**Remote Sensing Analysis of Madagascar Using PlanetScope and Sentinel-1 Imagery**

**DAT-103**

****

**Golla Sahithi , DSAI**

**Abstract**

This study examines feature extraction and comparison of PlanetScope and Sentinel-1 satellite imagery over a 5 km × 5 km Area of Interest (AOI) using classical image analysis and deep learning models. A cloud-free PlanetScope image from 2024 was obtained, and initial statistical analysis was conducted with boxplots for each spectral band. The Digital Number (DN) values were converted to reflectance, and box plots were generated for visualization. Concurrently, Sentinel-1 SAR intensity imagery was acquired and analyzed using ESA SNAP software, producing decibel (dB) scale boxplots.

A comparative analysis revealed distinct radiometric characteristics in both datasets. Canny edge detection highlighted various features, with PlanetScope showing roads and urban boundaries, while Sentinel-1 focused on structural and moisture-related edges. EfficientNet models were employed for deep feature extraction, with dimensionality reduction techniques (e.g., PCA, t-SNE) utilized to visualize meaningful structures. Hyperparameter tuning, data augmentation, and learning rate optimization improved classification accuracy. Performance was assessed using metrics like accuracy, precision, recall, and F1-score, showcasing the robustness across sensor types. This framework illustrates the complementary strengths of optical and SAR imagery, integrating classical techniques with deep learning for enhanced satellite image interpretation.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### **Table of Contents**

* Introduction
* Experimental Setups
* Satellite imagery acquisition and processing
* Graphical Analysis of PlanetScope
* Graphical Analysis of Sentinel-1 dataset
* Canny Edge Detection
* Feature detection and comparison of sentinel and planetscope
* Introduction to EfficientNet model
* Hyperparameter tuning and model optimization

### 

### **1. Introduction**

Satellite remote sensing has revolutionized how we observe and analyze Earth's surface. Using multi-spectral optical and radar-based satellite data, we can gain insights into land cover, vegetation, urban structures, and more. In this study, we focus on a 5 km x 5 km area in Madagascar and perform a complete satellite image analysis using two datasets: PlanetScope (optical) and Sentinel-1 (radar).

The process begins with selecting a suitable Area of Interest (AOI) in Madagascar. We then download a cloud-free PlanetScope image for the year 2024 using Planet’s Docker-based platform. The raw image contains Digital Numbers (DN), which we visualize using boxplots. These DNs are converted to reflectance values using a Planet-provided Jupyter Notebook. New boxplots help us understand the reflectance distribution.

Separately, we acquire Sentinel-1 radar imagery from ESA Hub Data. The radar image, which is in decibel (dB) format, is analyzed in SNAP software to understand intensity variations. Following this, we apply Canny edge detection to both datasets to extract high-frequency features.

For deeper analysis, we use the EfficientNet family of models to identify and classify features in both images. To improve the accuracy, we employ techniques such as hyperparameter tuning and data augmentation. Finally, we compare results and present our findings using performance metrics.

This report documents each step in detail and provides visual and quantitative analysis throughout.

### **2. Experimental setup**

1. PlanetScope Image Acquisition

Platform: Planet.com  
Tool: Planet Downloader (Docker-based)  
Notebook: quick\_start\_download (official Planet notebook)  
Objective: Download a 2024 PlanetScope scene covering the madagascar AOI, ensuring less than 15% cloud cover.

2. Data Preprocessing and Conversion

Task: Conversion of Digital Numbers (DN) to surface reflectance  
Tool: Dockerized processing utilities provided by Planet  
Purpose: Standardize imagery for analysis by converting raw DN values into physically meaningful reflectance values.

3. Statistical Analysis and Visualization

Libraries Used:  
 matplotlib– for visual plots and data presentation  
 rasterio– for reading and handling raster data

Application:  
Generated boxplots to compare DN and reflectance values across all bands for exploratory data analysis.

4. Feature Extraction via Edge Detection

Library Used: openCV  
Method: Canny Edge Detection  
Data Sources: Applied to both PlanetScope and Sentinel-1 images to highlight sharp features and boundaries.

5. Radar Data Preprocessing

Software: SNAP (Sentinel Application Platform)  
Use Case: Preprocessing Sentinel-1 SAR imagery (e.g., calibration, speckle filtering, terrain correction).

6. Supporting Environment and Dependencies

Containerization: Docker was employed to create an isolated and reproducible environment for Planet data processing.

Python Packages: rasterio, openCV, matplotlib**3. Satellite Imagery Acquisition and Processing**

### **3.1 Acquisition and Processing of PlanetScope Optical Imagery**

High-resolution optical imagery over the Madagascar area of interest (AOI) was obtained using PlanetScope satellite data, accessed through Planet.com’s API and Docker-integrated tools.

1. Image Acquisition Date: 17 January 2024
2. Product Specification: ortho\_analytics\_4b (4 spectral bands: Blue, Green, Red, and Near-Infrared)
3. Cloud Cover: Estimated at 5–10%, within acceptable limits for optical analysis
4. Download Methodology: Utilized the quick\_start\_download Jupyter notebook (official Planet.com release), executed within a Docker environment

