The background of the entire image is a high-resolution aerial satellite photograph of agricultural land in Hawaii. The fields are organized into a grid pattern, with varying shades of green and brown indicating different crops or soil types. In the upper left corner, a small portion of the ocean and a coastal area are visible. The overall image has a slightly grainy texture.

REMOTE SENSING ANALYSIS

HAWAII LULC REPORT

INDEX

1. Planet-Scope Image

- 1.1 Acquiring the Data from Planet
- 1.2 Creating Box-Plots

2. Sentinel-1 Image and SNAP Pre-Processing

- 2.1 Acquiring Data From ASF Data Search
- 2.2 Preprocessing Steps in SNAP
- 2.3 Box Plot and Histogram

3. Comparison of S-1 Intensity image with Planet Reflectance

4. Canny Edge Detection

- 4.1 General Steps in getting the Canny Edges from a Preprocessed Data
- 4.2 Pipeline Employed to get Canny Edges from Raw ASF

5. Application of Efficient Net Models

- 5.1 Augmentation:
- 5.2 Patch Creation:
- 5.3 Labeling
- 5.4 Efficient New Model Training

Pratham Patel (23125025)

Project Code: [GitHub Code](#)

Manas Patil (23125020)

Vudit Aggarwal (23125037)

Amritanshu Tiwari (23125006)

Introduction

We know that remote sensing plays a key role in understanding and analyzing the Earth's surface, with applications ranging from land cover mapping and environmental monitoring to disaster response and urban planning. With advancements in satellite technology and data accessibility, we now have the tools to study specific regions in great detail using multiple types of satellite imagery. In this project, we focus on the region of **Hawaii** as our Area of Interest (AOI). Hawaii's geography is wide ranging and unique, varying from volcanic terrain, to oceans, to diverse vegetation, which makes it an interesting location for analysis.

For this study, we used imagery from two satellite sources: **PlanetScope**, which provides high-resolution optical data, and **Sentinel-1**, which is a radar-based satellite that captures intensity images using Synthetic Aperture Radar (SAR). We downloaded a cloud-free PlanetScope scene from 2024 using Docker, making sure that cloud cover was under 10–15%.

Our initial step was to generate boxplots for each spectral band using the Digital Number (DN) values in the image, which helped us understand the range and distribution of pixel intensities. Then, using tools provided by Planet, we converted these DN values into surface reflectance values, again through Docker. After the conversion, we generated another set of boxplots to study the changes in the data.

At the same time, we obtained Sentinel-1 SAR data for the same area using platforms like the ESA Hub, AWS Open Data, or ASF Data Search. Since SAR data is represented

in decibel (dB) scale, we processed the image using ESA's SNAP software and created boxplots to visualize its intensity distribution. Comparing the PlanetScope and Sentinel-1 images gave us an idea of how different sensors represent the same physical area and what kind of information each one captures best.

To explore more specific features in both datasets, we applied the Canny Edge Detection algorithm, which helps identify edges and sharp changes in the image. This allowed us to extract structural features like boundaries, coastlines, and other high-frequency elements.

Moving beyond traditional techniques, we also used a deep learning model EfficientNet to automatically detect features in both the PlanetScope and Sentinel-1 images. EfficientNet is known for its high accuracy and efficiency, making it a good fit for this kind of analysis. We further improved its performance by applying hyperparameter tuning, data augmentation, and learning rate optimization. The final results were evaluated using metrics like accuracy, precision, recall, and F1-score to see how well the model performed on our data.

The goal of this project is to combine traditional image analysis methods with deep learning techniques to better understand and compare multi-sensor satellite imagery. By working with data from both optical and radar sources and applying a mix of classical and modern approaches, we aim to gain deeper insights into the strengths and limitations of each method—especially when analyzing a complex and dynamic region like Hawaii.

1. Planet-Scope Image

1.1 Acquiring the Data from Planet

To begin our analysis, we selected Hawaii as our Area of Interest (AOI), considering its varied land cover, topography, and ecological diversity. The first step involved acquiring high-resolution optical satellite imagery from PlanetScope, a satellite constellation operated by Planet Labs. PlanetScope offers daily global coverage with a spatial resolution of approximately 3–5 meters, making it suitable for detailed regional analysis.

We started by identifying and generating the GeoJSON coordinates for our AOI using the tools provided on [Planet.com](https://planet.com). This file defines the geographic boundaries of our selected region and is required for querying satellite imagery via the Planet API. Once the AOI was finalized, we applied filters to ensure the cloud cover was below 10–15%, aligning with the project requirement for minimal atmospheric interference in the optical data.

```
{"type": "FeatureCollection", "features": [{"type": "Feature", "properties": {}, "id": 1, "geometry": {"type": "Polygon", "coordinates": [[[[-158.20048933, 21.4132616], [-158.146119, 21.4132616], [-158.146119, 21.45700279], [-158.20048933, 21.45700279], [-158.20048933, 21.4132616]]]]}}]}
```

Geojson Coordinates

After setting these parameters, we explored a suitable time frame to locate an appropriate PlanetScope scene that matched both our spatial and atmospheric criteria. Once the optimal image was identified, we used Planet's official API to request and download the image. This process was carried out by following the official documentation provided by Planet and executing the request through their Docker-based implementation.

The imagery returned from the API included a larger patch of satellite data than our original AOI. This is because Planet returns a scene that overlaps with the AOI but may exceed its exact boundaries to ensure complete coverage. To focus solely on our selected region, we imported the image into QGIS, an open-source Geographic Information System, and performed manual cropping using our original AOI boundaries from the GeoJSON file.

This process ensured that we acquired a high-quality, cloud-free PlanetScope image that precisely covered our area of interest, and was preprocessed and ready for further analysis in terms of boxplots, reflectance conversion, and machine learning-based feature detection.



Blue – ~455–515 nm



RGB Combined



Green – ~500–590 nm



Red – ~590–670 nm



Near Infrared (NIR) – ~780–860 nm

1.2 Creating Box-Plots

Every satellite image is made up of several spectral bands, which captures data at different parts of the electromagnetic spectrum.

Each pixel has a DN value recorded. A Digital Number (DN) is a numerical value that represents the amount of energy (light or radiation) detected by a satellite sensor for a particular pixel in a specific spectral band.

