<u>AI-Driven Land Use and Land Cover Mapping using Supervised Machine Learning Algorithms</u>

Project Report Submitted in fulfillment of the

Requirements for the Award of the Internship of

Winter Training Program on Space Science Technology

Subject Name: Remote Sensing and GIS

By

R Sanjeev Prasad

Vellore Institute of Technology, Chennai

Institute Roll No: 25BCE1030

Enrollment No: 324720

Under the Supervision of

Ms Alisha Sinha

India Space Academy

Department of Space Education

Acknowledgement

I would like to express my sincere gratitude to my faculty mentor, **Ms. Alisha Sinha**, for her valuable guidance, continuous support, and constructive feedback throughout the completion of this project on Remote Sensing and GIS. Her encouragement and technical insights greatly helped me understand the practical aspects of Land Use Land Cover (LULC) classification using QGIS and Google Earth Engine.

I would also like to thank **India Space Academy, Department of Space Education,** for providing the necessary academic environment and resources to successfully carry out this project. The tools and knowledge gained during the course significantly enhanced my understanding of geospatial analysis and satellite image processing.

Finally, I would like to acknowledge the use of Sentinel-2 satellite data and the platforms QGIS and Google Earth Engine, which made this study possible.

TABLE OF CONTENTS

1. Title

AI-Driven Land Use and Land Cover Mapping using Supervised Machine Learning Algorithms

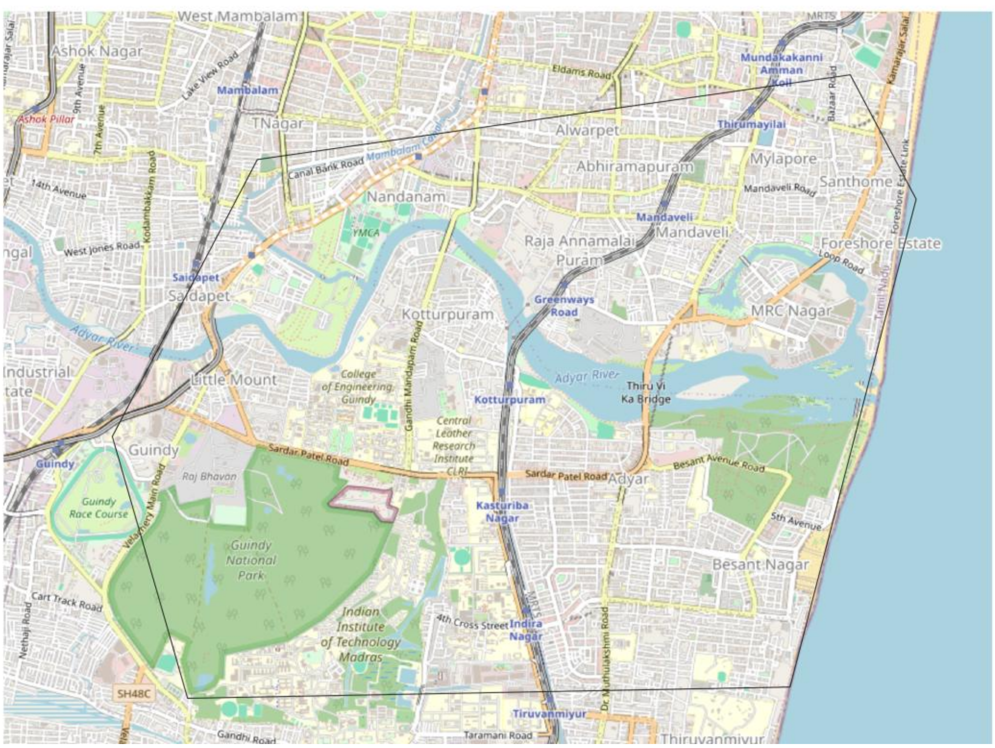
2. Objective

The objective of this project is to generate an accurate Land Use and Land Cover (LULC) map using supervised Machine Learning algorithms applied to multispectral satellite imagery. The study integrates Artificial Intelligence techniques with remote sensing data to improve classification accuracy and spatial interpretation.

3. Study Area

The study area represents the selected Area of Interest (AOI) in Chennai, Tamil Nadu, India defined using administrative boundary shapefiles. The AOI map was prepared in QGIS using the original Raster Data file. Later was imported in GEE marking Reusability.