### **3.2 Acquisition and Preprocessing of Sentinel-1 SAR Data**

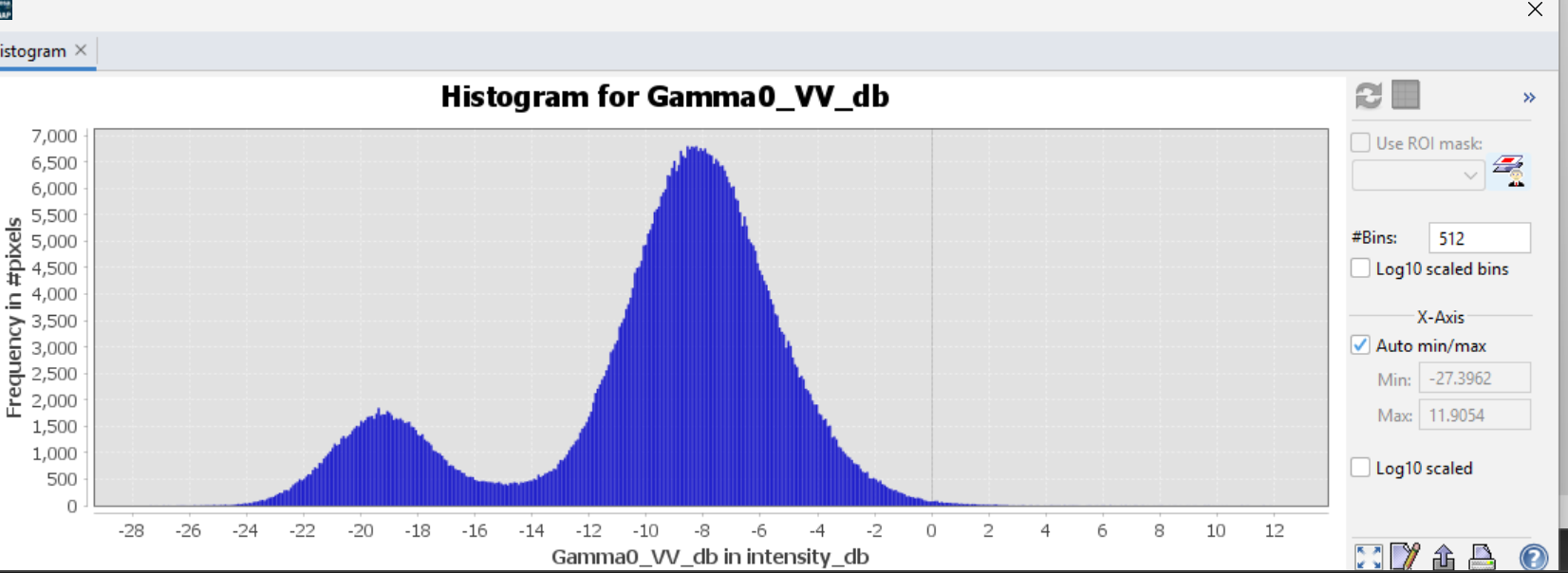
To complement optical data with cloud- and light-independent radar imagery, Sentinel-1 SAR (Synthetic Aperture Radar) data was downloaded from the ESA-Hub Data Portal.

1. Image Acquisition Date: 17 January 2024
2. Product Type: Interferometric Wide (IW) mode – Ground Range Detected (GRD)
3. Temporal Proximity: Although acquired two days earlier than the PlanetScope image, the data was close enough for temporal comparison

#### **SAR Image Preprocessing Workflow (Conducted in SNAP)**

Preprocessing was performed using SNAP (Sentinel Application Platform), following a structured approach for optimal radar image calibration and georeferencing:

1. Band Selection:  
    Extracted the VV polarization (vertical transmit & receive) band for analysis.
2. Precise Orbit File Application:  
    Applied updated orbit files to refine geolocation precision.
3. Thermal Noise Removal:  
    Eliminated system-generated noise to improve radar signal quality.
4. Radiometric Calibration:  
    Calibrated the image to gamma0 , ideal for hilly terrain conditions typical of Madagascar.
5. Speckle Noise Filtering:  
    Applied the Refined Lee Filter to suppress speckle noise while retaining edge details.
6. Geometric Terrain Correction:  
    Performed using Range Doppler Terrain Correction with the following parameters:  
    DEM Source: Copernicus 30-meter Global DEM  
    Map Projection: UTM, WGS 84 datum
7. Conversion to Decibel (dB) Scale:  
    Transformed backscatter values from linear scale to dB units to enhance interpretability and statistical analysis.
8. Histogram Evaluation:  
    Create a subset until you get a bimodal histogram signifying distinction between land and water.

Finally we get the above histogram:

The two peaks are extremely well separated: One peak around -19 dB (smaller → likely water), One peak around -8 dB (larger → likely land, vegetation, or urban areas). Two distinct classes → Different surfaces

**The first peak (~ -19 dB):**

* Very low backscatter.
* Likely smooth surfaces → e.g., calm water, flat bare soil, or very sparse vegetation.

**The second peak (~ -8 dB):**

* Higher backscatter.
* Associated with rougher surfaces, such as vegetated areas, urban areas, or complex terrain (like hilly areas or forests).
* It's not extremely sharp (it's a bit broad), which hints that the second class (rougher terrain) is not perfectly uniform. In flat agricultural land, you'd usually get a very narrow peak. In hilly or forested terrain, surface slope, vegetation height, and orientation relative to radar vary a lot → you get broader backscatter distribution. hence our histogram goes with our filter selection and proves that the terrain is hilly

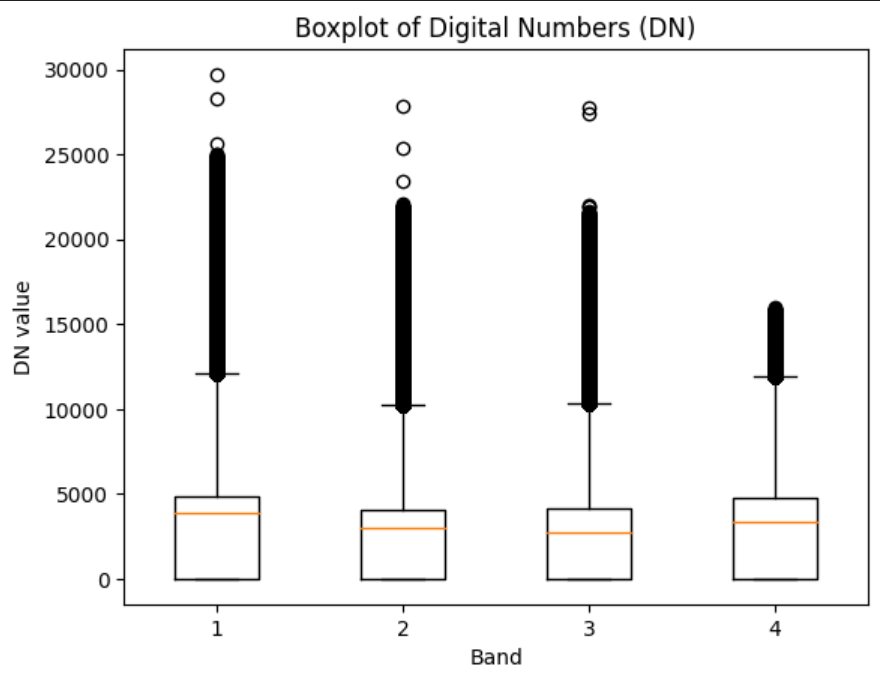
1. Spatial Subsetting:  
    Used the Raster → Subset tool to crop the image spatially to the defined AOI (5 km × 5 km extent).
2. Final Export:  
    Saved the processed radar image in GeoTIFF (.tif) format, ready for integration with optical data and further analysis.

