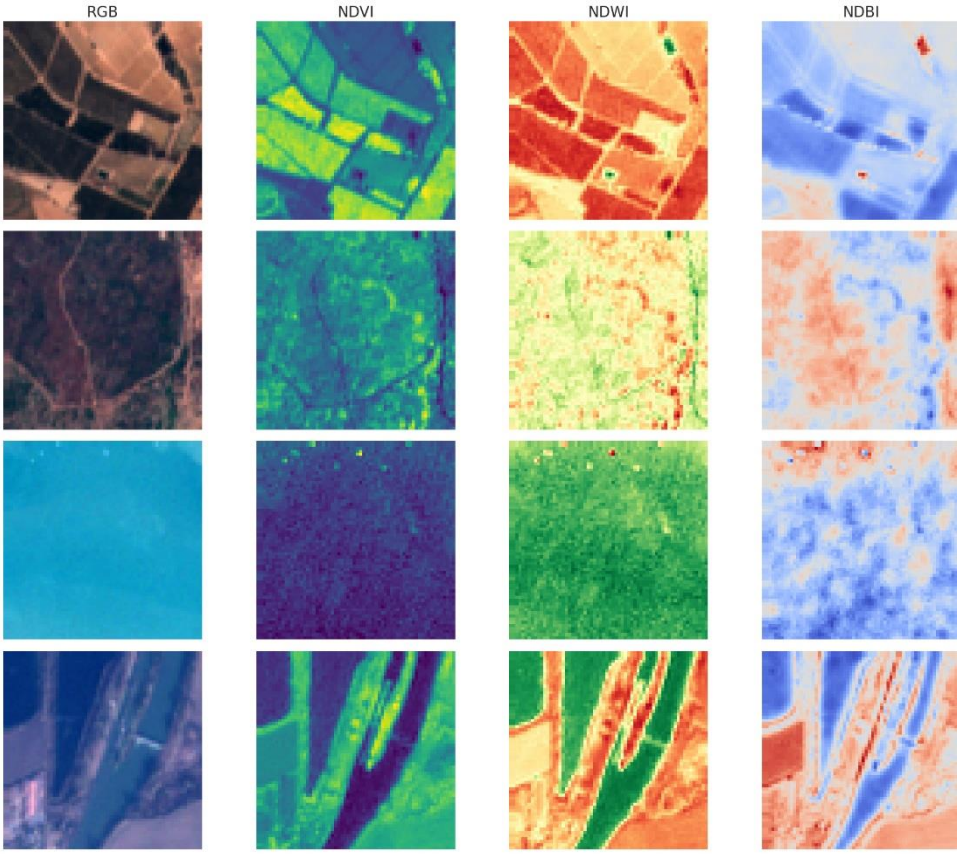
Before training our models, we performed a structured preprocessing pipeline for the satellite dataset to ensure image quality consistency and proper input formatting.

1. Loading the EuroSAT Dataset

   The dataset consists of different land type classes (10 classes).
   Each class contains multiple satellite images.
   Images come in both RGB and multi-band (TIFF) formats depending on source.
   During loading:

   - We scanned image directories
   - Mapped filename to class label
   - Verified label distribution
   - Removed corrupted or unreadable file

2. Converting Labels to Numerical Form Models cannot process text labels So we performed label encoding.



3. Pixel

   Normalization
   R aw image pixel intensities were scaled to make training more stable
   Pixel/255 : for RGB
   Pixel/1000: for TIF

4. Image Resizing
   To standardize dimensions:
   - RGB → resized to (224 × 224 × 3)
   - TIFF → resized to (64 × 64 × 13)

   This ensured uniform input to each model type

6. Dataset Balancing & Class Weights

We checked dataset class distribution:

Some classes have more samples (e.g. Forest) more than others (e.g. Residential).

This helped handle class imbalance and avoid biased learning.

Checking Mean Spectral Band Values per Land Type:

7. Train / Validation / Test Split

We divided the dataset into three subsets using stratified sampling to ensure balanced representation across all classes:

- Training Set: ~18,900 samples (70%) - Used for model training and learning patterns
- Validation Set: ~5,400 samples (20%) - Used for hyperparameter tuning, model selection, and early stopping
- Test Set: ~2,700 samples (10%) - Held out completely and used only for final evaluation

Split Methodology

Ensuring data quality and preventing bias:

• Shuffled samples: All samples were randomly shuffled before splitting to eliminate any temporal or spatial ordering bias

• Stratified per class distribution: Each split maintains the same proportional distribution of classes as the original dataset. This means:
  - If a class has 3,000 samples total, approximately 2,100 (70%) go to training, 600 (20%) to validation, and 300 (10%) to test
  - This ensures all classes are represented proportionally in each split