Location of Chennai AOI



4. Data Used

Satellite Imagery:

- Sentinel-2 MSI (10m spatial resolution)

Data Source:

- Copernicus Browser: https://browser.dataspace.copernicus.eu/

Bands Used:

- Band 2 (Blue),
- Band 3 (Green),
- Band 4 (Red),
- Band 8 (NIR)

Platform:

- Google Earth Engine
- QGIS

Ancillary Data:

- Area of Interest (AOI) shapefile (Created in QGIS)
- Google Earth imagery for visual validation

Data Range:

The input Sentinel-2 Surface Reflectance image showed the following pixel value ranges within the study area:

- Band 2 (Blue):1112-7614
- Band 3 (Green):1177-7546
- Band 4 (Red):1091.5-7982
- Band 8 (NIR)1015.5-7968

These values represent scaled surface reflectance (0-10000), which are dimensionless reflectance measurements.

5. Methodology

5.1 QGIS-Based Workflow – Supervised LULC Classification

The Land Use and Land Cover (LULC) classification was performed in QGIS using a supervised Machine Learning approach. The workflow included preprocessing, training sample preparation, classification, spatial analysis, and map layout generation.

Step 1: Satellite Data Acquisition

Cloud-free Sentinel-2 multispectral imagery was downloaded for the selected study area. Sentinel-2 MSI data with 10 m spatial resolution was selected due to its high spatial detail and availability.

The following spectral bands were used:

- ✓ Band 2 (Blue)
- ✓ Band 3 (Green)
- ✓ Band 4 (Red)
- ✓ Band 8 (Near Infrared – NIR)

These bands provide sufficient spectral information for distinguishing major land cover classes.

Step 2: Importing and Clipping to AOI

- The spectral bands were loaded into QGIS as raster layers.
- To restrict analysis only to the defined study area, each band was clipped using:
- Raster → Extraction → Clip Raster by Mask Layer
- The AOI shapefile was used as the mask layer. This ensured all further processing was limited to the selected boundary.

Step 3: Creation of Multiband Raster

- The clipped spectral bands were stacked into a single multiband raster using:
- Raster → Miscellaneous → Build Virtual Raster (VRT)
- Band stacking allows the classifier to analyze all spectral information simultaneously, improving classification performance.

Step 4: Identification of Land Cover Classes

Based on visual interpretation and reference imagery, the following LULC classes were defined:

- ➢ Water Bodies
- ➢ Vegetation
- ➢ Built-up Areas
- ➢ Barren Land

These classes represent dominant surface features within the study area.

Step 5: Collection of Training Samples

Training samples were created using polygon digitization in QGIS.

For each class:

- ✓ Multiple representative polygons were drawn (10 for each Class).
- ✓ Homogeneous areas were selected.
- ✓ Edge pixels and mixed areas were avoided.
- ✓ Samples were distributed across the study area.
- ✓ Each polygon was assigned a class ID for supervised learning.
- ✓ Saved as Shape File for Further Usage (In GEE).

Step 6: Supervised Classification using Random Forest

A Random Forest Machine Learning classifier was applied using the prepared training samples.

The classifier:

- ✓ Learned spectral signatures of each class.
- ✓ Classified the multiband raster into LULC categories.
- ✓ Generated a classified raster output.
- ✓ Random Forest was selected due to its robustness and higher classification reliability compared to traditional statistical classifiers.

Step 7: Raster to Vector Conversion

- ➢ The classified raster was converted into vector polygons using:
- ➢ Raster → Conversion → Polygonise (Raster to Vector)
- ➢ This step enabled further spatial analysis and area computation for each class.

Step 8: Area Statistics Calculation

- ➢ Using the Field Calculator in QGIS, area for each land cover class was calculated using: area($geometry) / 1,000,000
- ➢ This provided class-wise area in square kilometers.
- ➢ The total area of each LULC category was computed and tabulated for interpretation.

Step 9: Map Layout Preparation

A professional cartographic layout was created using the QGIS Layout Manager.

The layout included:

- ✓ Map Title
- ✓ Legend
- ✓ Scale Bar
- ✓ North Arrow
- ✓ Coordinate Reference System
- ✓ Data Source Information

The final thematic LULC map was exported for documentation and reporting

5.2 Google Earth Engine (GEE) Workflow – Code Explanation

The Land Use and Land Cover (LULC) classification was performed in Google Earth Engine (GEE) using supervised Machine Learning. The workflow included importing the study area, filtering satellite data, preparing training samples, training a Random Forest classifier, generating the classified map, and exporting the result.

Each step is explained below along with the corresponding code logic.