**4. Graphical Analysis of the planetscope image:**

PlanetScope imagery consists of four multispectral bands:

1. Blue (455–515 nm) – Useful for water body analysis and atmospheric correction.
2. Green (500–590 nm) – Helps in vegetation monitoring and assessing plant health.
3. Red (600–670 nm) – Important for vegetation classification and soil analysis.
4. Near-Infrared (NIR) (780–860 nm) – Crucial for detecting vegetation vigor and biomass estimation.

**4.1. Boxplot of DN Numbers:**

****

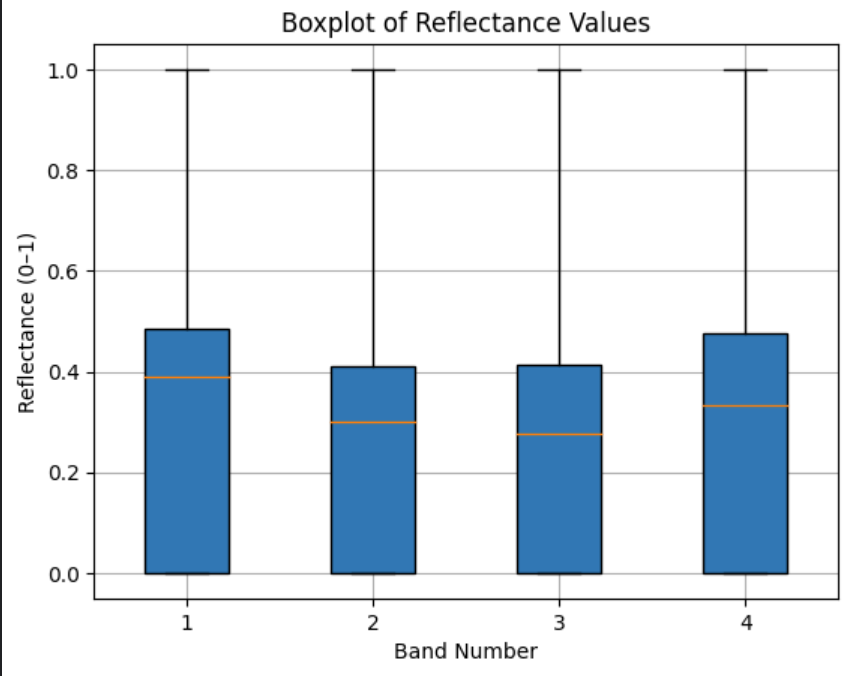
*Figure 4.1.0 Boxplot of Digital Numbers for planetscope image*

The generic term for **pixel values** is Digital Number or DN. It is commonly used to describe pixel values that have not yet been calibrated into physically meaningful units. The boxplot visualizes the Digital Numbers of the planetscope image across four spectral bands. It helps analyse:

* Central tendency
* Variability (Spread within each box)
* Outliers (Spread beyond whiskers)

**Key components of a box plot:**

* Median (median DN values shown as the orange colored lines): The orange value depicts the median of the data; in our case, the median is skewed towards the top, indicating a **non-uniform distribution.**
* Interquartile Range (IQR - the box): The length of the box represents the spread of the central 50% of the data. Here, two boxes, **1 and 4**, are taller, suggesting **higher variability**, and the other two, **2 and 3**, are shorter, suggesting **lower variability**.
* Whiskers (lines lying exceeding the box): These cover data within 1.5 times the IQR. Longer whiskers mean **greater range;** shorter whiskers indicate a tightly clustered dataset. In our dataset, the whiskers are longer, indicating a greater range.
* Outliers: dots outside the whiskers representing the extreme values are outliers. We see a lot of outliers in our dataset here, which suggests too many **anomalies and irregularities**.

**4.2 Boxplot Values of reflectance values:**

*Figure 4.2.0 Boxplot of Reflectance Values*

**Reflectance Values:**

Reflectance values describe how much light is reflected by a surface compared to the total incoming light. They are typically **normalized between 0 and 1**, where:

* **0** means no reflectance (all light is absorbed).
* **1** means total reflectance (all light is reflected).

**Why Reflectance Values Matter?**

These values help analyze:

* **Surface properties**: Different materials (water, vegetation, soil, urban structures) reflect light differently.
* **Spectral characteristics**: Each band in satellite imagery (Blue, Green, Red, and Near-Infrared) captures unique reflectance patterns.
* **Vegetation health**: Healthy plants reflect more in the **NIR band** and absorb **Red light**.
* **Water identification**: Water bodies strongly absorb **NIR and Red light** but reflect **Blue and Green**.

**Key Components of plot:**

* Median:The central orange line represents the median of each band, here we can see that median reflectance is relatively smaller, hovering around 0.3 and 0.4.
* Interquartile range indicates the spread of 50% of data, band1 and band4 have comparatively larger interquartile range that band2 and band3, suggesting more variability in their central reflectance.
* Whiskers capture the range of each band, excluding outliers and all the four bands exhibit a wide range.
* Outliers: The outliers are absent in this distribution, as we dont see any points explicitly beyond the whiskers.
* Symmetry of the distribution: The median of a distribution gives a fair idea on the symmetry of the data and here we can see that all the bands have skewness present, yet in most of the boxplots you notice that the median is somewhat in the centre, suggesting a relatively symmetric distribution of the central data.

**Statistics in Reflectance values:**

Reflectance Stats:

Band 1: min=0.000000, max=1.000000, mean=0.416514

Band 2: min=0.000000, max=1.000000, mean=0.383924

Band 3: min=0.000000, max=1.000000, mean=0.307059

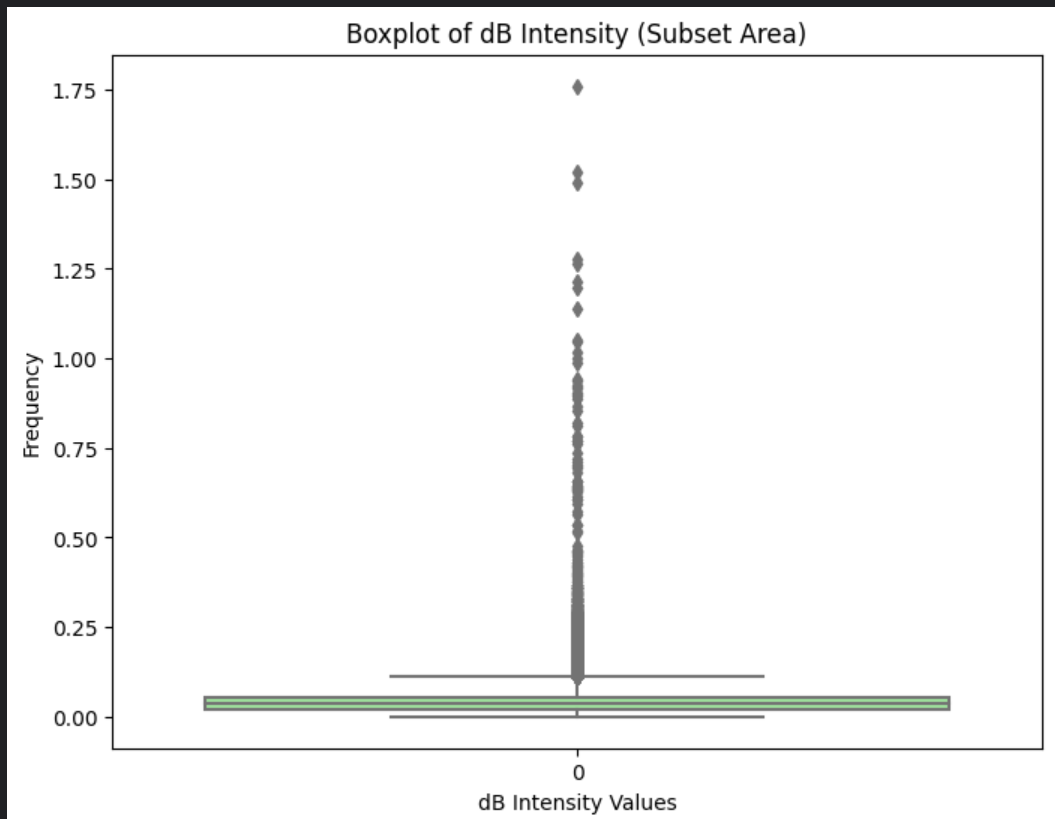
Band 4: min=0.000000, max=1.000000, mean=0.437432

**Inference from the statistics in reflectance values:**

Every band has a maximum of 1 and minimum of 0, suggesting a normal distribution. This indicates that data has been preprocessed for the validation

**5. Graphical Analysis of the Sentinel Image:**

**5.1 Box Plot of dB Intensity**

*Figure 5.1.0 db Intensity values for sentinel data*

**dB Intensity:** "dB intensity" (or often just "dB") refers to the backscatter intensity of the radar signal, expressed in decibels.

**Why dB Intensity matters?**

These values help analyze:

* **Surface properties:** Different materials (water, vegetation, soil, urban structures) reflect light differently.
* **Spectral characteristics:** Each band in satellite imagery (Blue, Green, Red, and Near-Infrared) captures unique reflectance patterns.
* **Vegetation health:** Healthy plants reflect more in the NIR band and absorb Red light.
* **Water identification:** Water bodies strongly absorb NIR and Red light but reflect Blue and Green.

**Key Components of Plot:**

* Median: The orange line inside the narrow box is the median dB intensity value. It's positioned very close to **0**. This tells us that the central value of the dB intensity in this subset area is around 0 dB.
* Interquartile range: In this plot, the box is quite **narrow**, spanning roughly from just below 0 to just above 0 dB. This indicates that the middle 50% of the dB intensity values are clustered very tightly around the median.
* Whiskers:
  + The **lower whisker** extends downwards from the bottom of the box to a dB intensity value slightly below **0 dB**. This shows the range of the lower 25% of the data (excluding outliers).
  + The **upper whisker** extends upwards from the top of the box to a dB intensity value around **0.1 to 0.15 dB**. This shows the range of the upper 25% of the data (excluding the high outliers).
* Outliers: The individual diamond-shaped points plotted *above* the upper whisker are the outliers. There are a **significant number** of these, and they extend to much higher dB intensity values, reaching up to around **1.75 dB**. These points represent dB intensity values that are considerably higher than the main distribution of the data in this subset area.

**Inference:**

* Dominance of Low dB Intensity Values: The box itself is positioned very close to 0 dB and is quite narrow. This indicates that the majority (the central 50%) of the dB intensity values in this subset area are clustered around 0 dB.
* Low Median Backscatter: The orange line within the box represents the median dB intensity, which is also very close to 0 dB. This suggests that the typical backscatter intensity in this area is low.
* Positive Skewness: The upper whisker extends significantly higher than the lower whisker (which is close to the bottom of the box). This indicates a positive skew in the distribution. There are more data points with higher dB intensity values compared to lower ones, even though the majority are low.
* Presence of Numerous Outliers with High dB Intensity: The individual points plotted above the upper whisker represent outliers with considerably higher dB intensity values. There are a substantial number of these outliers, and some reach very high dB levels (up to around 1.75 dB).

**What do we infer about the aoi?**

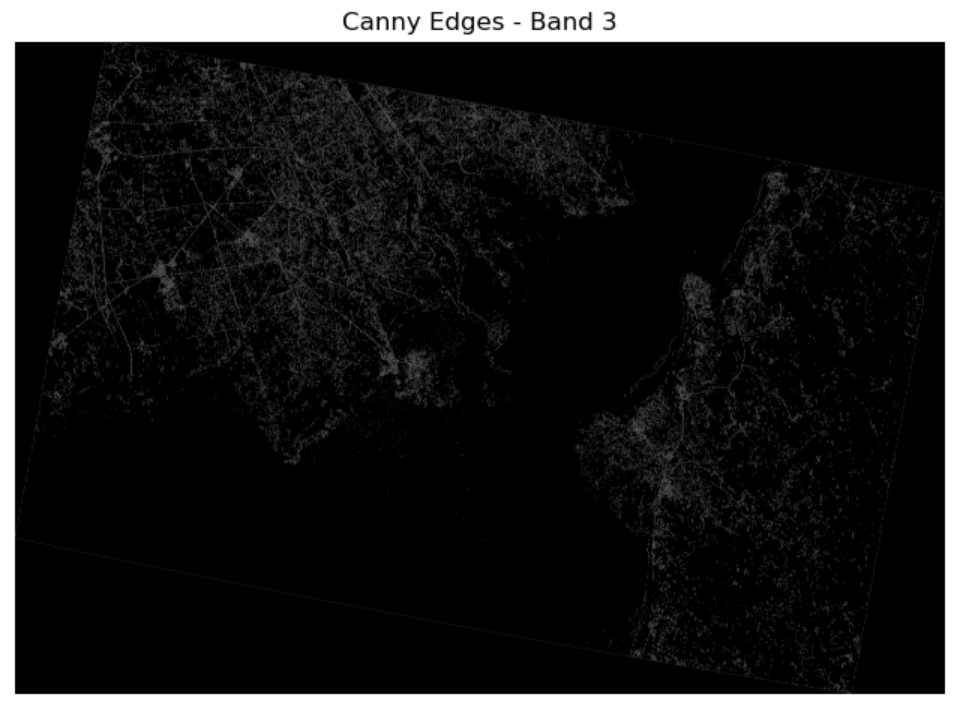
* Predominantly Smooth Surfaces: The dominance of low dB intensity values suggests that a large portion of this subset area likely consists of relatively smooth surfaces that cause specular reflection away from the sensor. This could include calm water bodies or smooth, bare soil.
* Scattered Strong Reflectors: The presence of numerous high dB intensity outliers indicates the presence of strong reflectors within this area. These could be:
* Urban structures: Buildings, metal objects, etc., cause strong backscatter due to corner reflections.
* Rough surfaces: Very rough terrain or dense, complex vegetation can lead to higher backscatter.
* Features with high dielectric constant: Certain materials or moisture conditions can increase reflectivity.
* Limited Variability in the Majority: The narrow box suggests that the backscatter characteristics are quite consistent across the central 50% of the area.