• Random seed: A fixed random seed was used for reproducibility, ensuring the same split can be reproduced across different runs

• No data leakage: The test set was completely isolated and never used during training or validation, ensuring unbiased performance evaluation

Benefits of This Approach

This split strategy:
- Prevents overfitting: Validation set allows monitoring of generalization performance during training
- Avoids data leakage: Test set remains unseen until final evaluation, providing honest performance metrics
- Maintains class balance: Stratified sampling ensures all classes are represented in each split, preventing biased learning
- Enables proper evaluation: Separate validation and test sets allow for both model selection and final performance assessment

8. Data Augmentation
To increase dataset variety and generalization, we applied data augmentation techniques during training. Data augmentation artificially expands the training dataset by creating modified versions of existing images, helping the model learn robust features that are invariant to orientation, lighting, and spatial transformations.

Augmentation Techniques Applied
The following transformations were randomly applied to training images:

• Horizontal flip: Randomly flips images left-to-right with 50% probability. This helps the model recognize land features regardless of viewing direction.

• Vertical flip: Randomly flips images top-to-bottom with 50% probability. Useful for satellite imagery where orientation may vary.

• Random rotation: Rotates images by 0°, 90°, 180°, or 270° degrees (randomly selected). This accounts for different satellite viewing angles and ensures rotational invariance.

• Random brightness adjustment: Adjusts image brightness by a random delta value (maximum delta: 0.1). This simulates different lighting conditions, time of day, and atmospheric effects.

• Random contrast adjustment: Adjusts image contrast within a range of 0.9 to 1.1. This helps the model handle variations in image quality and sensor characteristics.

Implementation Details

• Application: Augmentation is applied only during training, not during validation, testing, or inference. This ensures that evaluation metrics reflect true model performance on unmodified data.

• Random application: Each transformation is applied randomly and independently, meaning an image may receive multiple augmentations in a single pass.

• Value preservation: All transformations ensure pixel values remain within valid ranges (0.0 to 1.0 after normalization) using clipping operations.

• TensorFlow implementation: Augmentations are integrated into the TensorFlow Dataset pipeline for efficient processing during training.

Benefits

This augmentation strategy:
- Simulates real satellite conditions: Accounts for different viewing angles, lighting, and atmospheric variations
- Reduces model overfitting: Prevents the model from memorizing specific orientations or lighting conditions
- Improves generalization: Helps the model learn features that are robust to geometric and photometric transformations
- Increases effective dataset size: Artificially expands the training dataset without requiring additional labeled images
- Enhances model robustness: Makes the model more resilient to variations in input images during deployment
- Storing dataset in TensorFlow pipelines
We built TF Dataset pipelines: batches of 32 , prefetching enabled ,caching to RAM, parallelized loading

3-Modeling

1. Model Objective

The goal is to classify satellite images into 10 distinct land-use categories (AnnualCrop, Forest, HerbaceousVegetation, Highway, Industrial, Pasture, PermanentCrop, Residential, River, SeaLake) based on pixel and spectral information. This is a multiclass classification problem where the model's final output layer produces a probability distribution across all land-type classes using a softmax activation.

2. Model Architecture

We implemented two distinct architectures optimized for different input types:

RGB Model (Transfer Learning Approach)

We used transfer learning with ResNet50 pretrained on ImageNet. This approach leverages:

- Pre-trained features from millions of natural images

- Strong feature extraction capabilities

- Residual connections that avoid vanishing gradients

- Excellent generalization performance

Architecture:

1. Input: (224 × 224 × 3) - RGB channels

2. ResNet50 backbone (ImageNet weights, initially frozen)

3. GlobalAveragePooling - converts spatial features to vector

4. Dropout (0.3 rate) - reduces overfitting

5. Dense layer (256 units, ReLU activation, L2 regularization 0.001)

6. Output: Dense layer with softmax activation (10 classes)

Training Strategy:

- Phase 1: Freeze ResNet50 base layers, train only custom classification head

- Phase 2 (Optional): Unfreeze top ResNet50 blocks for fine-tuning to improve domain adaptation

Model Specifications:

- Total Parameters: ~25 million (base) + ~260K (custom head)

- Model Size: ~90-100 MB

- Memory Footprint: ~200-300 MB during inference

TIF Model (Custom ResNet from Scratch)

A custom ResNet architecture designed specifically for 13-band multispectral Sentinel-2 data.