Step 1: Importing the Area of Interest (AOI)

```
var AOI = ee.FeatureCollection("projects/concise-slate-470913-j3/assets/AOI");
Map.centerObject(AOI, 10);
Map.addLayer(AOI, {}, 'AOI');
```

Explanation:

- ee.FeatureCollection() loads the uploaded AOI shapefile from GEE Assets.
- Map.centerObject(AOI, 10) centers the map view over the study area at zoom level 10.
- Map.addLayer() displays the AOI boundary on the map.

This ensures all processing is restricted to the selected study region.

Step 2: Loading Sentinel-2 Image Collection

```
var s2 = ee.ImageCollection("COPERNICUS/S2_SR")
  .filterBounds(AOI)
  .filterDate('2023-01-01', '2023-12-31')
  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
  .select(['B2','B3','B4','B8']);
```

Explanation:

- "COPERNICUS/S2_SR" loads Sentinel-2 Surface Reflectance data.
- filterBounds(AOI) restricts imagery to the study area.
- filterDate() selects imagery within the chosen time period.
- filter(ee.Filter.lt(...)) removes images with more than 10% cloud cover.

- select(['B2','B3','B4','B8']) selects only required spectral bands.

This creates a clean image collection ready for analysis.

Step 3: Creating Median Composite and Clipping

```
var image = s2.median().clip(AOI);
Map.addLayer(image, {bands: ['B4','B3','B2'], min:0, max:3000}, 'RGB');
```

Explanation:

- .median() creates a median composite to reduce cloud and noise.
- .clip(AOI) ensures the image covers only the study boundary.
- The RGB visualization uses bands B4, B3, B2 (True Color).

This generates a cloud-minimized base image for classification.

Step 4: Creating False Color Composite (Visual Interpretation)

```
Map.addLayer(image, {bands: ['B8','B4','B3'], min:0, max:3000}, 'False Color');
```

Explanation:

- B8 (NIR), B4 (Red), B3 (Green) are used.
- Vegetation appears bright red due to strong NIR reflectance.
- This helps visually verify land cover separability before classification.

Step 5: Loading Training Data

```
var training = ee.FeatureCollection(
  "projects/concise-slate-470913-j3/assets/TRAINING_LAYERS"
);
```

Explanation:

- Loads training polygons created in QGIS.
- Each polygon contains a class_id attribute.
- These samples represent known land cover classes.

Step 6: Sampling Image Using Training Data

```
var bands = ['B2','B3','B4','B8'];

var trainingSample = image.select(bands).sampleRegions({
  collection: training,
  properties: ['class_id'],
  scale: 10
});
```

Explanation:

- select(bands) selects input features.
- sampleRegions() extracts spectral values at training polygon locations.
- properties: ['class_id'] links each sample to its class label.
- scale: 10 ensures 10m resolution sampling.

This creates a structured dataset for Machine Learning.

Step 7: Training the Random Forest Classifier

```
var classifier = ee.Classifier.smileRandomForest(100)
 .train({
  features: trainingSample,
  classProperty: 'class_id',
  inputProperties: bands
 });
```

Explanation:

- smileRandomForest(100) creates a Random Forest model with 100 trees.
- .train() trains the model using:
    - Spectral bands as input features
    - class_id as output label

Random Forest builds multiple decision trees and combines them for robust classification.

Step 8: Applying Classification

var classified = image.select(bands).classify(classifier);

Explanation:

- Applies the trained classifier to the full image.
- Each pixel is assigned a land cover class.
- Output is a classified raster.

Step 9: Visualizing Classified Output

```
Map.addLayer(classified, {
  min: 1,
  max: 4,
  palette: ['blue','green','grey','yellow']
}, 'LULC Classification');
```

Explanation:

- Assigns colors to class IDs:
    - Blue – Water
    - Green – Vegetation
    - Grey – Built-up
    - Yellow – Barren Land
- Displays final LULC map.

Step 10: Exporting the Classified Image

```
Export.image.toDrive({
  image: classified,
  description: 'LULC_Classified_AOI',
  region: AOI,
  scale: 10,
  maxPixels: 1e13
});
```

Explanation:

- Exports classified raster to Google Drive.
- scale: 10 maintains Sentinel-2 resolution.
- region: AOI restricts export to study boundary.
- maxPixels prevents pixel limit errors.

The exported GeoTIFF is used for reporting and further GIS analysis.