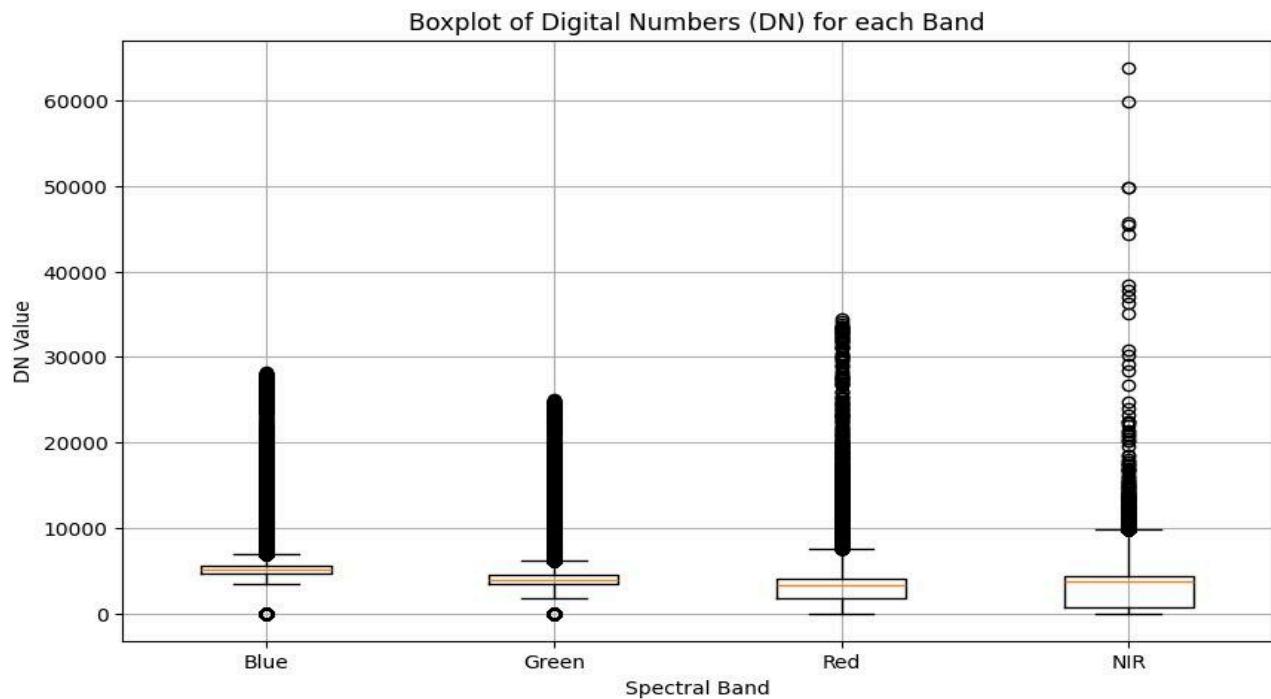
When a satellite captures an image, it doesn't store real-world measurements like reflectance or radiance directly. Instead, it measures the energy in each band (like red, green, blue, near-infrared, etc.) and assigns a discrete integer value, the DN, for each pixel.

DNs are sensor-specific and affected by:

- Sensor calibration
- Atmospheric conditions
- Time of day and season
- Illumination angle

To make the data physically meaningful and comparable across images or sensors, we convert DNs to reflectance or radiance using calibration coefficients provided by the

satellite provider.



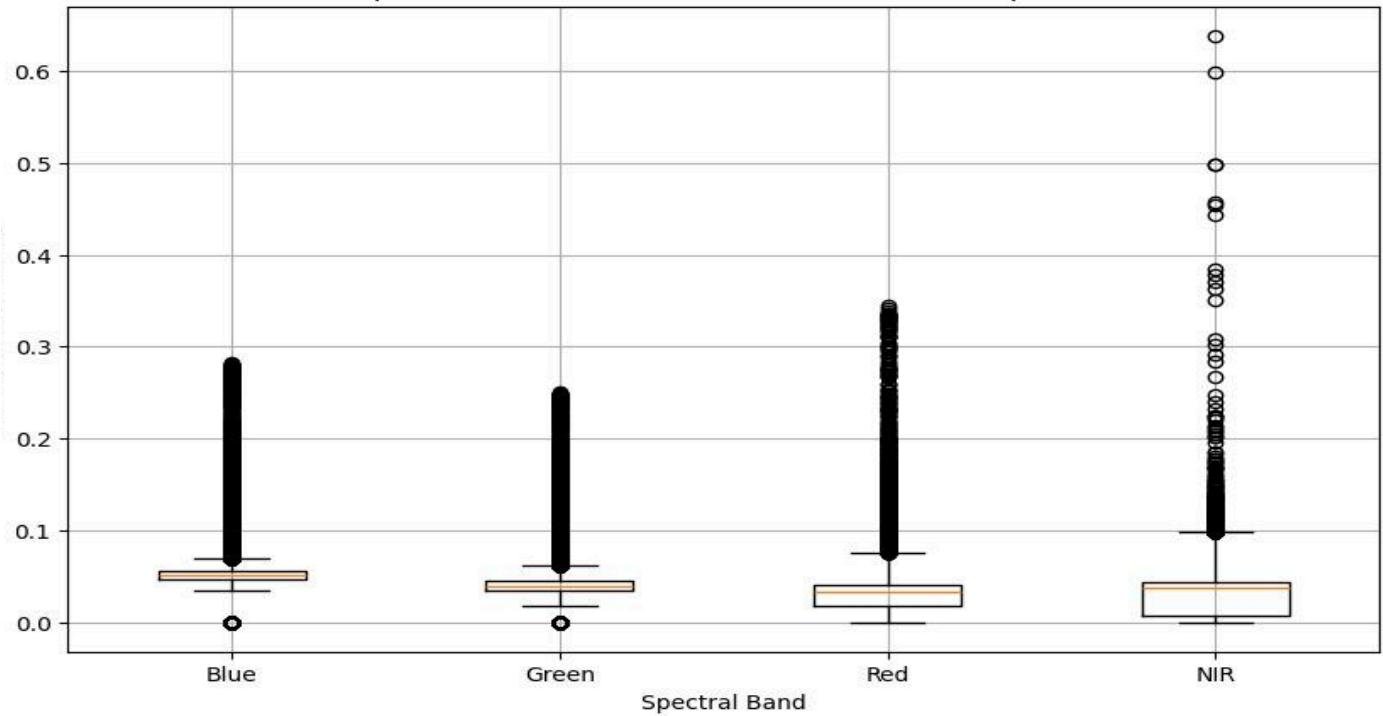
Boxplot for DN Values

*The boxplot shows the distribution of Digital Numbers (DN) across the Blue, Green, Red, and Near-Infrared (NIR) spectral bands. The **Blue** and **Green** bands have similar distributions, with moderate median DN values (~5000–6000 for Blue, ~4500 for Green) and many high outliers, suggesting consistent reflectance with occasional strong reflectors like urban surfaces or vegetation.*

*The **Red** band shows a slightly lower median (~4000) and greater variability, indicating more diverse land cover responses. Outliers are frequent, likely due to the mixture of vegetation and soil, as red light is strongly absorbed by healthy vegetation.*

*The **NIR** band stands out with the highest median (>6000), the widest spread, and extreme outliers reaching over 60,000. This reflects the strong response of healthy vegetation in the NIR range, making it especially useful for vegetation analysis.*

Boxplot of Reflectance Values for Each PlanetScope Band



Boxplot for Reflectance Values

These are the plots for the reflectance values of the previous box plots. We take reflectance values to convert raw satellite measurements into standardized, physically meaningful data that can be reliably used for scientific analysis and machine learning tasks.