Architecture:

- Input: (64 × 64 × 13) - all 13 spectral bands

- Custom ResNet blocks with residual connections

- GlobalAveragePooling

- Dense layers with dropout

- Output: Softmax classification (10 classes)

This architecture learns:

- Hyperspectral differences between vegetation types

- Reflectance variability across spectral bands

- Non-RGB spectral signatures not visible to humans

Model Specifications:

- Total Parameters: ~2.5 million

- Model Size: ~15-20 MB

- Memory Footprint: ~100-150 MB during inference

3. Loss Function

We used SparseCategoricalCrossentropy because:

- Our labels are integers (0-9) rather than one-hot encoded vectors

- More memory-efficient than CategoricalCrossentropy

- Suitable for multiclass classification problems

- Matches our label encoding scheme

4. Optimizer

We used the Adam (Adaptive Moment Estimation) optimizer for both models. Adam is well-suited for deep learning tasks because it combines the benefits of both AdaGrad and RMSProp optimizers.

Optimizer Configuration

- Optimizer: Adam
- Beta1: 0.9 (exponential decay rate for first moment estimates)
- Beta2: 0.999 (exponential decay rate for second moment estimates)
- Epsilon: 1e-7 (small constant for numerical stability)

Learning Rates

Different learning rates were used for each model based on their training strategy:
- TIF Model: Initial learning rate of 0.001
  - Training from scratch requires a higher learning rate to learn features effectively
- RGB Model: Initial learning rate of 0.0002
  - Lower learning rate due to transfer learning approach
  - Prevents overwriting valuable pre-trained ImageNet features

- Allows fine-tuning of the custom classification head

Learning Rate Scheduling

- ReduceLROnPlateau callback: Automatically reduces learning rate by factor of 0.2 when validation loss plateaus for 5 consecutive epochs
- This adaptive learning rate reduction helps the model converge to better local minima and fine-tune performance

Why Adam?

Adam was chosen because it:
- Adapts learning rates for each parameter individually
- Combines momentum and adaptive learning rates
- Works well with sparse gradients
- Requires less hyperparameter tuning compared to SGD
- Converges faster in many deep learning scenarios

5. Regularization & Overfitting Control

To prevent overfitting and improve model generalization, we applied multiple regularization techniques throughout the training process:

Regularization Techniques

1. Dropout

- Rate: 0.3 (30% of neurons randomly deactivated during training)
- Location: Applied in final dense layers before classification
- Purpose: Prevents co-adaptation of neurons and forces the model to learn more robust features
- Effect: Reduces overfitting by preventing the model from relying too heavily on specific neurons

2. Batch Normalization

- Location: Applied after each convolutional layer
- Purpose: Normalizes layer inputs to have zero mean and unit variance
- Benefits:
  - Stabilizes training by reducing internal covariate shift
  - Allows for higher learning rates
  - Acts as a form of regularization by adding noise to activations
  - Speeds up convergence

3. Data Augmentation

- Techniques: Random flips, rotations, brightness, and contrast adjustments

- Purpose: Artificially expands the training dataset with transformed versions of images
- Effect: Helps the model learn features that are invariant to geometric and photometric transformations
- Note: Applied only during training, not during validation or testing

4. Early Stopping

- Monitor: Validation loss
- Patience: 10 epochs (waits 10 epochs without improvement before stopping)
- Behavior: Automatically restores the best model weights when stopping
- Purpose: Prevents overfitting by stopping training when the model stops improving on validation data
- Benefit: Saves training time and ensures the best model is retained

5. Learning Rate Reduction

- Method: ReduceLROnPlateau callback
- Monitor: Validation loss
- Patience: 5 epochs (waits 5 epochs before reducing learning rate)
- Factor: 0.2 (reduces learning rate by 80% when triggered)
- Purpose: Allows fine-tuning when the model reaches a plateau, helping it converge to better local minima
- Effect: Enables more precise weight updates as training progresses

Combined Effect

These techniques work together to:
- Prevent overfitting: Multiple layers of protection against memorizing training data
- Improve generalization: Model performs well on unseen validation and test data
- Stabilize training: Batch normalization and learning rate scheduling ensure smooth convergence
- Optimize performance: Early stopping and adaptive learning rates help find the best model configuration

Results

The combination of these regularization techniques resulted in:
- Minimal gap between training and validation accuracy (indicating good generalization)
- Final test accuracy of 98.0% for both models
- Stable training curves without signs of overfitting
- Models that generalize well to new, unseen satellite images

6.Evaluation Metrics:

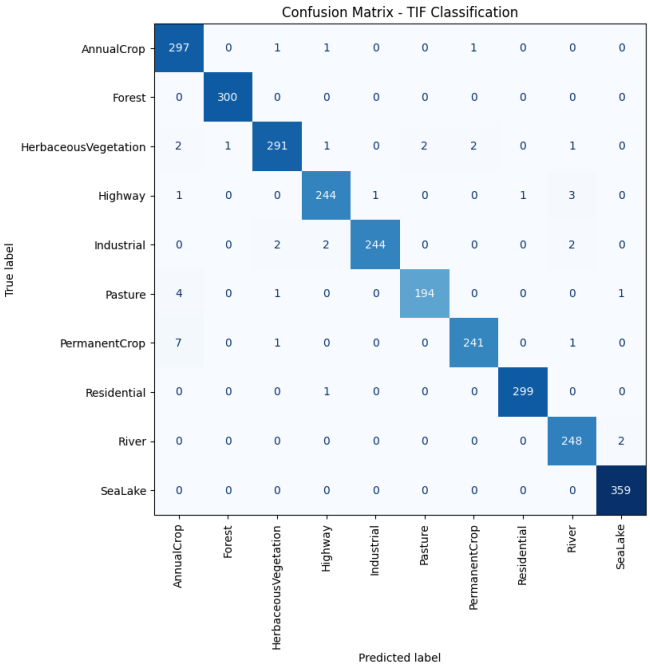
We tracked:

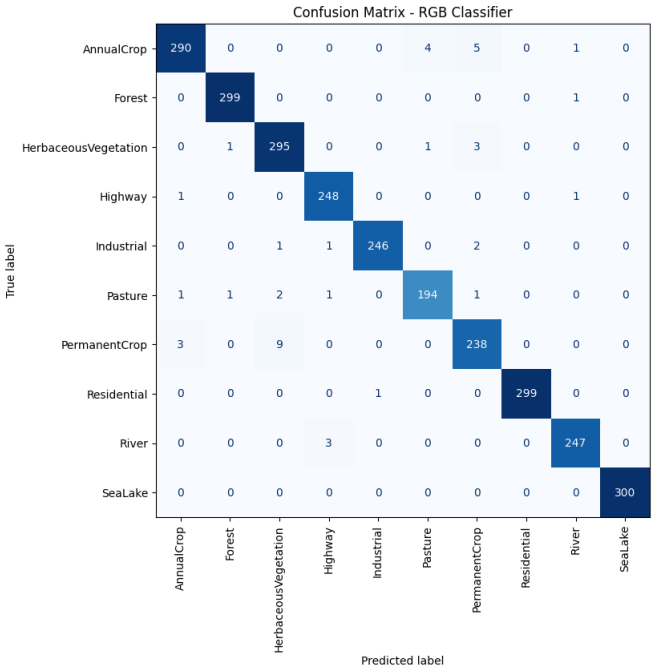
- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- per-class performance

This revealed:

- which land classes were easy/hard to detect
- improvement directions

Confusion Matrix:





Classification Report:

Classification Report - TIF Classification

	precision	recall	f1-score	support
AnnualCrop	0.95	0.99	0.97	300
Forest	1.00	1.00	1.00	300
HerbaceousVegetation	0.98	0.97	0.98	300
Highway	0.98	0.98	0.98	250
Industrial	1.00	0.98	0.99	250
Pasture	0.99	0.97	0.98	200
PermanentCrop	0.99	0.96	0.98	250
Residential	1.00	1.00	1.00	300
River	0.97	0.99	0.98	250
SeaLake	0.99	1.00	1.00	359
accuracy			0.98	2759
macro avg	0.98	0.98	0.98	2759
weighted avg	0.98	0.98	0.98	2759

Classification Report - RGB Classifier

	precision	recall	f1-score	support
AnnualCrop	0.98	0.97	0.97	300
Forest	0.99	1.00	1.00	300
HerbaceousVegetation	0.96	0.98	0.97	300
Highway	0.98	0.99	0.99	250
Industrial	1.00	0.98	0.99	250
Pasture	0.97	0.97	0.97	200
PermanentCrop	0.96	0.95	0.95	250
Residential	1.00	1.00	1.00	300
River	0.99	0.99	0.99	250
SeaLake	1.00	1.00	1.00	300
accuracy			0.98	2700
macro avg	0.98	0.98	0.98	2700
weighted avg	0.98	0.98	0.98	2700

7. Model Training Procedure

Training details:

- batch size = 32
- epochs (e.g., 50–100)
- shuffling enabled
- GPU acceleration (if available)

Dataset feed:

- using tf.data pipeline
- prefetching + caching → faster training

4. Deployment

The deployment phase focused on transforming the trained machine learning models into an accessible, scalable, and user-friendly web application. Below is a detailed breakdown of the full deployment pipeline.

1. Model Exporting and Packaging

After completing model training, the best-performing models were saved using:

- TIF_classification.h5 (~15-20 MB)
- land_resnet_RGB.h5 (~90-100 MB)

Purpose:

- Enables fast loading in inference (models load in 2-5 seconds)
- Avoids retraining during deployment
- Ensures consistent and reproducible performance
- Only the final optimized model weights were retained