Why This is AI-Driven

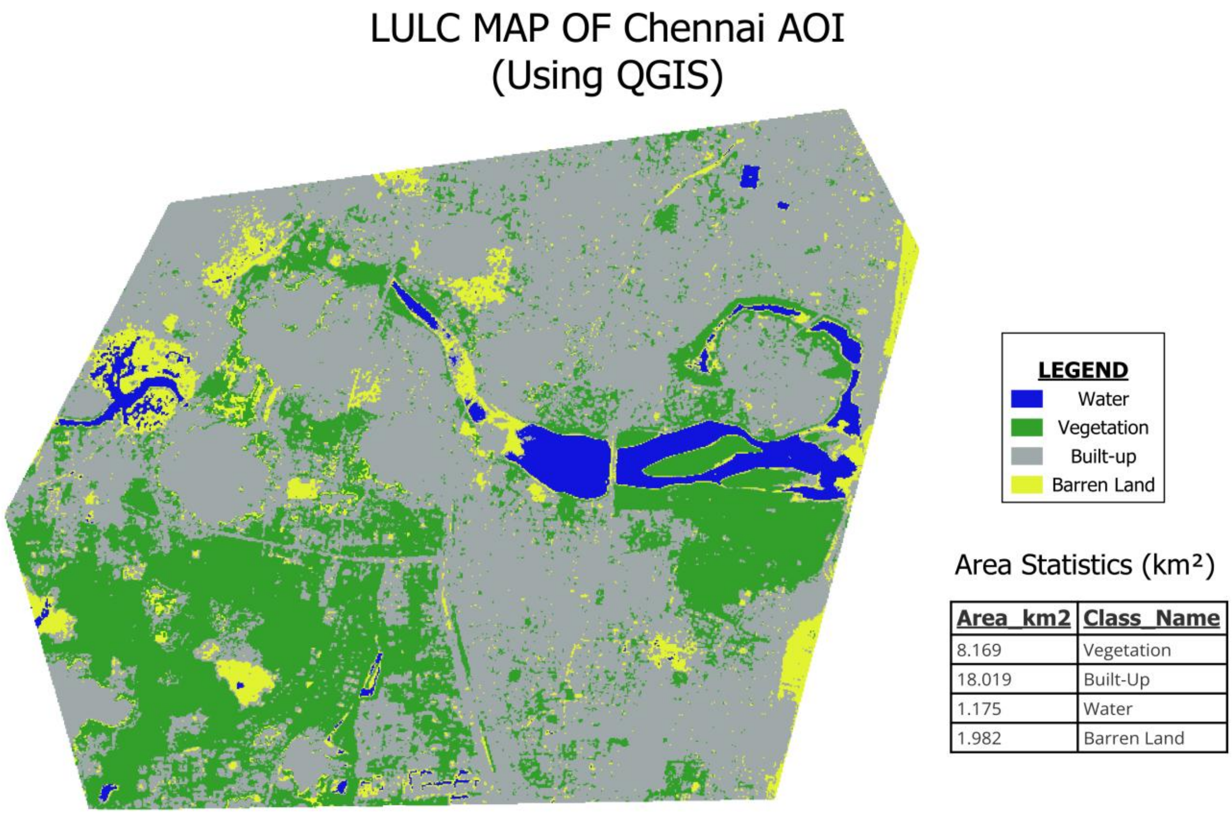
The classification uses a supervised Machine Learning algorithm (Random Forest), which:

- Learns spectral patterns from training data.
- Automatically separates land cover classes.
- Reduces manual threshold dependency.
- Improves classification accuracy compared to rule-based methods.

This demonstrates practical implementation of AI within a geospatial workflow.