2. Sentinel-1 Image and SNAP Pre-Processing

2.1 Acquiring Data From ASF Data Search

We downloaded the Sentinel-1 image through the **ASF (Alaska Satellite Facility) Data Search portal**, selecting a product that covered our AOI. The image included the **VV polarization band**, which represents vertical transmission and vertical reception of radar signals. Notably, Sentinel-1 data is delivered in a format where the image appears **reversed**.

All preprocessing was conducted using **SNAP (Sentinel Application Platform) software**, following a standard SAR data processing workflow. This ensured the data was corrected for various distortions and made ready for analysis and comparison with optical imagery.

2.2 Preprocessing Steps in SNAP:

1. Orbit File Application:

We began by applying the **precise orbit file** via Radar → Apply Orbit File, which improves the geolocation accuracy of the radar image. The output file is suffixed with _Orb.

2. Thermal Noise Removal:

Next, thermal noise was removed from the image using Radar → Radiometric → Thermal Noise Removal, producing a file suffixed with _tnr. This step eliminates background noise introduced by the sensor.

3. Radiometric Calibration:

We then calibrated the image through Radar → Radiometric → Calibration, selecting the option to output **Sigma0**, which is recommended for flat or moderately sloped terrain like much of our AOI. The output file is suffixed with

_Cal.

4. Speckle Filtering:

SAR images inherently contain **speckle noise**, which is a granular pattern that can obscure meaningful features. We applied a **Refined Lee Filter** through Radar → Speckle Filtering → Single Product Speckle Filter. This smoothing process produces a file suffixed with _Spk.

5. Terrain Correction:

Using Radar → Geometric → Terrain Correction → Range Doppler Terrain Correction, we corrected geometric distortions and aligned the image to real-world coordinates. We used the **Copernicus 30m Global DEM** and the **UTM/WGS 84** projection. This step outputs a properly georeferenced image.

6. Conversion to dB Scale:

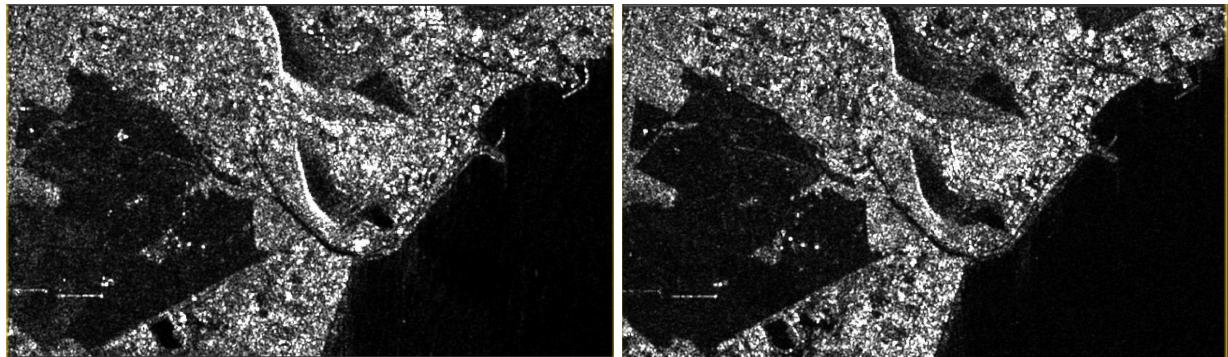
To improve visual interpretation and analysis, we converted the calibrated Sigma0 band to **decibel (dB) scale** using Right-click on band → Linear to dB. This produced a new virtual band named **Sigma0_VV_db**, which was then made permanent.

7. Cropping:

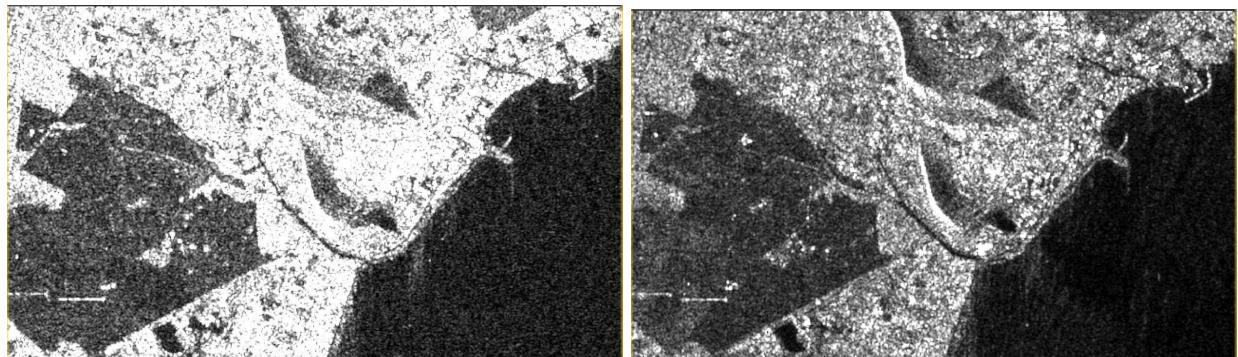
To focus on a more relevant portion of the data, we used Raster → Subset to crop the image to a manageable size, aligned with our AOI.

8. Export:

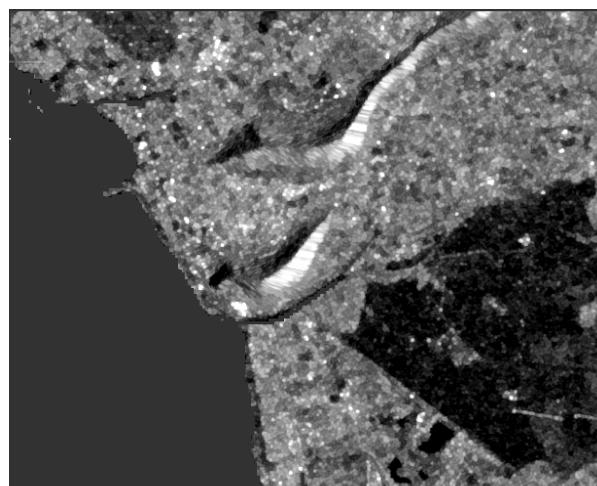
Finally, we exported the preprocessed image in **GeoTIFF format**, suitable for further use in Python scripts and visualization tools.