**6. Canny Edge Detection:**

The Canny edge detection algorithm is a robust method that aims to find accurate and thin edges by:

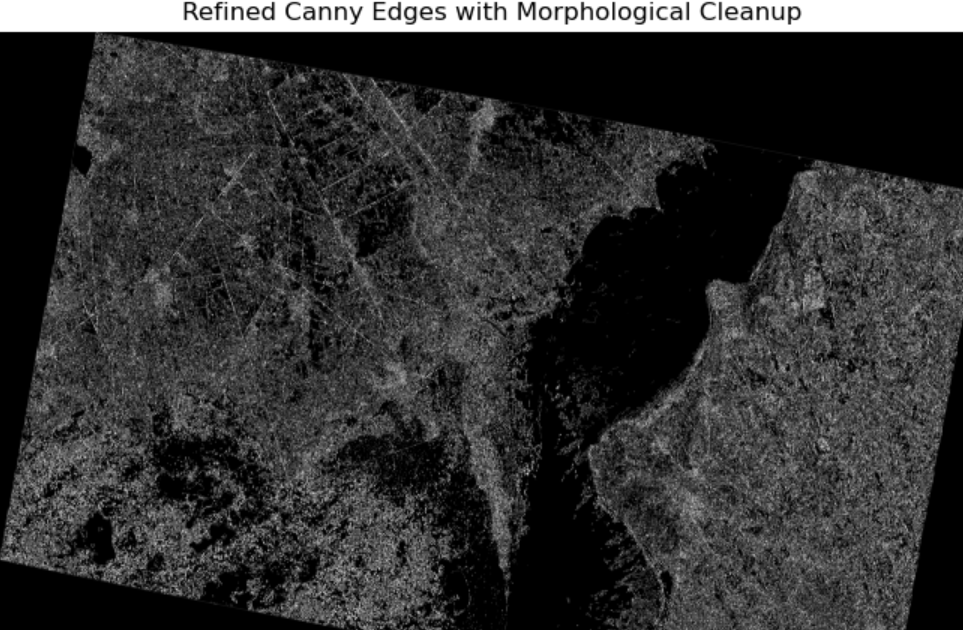
* Smoothing the image to reduce noise.
* Calculating gradient magnitude and orientation to identify potential edges.
* Suppressing non-maximum pixels to thin the edges.
* Using double thresholds to classify edge pixels.
* Applying hysteresis to link edges and reduce false positives.

**6.1 Canny Edge Detection of the Planetscope dataset:**

We perform the regular Canny edge on band-3 detection on the image and get the image below.

*Figure 6.1.0 Canny Edge Detection in planetscope*

As the image is not clear and still has some noise we use Morphological Cleanup to get refined canny edge detection.



*Figure 6.1.2 Refined Canny Edge after morphological cleanup*

**6.1.2 Morphological Cleanup:**

* Noise Residue: Even with the initial Gaussian blur, some small noise artifacts or isolated pixels might still be detected as edges by the Canny algorithm.
* Broken or Disconnected Edges: The Canny algorithm might produce broken or slightly disconnected edges, especially in areas with subtle intensity changes or noise.
* Spurious Small Edge Features: You might have tiny, irrelevant edge segments that you want to remove for cleaner results.
* Filling Small Gaps: Sometimes, you might want to close small gaps in otherwise continuous edges.

Common Morphological Operations for Cleanup:

1. Erosion:  
   * Effect: Thins edges and removes small, isolated bright pixels (which can be noise). It can also break thin connections between objects.
   * How it works: A structuring element (a small kernel or shape) is passed over the binary edge image. An output pixel is '1' (edge) only if *all* pixels within the structuring element centered at that location are '1' in the input image.
   * Use case: Removing small speckles of noise detected as edges.
2. Dilation:  
   * Effect: Thickens edges and fills small holes or gaps within or between edge features.
   * How it works: An output pixel is '1' if *at least one* pixel within the structuring element centered at that location is '1' in the input image.
   * Use case: Connecting broken edges or filling small gaps in lines.
3. Opening:  
   * Effect: Smooths contours, breaks narrow isthmuses, and eliminates small protrusions and thin appendages. It's essentially an erosion followed by a dilation using the same structuring element.
   * Use case: Removing small, isolated edge elements while preserving the shape and size of larger edge features. It's good for removing "salt" noise in binary images.
4. Closing:  
   * Effect: Smooths contours, fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contours. It's a dilation followed by an erosion using the same structuring element.
   * Use case: Filling small gaps within edges or connecting nearby edge segments. It's good for removing "pepper" noise in binary images.

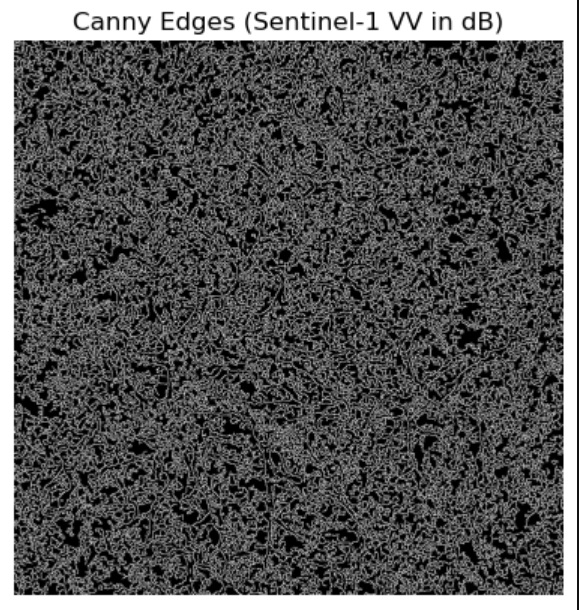
Applying Morphological Operations to PlanetScope Canny Edges:

For PlanetScope imagery, which often has relatively high spatial resolution, applying morphological operations can be particularly useful for:

* Cleaning up edges around small features: Removing noisy edges around individual trees, buildings, or field boundaries.
* Improving the continuity of linear features: Connecting edges of roads, rivers, or field edges that might have been broken by slight variations in pixel intensity.
* Removing isolated speckles: Eliminating small, isolated edge pixels that don't correspond to meaningful features.