6. Results

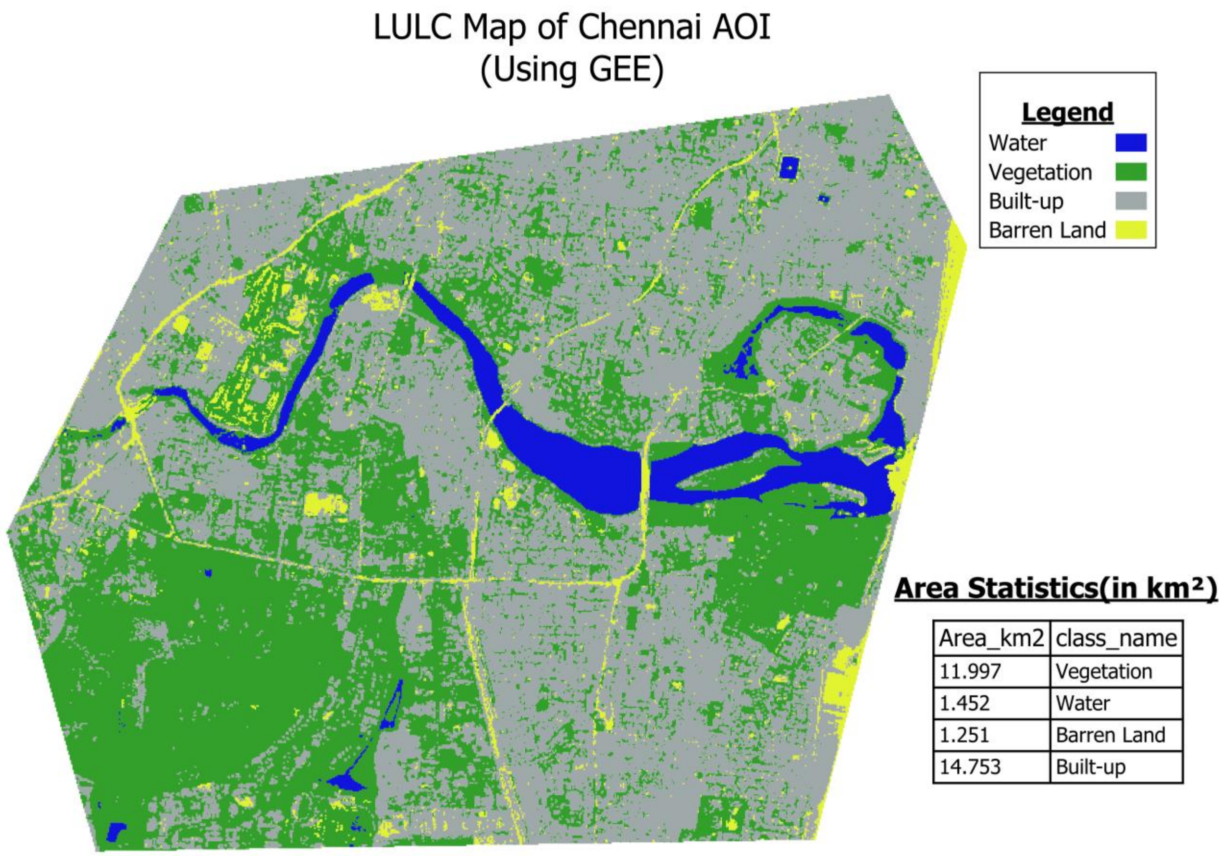
6.1. QGIS Layout Result



Area_km2	Class_Name
8.169	Vegetation
18.019	Built-Up
1.175	Water
1.982	Barren Land

- The LULC map was generated using supervised classification in QGIS based on Sentinel-2 imagery at 10 m spatial resolution.
- Four major land cover classes were identified: Water, Vegetation, Built-up, and Barren Land.
- Area statistics were calculated by converting the classified raster into vector polygons and computing area in km^2 using projected CRS

6.2. GEE Export Layout Result



- The LULC map was produced in Google Earth Engine using Sentinel-2 Surface Reflectance data and supervised classification.
- The classified image was exported at 10 m spatial resolution and further styled in QGIS for layout preparation.
- Minor variations in area statistics compared to QGIS results are due to differences in processing environment and classification workflow.

7. Conclusion

The Land Use Land Cover (LULC) classification of the Chennai AOI was successfully carried out using both QGIS and Google Earth Engine (GEE). The supervised classification approach proved effective in distinguishing major land cover classes such as Built-up area, Vegetation, Water bodies, and Barren land. The use of Sentinel-2 imagery with appropriate band combinations improved the separability of different surface features. Google Earth Engine enabled faster processing and efficient cloud-based computation, while QGIS provided better control over visualization, layout design, and spatial analysis.

However, some limitations were observed during the study. The classification accuracy depends heavily on the quality and distribution of training samples. Misclassification occurred in areas where spectral signatures of classes were similar (e.g., barren land and built-up areas). Moderate spatial resolution (10 m) of Sentinel-2 data may not capture very small features accurately. Seasonal variations and atmospheric conditions may also influence reflectance values and classification output.

Possible improvements include increasing the number and quality of training samples, applying advanced classifiers such a Support Vector Machine with parameter tuning, and incorporating additional indices like NDVI or NDBI to enhance class separability. Accuracy assessment using a confusion matrix could further validate the reliability of the results.

From the selected Chennai AOI, it was observed that built-up areas dominate the region, indicating rapid urban expansion. Significant vegetation patches are present in certain zones, while major water bodies form distinct and clearly classified regions. The spatial distribution reflects the urban growth pattern and highlights the importance of sustainable land management planning.

8. References

- Copernicus Open Access Hub – Sentinel-2 Data
- Google Earth Engine Data Catalog
- QGIS Documentation