Intensity_VV and Intensity_VH



Amplitude_VH and Amplitude_VV



Sigma0_VV_db

The final image after applying the preprocessing steps in SNAP Software

2.3 Box Plot and Histogram

We generated histograms using Analysis → Histogram and created boxplots for the **Sigma0_VV** band. These helped us understand the data distribution and identify feature separability, although the histogram appeared **unimodal**, making land-water discrimination difficult.

```
#Box plot and Histogram for Sentinel - 1 Data

import rasterio

import numpy as np

import matplotlib.pyplot as plt


tiff_path = "/Users/manaspatil/Desktop/Acads/DAT/DAT/sigma0_VV_db.tif"

with rasterio.open(tiff_path) as src:

    band3 = src.read(1).flatten()

    band3 = band3[~np.isnan(band3)]


plt.figure()

plt.boxplot(band3, vert=True, patch_artist=True)

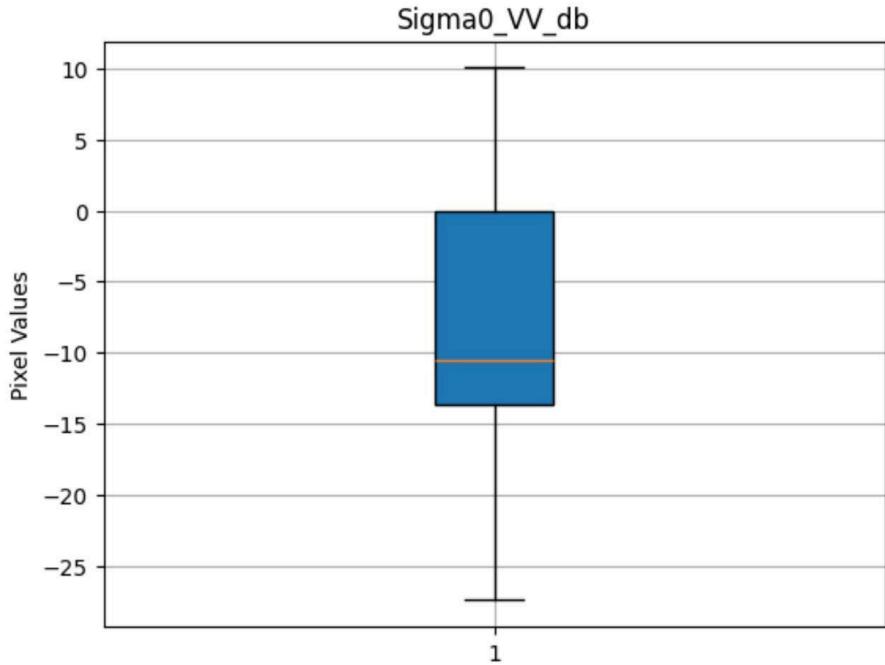
plt.title("Sigma0_VV_db - Boxplot")

plt.ylabel("Pixel Values")

plt.grid(True)

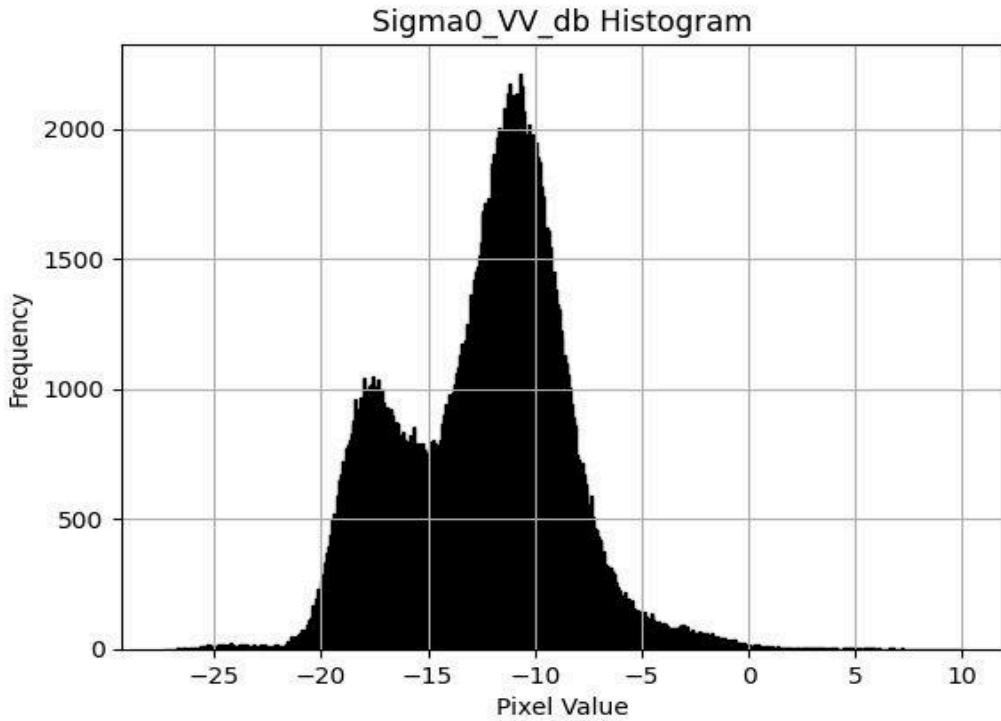
plt.show()
```

```
valid_data = band3[ (band3 != 0) ]\n\nplt.figure()\n\nplt.hist(valid_data.flatten(), bins=512, range=(-27.4311, 10.0404),\ncolor='skyblue', edgecolor='black')\n\nplt.title("Sigma0_VV_db - Histogram")\n\nplt.xlabel("Pixel Value")\n\nplt.ylabel("Frequency")\n\nplt.grid(True)\n\nplt.show()
```



Box Plot Formation

The box plot provides a summary of the pixel intensity distribution for the `Sigma0_VV_db` variable, which typically represents radar backscatter coefficients in decibels (dB). The median value is approximately around -12 to -13 dB, indicating the central tendency of the data. The interquartile range (IQR), which spans from the 25th percentile to the 75th percentile, lies between roughly -15 dB and -5 dB. This range captures the middle 50% of the pixel values. The whiskers extend further to about -27 dB on the lower end and to nearly 10 dB on the upper end, highlighting the presence of more extreme values in both directions. Notably, the longer upper whisker and wider spread above the median suggest a slight positive skew in the data distribution. The plot doesn't show any individual outliers, but the range and skew imply that there are a few higher backscatter values that might correspond to urban structures or other reflective surfaces in the radar image. Overall, the box plot reveals a moderately skewed distribution with a substantial spread in pixel values.



Histogram Plot

The histogram for **Sigma0_VV_db** provides a more granular view of the frequency distribution of pixel values, highlighting the detailed structure of the data. The most striking feature of the histogram is its bimodal distribution, with two prominent peaks. The first peak occurs around -18 to -16 dB, while the second, more dominant peak is centered around -12 to -10 dB. This bimodality suggests the presence of at least two distinct classes or land cover types within the radar scene, such as water bodies and vegetated areas, or built-up and natural surfaces. The distribution also exhibits a rightward (positive) skew, evident from the longer tail extending toward higher pixel values above 0 dB. This skew indicates that while high-reflectance surfaces are less frequent, they do exist and significantly affect the distribution. The histogram complements the box plot by revealing the underlying structure and frequencies of pixel intensities, confirming that the dataset contains heterogeneous surface types with varied backscatter characteristics.

3. Comparison of S-1 Intensity image with Planet Reflectance

PlanetScope Reflectance:

PlanetScope provides optical imagery in visible and near-infrared (NIR) bands. The values represent surface reflectance, indicating how much sunlight is reflected by the surface. These values are closely tied to surface characteristics like vegetation, water bodies, and built-up areas, and are affected by atmospheric conditions and illumination.