Applying morphological cleanup after Canny edge detection on PlanetScope imagery is a valuable step for refining the edge results, reducing noise, and improving the connectivity and clarity of the detected features. The specific operations and structuring elements you use will depend on the characteristics of your imagery and the specific features you are trying to delineate.

**6.2 Canny Detection on Sentinel1:**

****

*Figure 6.2.1 Canny Edge Detection on Sentinel1 dataset*

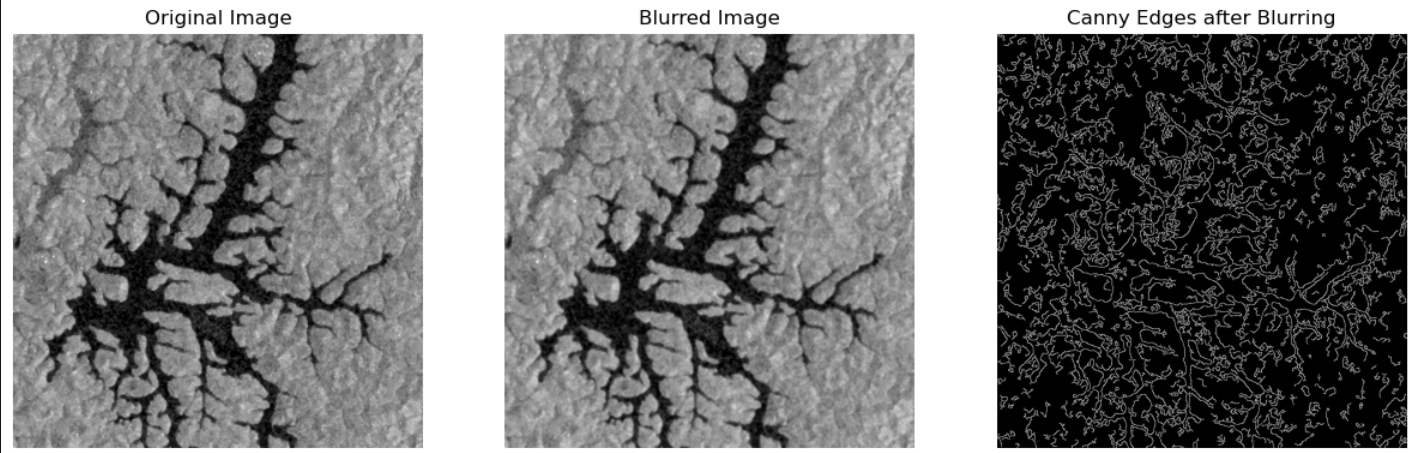
The above picture shows the canny edge detection in the sentinel dataset, as we can see the canny edge detection for the image is too rough so this also proves that its a hilly terrain apparently the canny detector detects even the noise for the edge so we will apply gaussian blur on this and then redo canny edge.

**6.2.1 Gaussian Blur:**

We apply Gaussian Blur because:

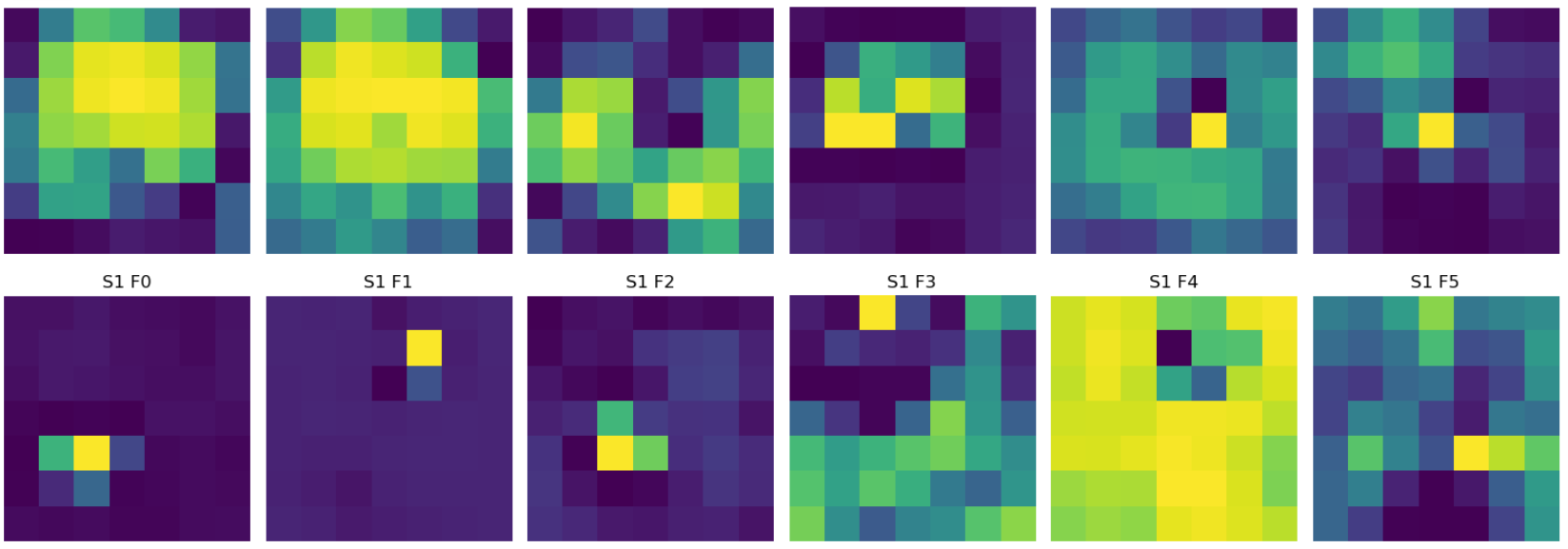
**Reducing Fine Details:** In some applications, you might want to suppress very fine-scale variations in the backscatter and focus on broader spatial patterns. Gaussian blur can achieve this by averaging pixel values over a defined neighborhood.