Sentinel-1 Intensity:

Sentinel-1 captures microwave backscatter using Synthetic Aperture Radar (SAR), specifically in the VV polarization band. The intensity values are expressed in decibels (dB) and reflect how strongly the radar signal is scattered back to the sensor. These values depend on surface roughness, structure, moisture content, and geometry, and are independent of lighting and weather conditions.

Aspect	Planet Reflectance (Blue, Green, Red, NIR)	Sentinel-1 Sigma0_VV_db (SAR Intensity)
Value Range	0 to ~1 reflectance	~ -28 dB to +10 dB
Median	Blue: ~0.5, Green: ~0.4, Red: ~0.32, NIR: ~0.37	~ -10 dB
Spread (Interquartile Range)	Relatively small, especially for Blue, Green, Red bands. NIR band shows more spread.	Very wide spread; large IQR shows high variability in pixel intensities
Outliers and Whiskers	Whiskers relatively short except NIR band (which extends toward 1.0)	Whiskers very long, indicating high variability, with extreme low and high values
Skewness (visual)	Slightly right-skewed in NIR; others are more symmetric	Left-skewed distribution (long tail on lower dB side)
Data Nature	Positive values (reflectance), normalized between 0 and 1	Decibel scale, includes negative values due to log transform of radar backscatter

Interpretation from Boxplots

Planet Reflectance Bands:

Generally less variable, values are constrained between 0 and 1.

NIR band shows the highest spread, likely reflecting vegetation variability.

Median reflectance values are moderate (~0.3–0.5), typical for terrestrial surfaces.

Whiskers indicate limited extreme reflectance values, except NIR with some high values.

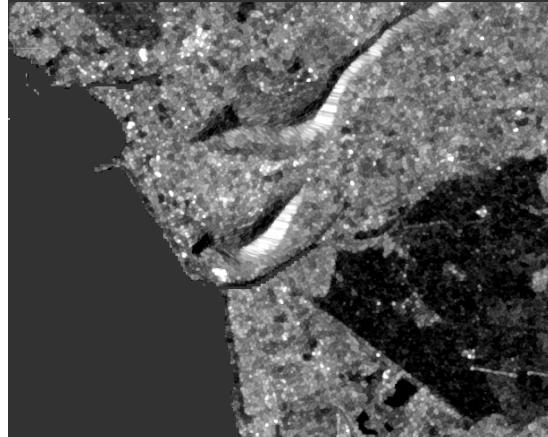
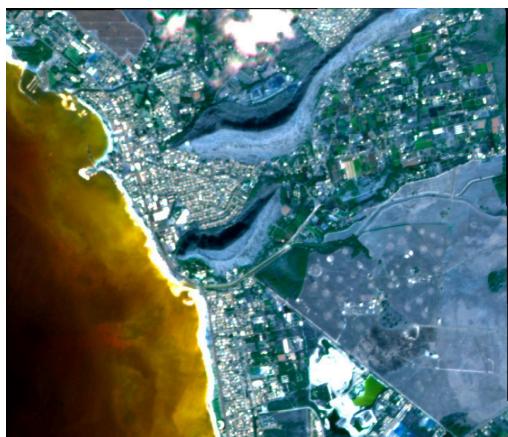
Sentinel-1 Intensity (Sigma0_VV_db):

Highly variable pixel values over a wide range (~38 dB spread).

Median around -10 dB indicates a strong bias toward lower backscatter values.

Long whiskers and boxes indicate high heterogeneity in surface backscatter properties.

Negative values are expected due to radar signal strength in dB scale.



Planet and Sential (preprocessed) Images

4. Canny Edge Detection

4.1 General Steps in getting the Canny Edges from a Preprocessed Data

After preprocessing, we applied Canny Edge Detection not just on the PlanetScope image, but also on the Sentinel-1 image using an enhanced SAR edge detection pipeline. This Python-based script incorporated adaptive filtering, speckle reduction, and improved detection algorithms tailored to the noise characteristics of SAR data. The aim was to extract clear and meaningful high-frequency features, like coastlines, structures, or abrupt land cover changes, that are often difficult to isolate in radar imagery.

Canny Edge Detection is a widely-used edge detection algorithm in computer vision and image processing. It aims to identify sharp intensity changes in an image that typically represent object boundaries, texture edges, or structural outlines. Unlike simple gradient-based methods, the Canny algorithm is known for its accuracy, low error rate, and ability to locate true edges with minimal false positives.

The algorithm follows several key steps:

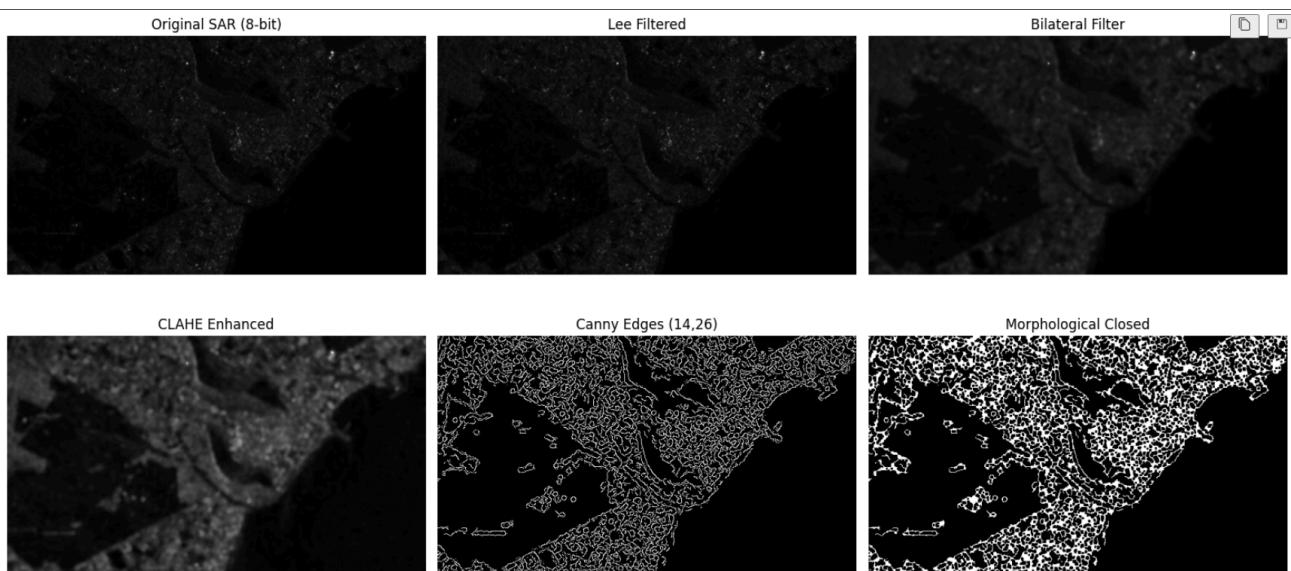
1. **Noise Reduction** – A Gaussian filter is applied to smooth the image and reduce noise.
2. **Gradient Calculation** – The intensity gradient (magnitude and direction) is computed to find regions with strong intensity changes.
3. **Non-Maximum Suppression** – Only the local maxima in the gradient direction are kept, ensuring thin and precise edges.

4. **Double Thresholding** – Two thresholds (high and low) are used to classify pixels as strong, weak, or non-edges.
5. **Edge Tracking by Hysteresis** – Weak edges connected to strong edges are kept, while isolated ones are discarded.

In the context of remote sensing and satellite image analysis, edge detection is particularly useful for:

- Highlighting structural features such as roads, coastlines, rivers, and buildings.
- Distinguishing land-water boundaries, vegetation patches, or geological formations.
- Simplifying image data before applying classification or deep learning models.

For SAR images like Sentinel-1, edge detection is more challenging due to speckle noise. To address this, we used a modified pipeline involving adaptive filtering and speckle reduction before applying the Canny algorithm. This helped enhance edge visibility and suppressed noise-related artifacts.



Canny Edge Detection for S-1 Image

Canny Edges For Planet data



Canny Edges (14,26)

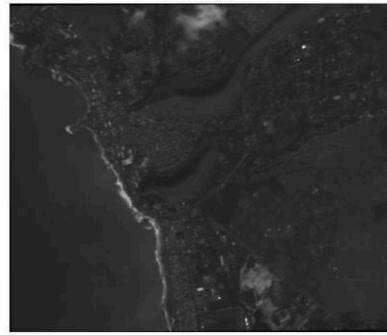


Edge Detection for Planet Data/S-1 Data

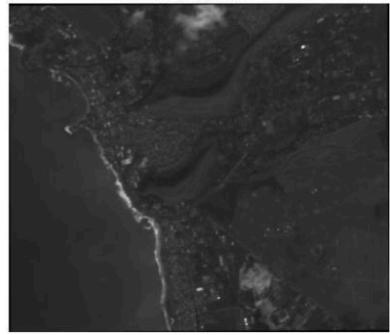
Original Planet Image (Normalized)



Lee Filtered



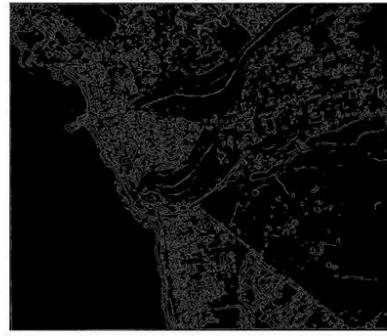
Bilateral Filter



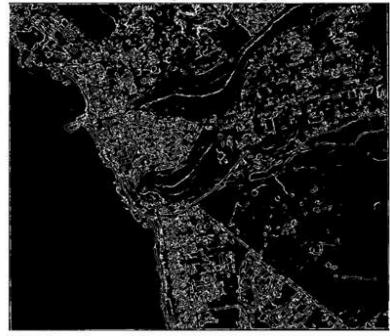
CLAHE Enhanced



Adaptive Canny Edges (17, 29)



Morphological Closed Edges



SAR Pipeline Output for PS-Data

4.2 Pipeline Employed to get Canny Edges from Raw ASF

1. Original SAR Image

The raw or terrain-corrected SAR intensity image in linear or dB scale

The backscattered radar signal strength, which varies depending on surface roughness, moisture, and structure.

Contains high levels of speckle noise, making it hard to visually or algorithmically identify consistent edges.

2. Lee Filtered Output

An image filtered using the Refined Lee Filter, a statistical speckle reduction technique.

A smoother version of the original image with reduced granular noise, while still preserving edges.

Helps suppress random speckle noise without overly blurring important structural features essential for accurate edge detection.

3. Bilateral Filtered Output

Output of the bilateral filter, a nonlinear filter that smooths images while preserving sharp edges.

A more refined and contrast-enhanced image where fine details and transitions are kept intact.

Improves visual distinction between features, especially useful when the intensity differences between edges and background are subtle.

4. CLAHE Enhanced Output

Image enhanced using Contrast Limited Adaptive Histogram Equalization (CLAHE).

A high-contrast image where local variations are emphasized, making weaker edges more visible.

Enhances faint or low-contrast features that might be lost in standard filtering. This step helps highlight subtle terrain boundaries and man-made structures before applying edge detection.

5. Canny Edge Output

The output from applying the Canny Edge Detection algorithm on the preprocessed SAR image.

A binary edge map where white lines represent detected edges—regions with sudden changes in intensity.

This is the primary output for identifying structural features such as coastlines, roads, field boundaries, or abrupt terrain changes. The quality of this result directly depends on the effectiveness of the previous preprocessing steps.

6. Morphological Closing Output

A morphological operation applied to the binary edge map to close small gaps in the detected edges.

A cleaned and connected version of the Canny edges, where disjoint or broken edges are linked into more meaningful continuous features.

This step improves structural integrity and visual clarity of detected edges, making them more suitable for further analysis or integration into classification models.

5. Application of EfficientNet Models

To extract meaningful semantic information from satellite images, we applied EfficientNet, a family of deep convolutional neural networks known for their balance between accuracy and computational efficiency. Our goal was to perform land use and land cover (LULC) classification, a common remote sensing task that involves categorizing regions of satellite imagery into predefined classes based on land type.

Before model training, we prepared the input dataset from the PlanetScope and Sentinel-1 images. This involved the following steps:

5.1 Augmentation:

All images were normalized to enhance contrast and bring pixel values into a standard scale, which helps the model converge faster during training and improves classification performance.

5.2 Patch Creation

We divided the AOI into 168 square patches, each representing a small, manageable area of the original image. These patches preserved spatial and visual context while allowing the model to learn features locally.

5.3 Labeling

Each patch was labeled into one of six LULC classes based on visual inspection and known geography:

Mountain, Ocean, Urban, Agricultural, Forest, Canal, Cloud

Below shown are two images, one is the satellite image unnormalized, and after augmentation we created the normalised image shown below it.

16-bit Satellite Image (Raw Band)



Normalized Satellite TIFF



Unnormalised (Above), Normalised (Below)

5.4 EfficientNet Model Training

We trained a series of EfficientNet models, B0 through B4, on the labeled patch dataset. Each model represents a progressively deeper and wider version of the EfficientNet architecture, with B0 being the lightest and B4 being significantly more complex.

All models were trained on the same dataset using consistent training parameters where possible, including data augmentation and hyperparameter tuning.