**Preprocessing for Certain Algorithms:** Some image processing or machine learning algorithms might perform better on data with reduced high-frequency noise, like speckle. Blurring can be a way to prepare the data for these algorithms.



*Figure 6.2.1.1 Comparison before and after gaussian blur*

**7. Feature Detection and Comparison of Sentinel and Planetscope:**

****

**Planet Features**

* F1 & F2: These filters exhibit widespread activations, suggesting sensitivity to large, homogeneous regions—possibly vegetation or open fields. The bright yellow areas indicate strong texture consistency or reflectance, meaning these features may highlight well-lit surfaces with minimal variation.
* F3: Displays a concentrated activation in a specific region, which suggests sensitivity to man-made structures or significant terrain features such as roads or dense urban areas. The suppression in surrounding regions implies high responsiveness to edges and sharp contrasts.
* F5: Isolated bright regions indicate potential edge or boundary detection. This feature may capture objects with high contrast differences, such as rooftops, vehicles, or shadowed areas. The selective activation suggests it is looking for defining elements in the landscape.

**Sentinel-1 Features**

* F1: Shows a strong, uniform response across the image, likely detecting large-scale flat surfaces with high radar reflectance. This could correlate with wet agricultural fields, built-up areas, or smooth terrain.
* F2 & F3: Present more localized responses, highlighting variations in surface roughness. These filters may be detecting roads, terrain discontinuities, or linear features like canals and land-use boundaries.
* F5: Displays a bright activation in the bottom-left, suggesting the presence of a highly reflective object—possibly a metallic structure, urban development, or another surface with strong radar return.

**Comparing PlanetScope and Sentinel-1 Features**

* PlanetScope features emphasize color, brightness, and texture, making them highly interpretable in visual imagery.
* Sentinel-1 features focus on structural, moisture-related, and roughness variations, providing insights that optical sensors may not capture.
* Complementary Insights: PlanetScope gives a natural appearance-based understanding, whereas Sentinel-1 provides deeper structure-based detail, making them valuable for joint analysis.

**8. Introduction to EfficientNet Models**

EfficientNet is a family of convolutional neural networks developed by Google AI, designed to achieve high accuracy while maintaining computational efficiency. The key innovation behind EfficientNet lies in its **compound scaling** method, which uniformly scales the network’s width (number of channels), depth (number of layers), and resolution (input image size) using a set of predefined coefficients. This balanced scaling approach significantly improves performance over traditional networks like ResNet and VGG with fewer parameters and reduced training time.

There are eight variants in the EfficientNet family, ranging from **EfficientNet-B0 to EfficientNet-B7**, with B0 being the baseline and the higher-numbered models offering increased capacity and accuracy. The B0 model, in particular, strikes a good balance between speed and accuracy, making it ideal for modest computational environments and medium-sized datasets like those commonly encountered in remote sensing applications.

In this project, **EfficientNet-B0** was chosen for its computational efficiency and suitability for transfer learning. Its backbone serves as the feature extractor in our classification pipeline, applied to high-resolution PlanetScope and Sentinel-1 imagery patches.

To classify land cover types—Vegetation, Urban, and Water—a multi-stage workflow was implemented combining remote sensing tools and deep learning techniques:

#### **Step 1: Preprocessing in SNAP**

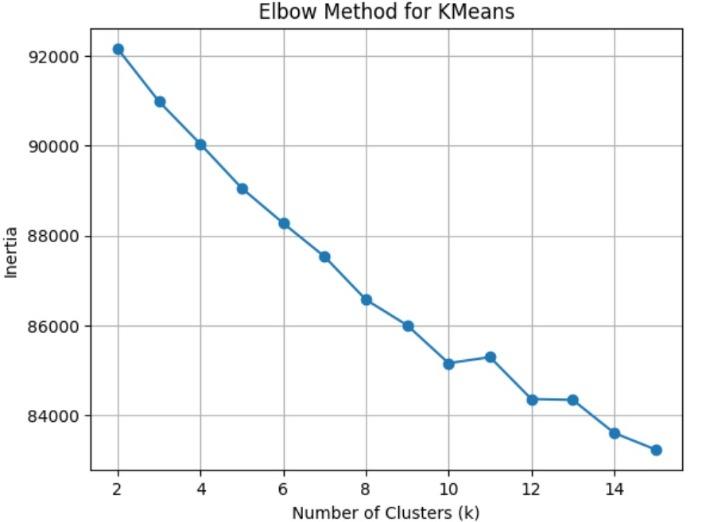
Both PlanetScope (optical) and Sentinel-1 (SAR) images underwent preprocessing using **ESA’s SNAP software**:

* **For Sentinel-1**:  
  + *Radiometric Calibration* was applied to convert raw backscatter values to standardized sigma-naught (σ⁰) values.
  + *Speckle Noise Filtering* (Refined Lee filter) was used to reduce noise while preserving edges.
  + *Terrain Correction* aligned the image geometrically using the Copernicus 30m DEM.
* **For PlanetScope**:  
  + Although already ortho-rectified, the data was visually assessed and cropped to the Area of Interest (AOI).

Both outputs were saved in GeoTIFF format for further processing.

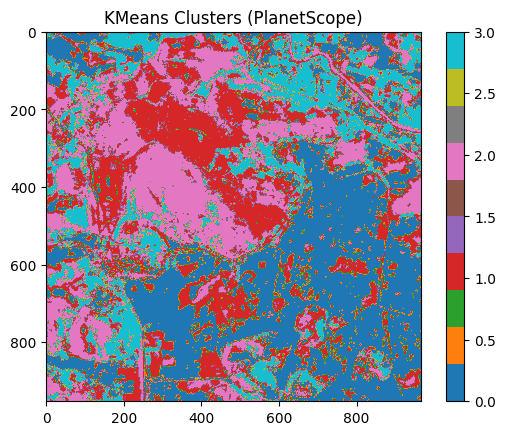
#### **Step 2: Unsupervised Classification**

To determine the optimal number of clusters, we applied the **Elbow Method**, which evaluates how the within-cluster sum of squares (inertia) decreases with increasing values of *k*. The idea is to identify a "knee" point after which additional clusters yield diminishing returns in reducing

inertia.

From the curve, it can be observed that the elbow occurs around **k = 4**, where the rate of decrease in inertia begins to level off. As a result, we also experimented with **k = 4**, which allowed inclusion of an additional class: **Road**, alongside Vegetation, Urban, and Water

The following is the planetscope image which was obtained when the number of clusters was set to 4 which are vegetation, water, Urban and Road.



#### **Step 3: Semantic Label Assignment**

Since KMeans does not inherently assign semantic meaning to clusters, each cluster was **manually mapped** to one of the land cover classes based on visual inspection and knowledge of the AOI. This mapping ensured consistency across both PlanetScope and Sentinel-1 imagery.

The result was a **labeled mask** representing the semantic categories for every pixel, forming the basis for supervised learning.

#### **Step 4: Patch Extraction**

To prepare data for deep learning, both the input images and their corresponding labeled masks were **divided into smaller patches** (e.g., 128x128 pixels). This approach has several advantages:

* Facilitates training on high-resolution imagery.
* Allows for data augmentation and balanced class distribution.
* Enhances the network’s ability to learn fine-grained spatial patterns.

Each patch pair (image and label) was stored systematically and used for model training.

#### **Step 5: Dataset Preparation for Deep Learning**

The dataset was structured into training and validation sets, with consistent image-label pairings. A range of data augmentations—such as flipping, rotation, and brightness variation—were applied to increase diversity and generalization ability during training.

#### **Step 6: EfficientNet Training**

A modified **EfficientNet-B0** model was trained using the prepared dataset. The model’s convolutional layers extracted spatial and spectral features from each patch, which were then classified into one of the three land cover types.

Key aspects of the training process:

* **Transfer Learning**: A EfficientNet-B0 (trained on ImageNet) was used as the base model to accelerate convergence and leverage existing feature representations.
* **Loss Function**: A categorical cross-entropy loss was used for multi-class classification.
* **Optimization**: Training was optimized using algorithms such as Adam or SGD, with learning rates tuned experimentally.

For semantic segmentation tasks, EfficientNet’s encoder can also be integrated with a decoder architecture (e.g., U-Net) to achieve pixel-level classification.

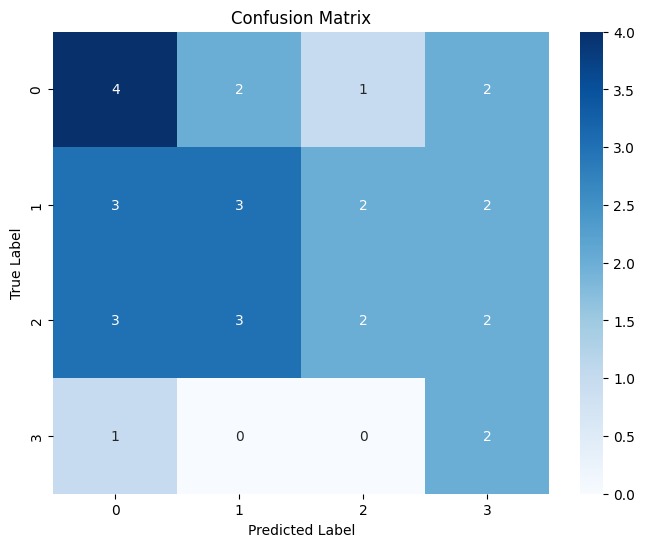
#### **Step 7: Evaluation and Visualization**

The trained model’s performance was evaluated using metrics like:

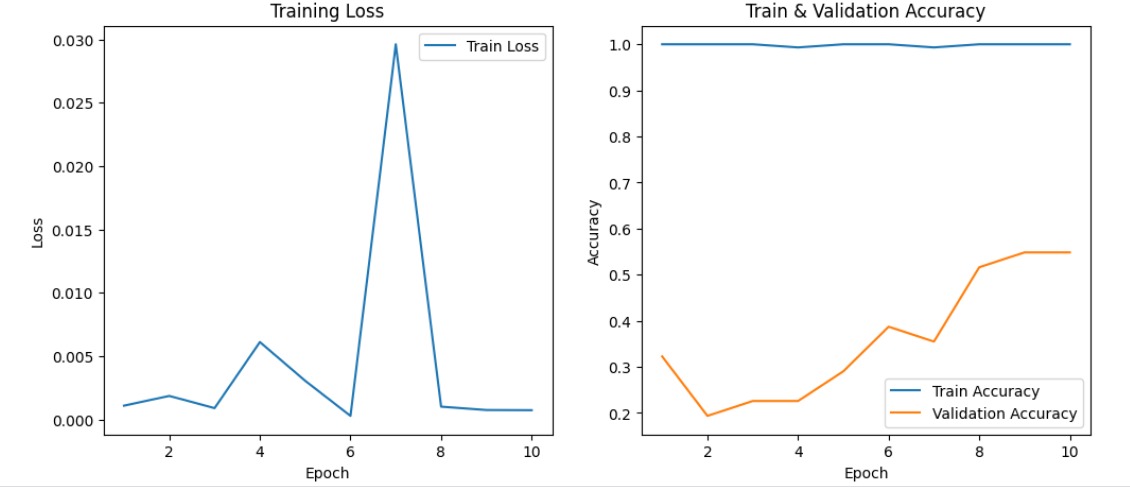
* **Accuracy**: Proportion of correctly classified pixels.
* **Confusion Matrix**: Identifies class-specific strengths and weaknesses.

Visualization techniques included overlaying predictions on true images, inspecting patch-wise performance, and generating heatmaps of confidence scores.

Given below is confusion matrix obtained after training:



Given below are loss and accuracy graphs of the model that was trained.



### 

### 

### **9.Hyperparameter Tuning and Model Optimization**

To improve model performance, several hyperparameter tuning strategies were applied during training with the EfficientNet-B0 architecture. The following key aspects were adjusted:

* **Learning Rate:** A relatively small learning rate (1e-4) was chosen to allow fine-grained weight updates and avoid overshooting during training.
* **Optimizer:** The **Adam** optimizer was used due to its adaptive learning capabilities.
* **Scheduler:** A **Cosine Annealing Learning Rate Scheduler** was implemented to dynamically reduce the learning rate over epochs, aiming to refine convergence.
* **Epochs:** The model was trained for 10 epochs, balancing training time and overfitting risk.
* **Data Augmentation:** The training pipeline included several augmentations to improve generalization:  
  + Random horizontal and vertical flips
  + Random rotations
  + Color jitter (brightness, contrast, saturation adjustments)

Despite these efforts, the validation accuracy did not improve significantly. This suggests that the limitations may not lie in hyperparameters alone, but also in:

* **Insufficient or imbalanced training data**
* **Spectral/radar similarity between classes (e.g., water vs shadowed urban)**
* **Label noise from unsupervised clustering (KMeans)**
* **Challenges in distinguishing mixed or edge pixels in high-resolution remote sensing imagery**

Future improvements could involve collecting more labeled data, refining the label generation process, or switching to architectures better suited for segmentation tasks.