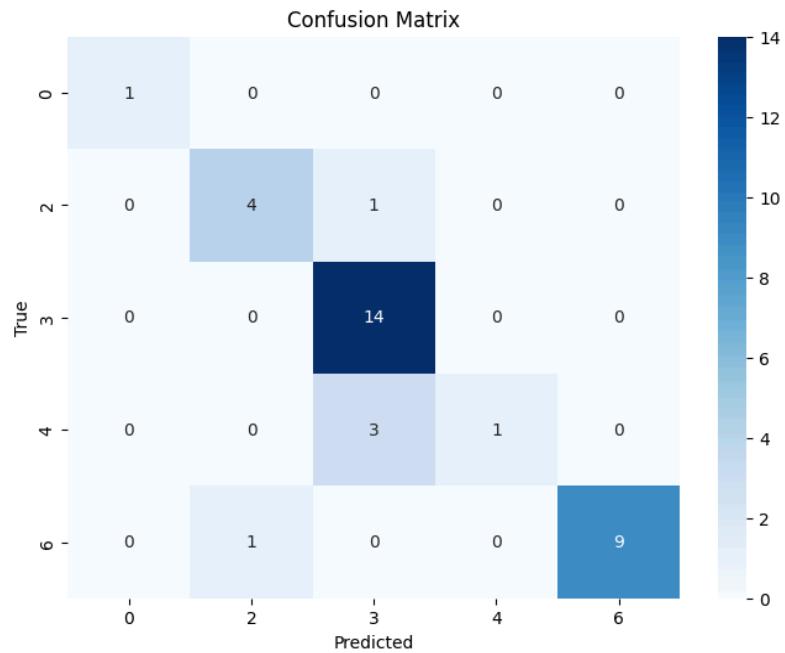
As part of hyperparameter tuning to identify the most effective configuration for the classification task using EfficientNet. Through systematic experimentation, We determined that an input resolution of 128×128 strikes a good balance between capturing spatial detail and maintaining computational efficiency. A dropout rate of 0.3 was found to be optimal for reducing overfitting while preserving model capacity. We used standard cross-entropy due to the simplicity of the Training Data. Finally, the AdamW optimizer with a learning rate of 1e-4 yielded the most stable and consistent training performance, making it the best choice among the optimizers tested.

Evaluating EfficientNet B0:

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
2	0.80	0.80	0.80	5
3	0.78	1.00	0.88	14
4	1.00	0.25	0.40	4
6	1.00	0.90	0.95	10

accuracy			0.85		34
macro avg	0.92	0.79	0.80		34
weighted avg	0.88	0.85	0.83		34

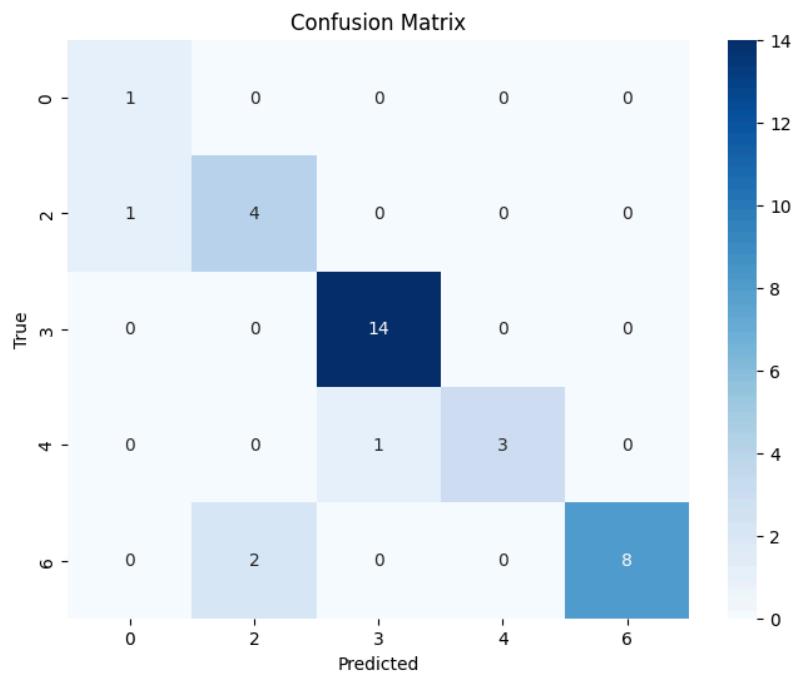


Evaluating EfficientNet B1:

Classification Report:

	precision	recall	f1-score	support
0	0.50	1.00	0.67	1
2	0.67	0.80	0.73	5
3	0.93	1.00	0.97	14
4	1.00	0.75	0.86	4
6	1.00	0.80	0.89	10

accuracy		0.88		34
macro avg	0.82	0.87	0.82	34
weighted avg	0.91	0.88	0.89	34

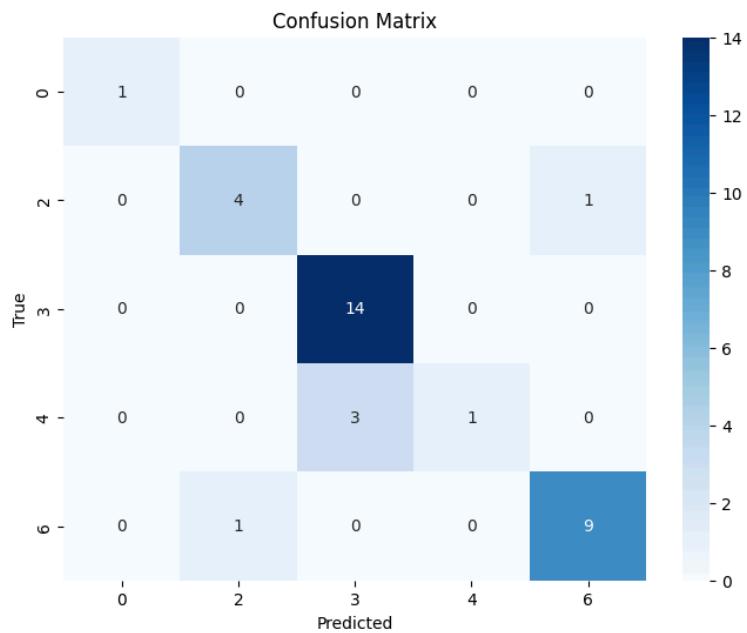


Evaluating EfficientNet B2:

Classification Report:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	1
2	0.80	0.80	0.80	5
3	0.82	1.00	0.90	14
4	1.00	0.25	0.40	4
6	0.90	0.90	0.90	10

accuracy			0.85	34
macro avg	0.90	0.79	0.80	34
weighted avg	0.87	0.85	0.83	34

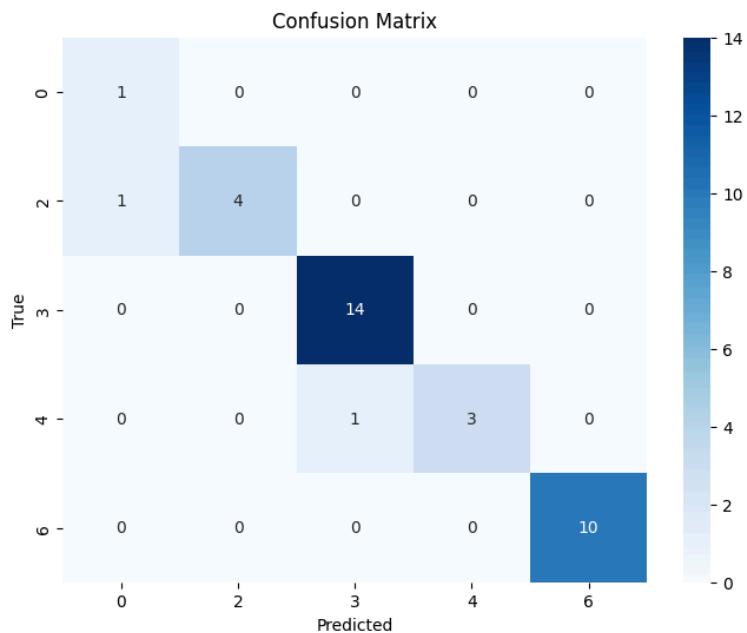


Evaluating EfficientNet B3:

Classification Report:

	precision	recall	f1-score	support
0	0.50	1.00	0.67	1
2	1.00	0.80	0.89	5
3	0.93	1.00	0.97	14
4	1.00	0.75	0.86	4
6	1.00	1.00	1.00	10

accuracy		0 .94		34
macro avg	0 .89	0 .91	0 .88	34
weighted avg	0 .96	0 .94	0 .94	34

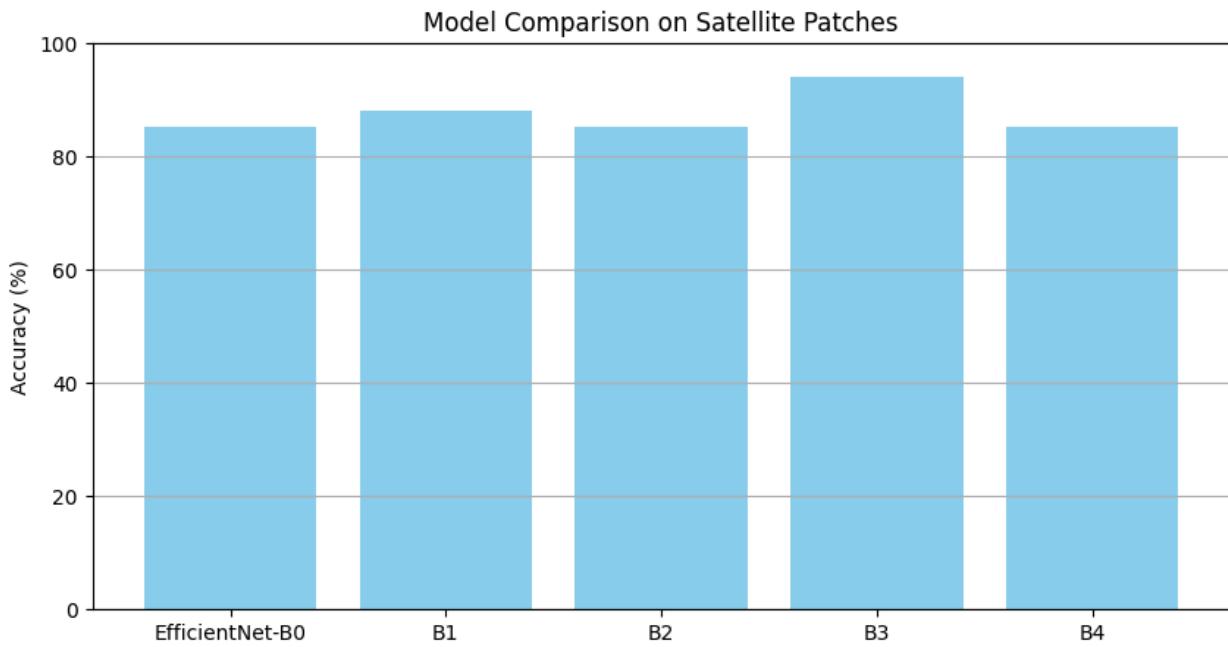
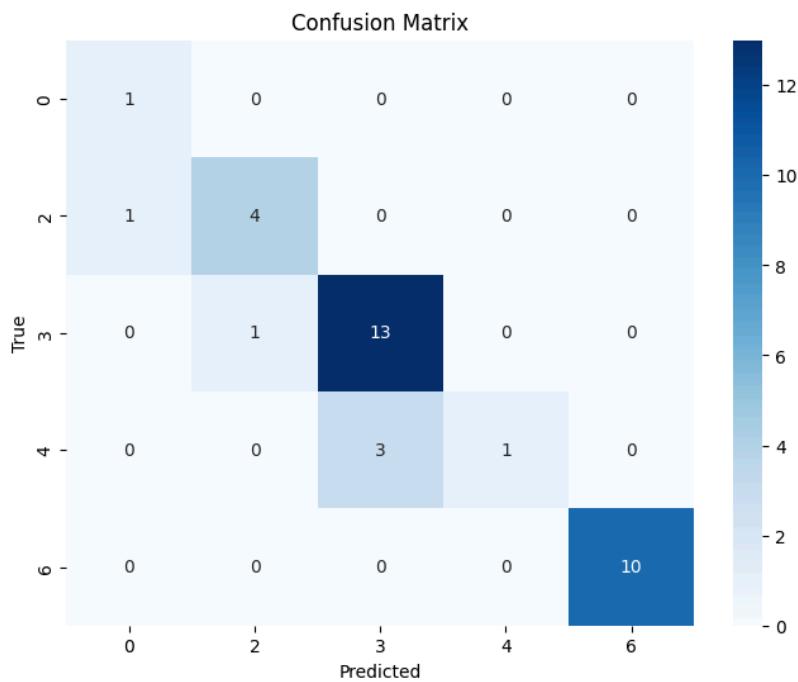


Evaluating EfficientNet B4:

Classification Report:

	precision	recall	f1-score	support
0	0.50	1.00	0.67	1
2	0.80	0.80	0.80	5
3	0.81	0.93	0.87	14
4	1.00	0.25	0.40	4
6	1.00	1.00	1.00	10

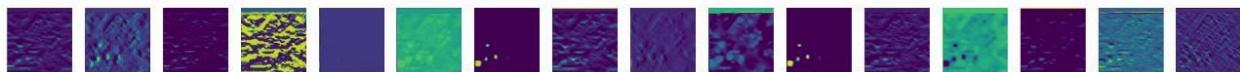
accuracy			0.85		34
macro avg	0.82	0.80	0.75		34
weighted avg	0.88	0.85	0.84		34



Among the EfficientNet variants trained on the satellite image patches, EfficientNet-B3 demonstrated the highest classification accuracy, outperforming all other models in the series. With an accuracy approaching 95%, B3 surpassed EfficientNet-B0, B1, B2, and B4, which hovered around 85–88%. This superior performance indicates that the B3 model, with its balanced architecture complexity and capacity, was better suited for capturing the relevant spatial and spectral features necessary for accurate land use and land cover classification in the dataset. The results suggest that increasing model complexity beyond B3 (e.g., B4) did not yield further accuracy improvements, possibly due to overfitting or limited training data.

However, the results from all of the models are still close to each other and thus show a similar range of performance across all models.

We also observe that the activation layer during the first 1-2 layers covered features relating to the texture of the terrain, and later on went onto more intricate and delicate features to differentiate classes.



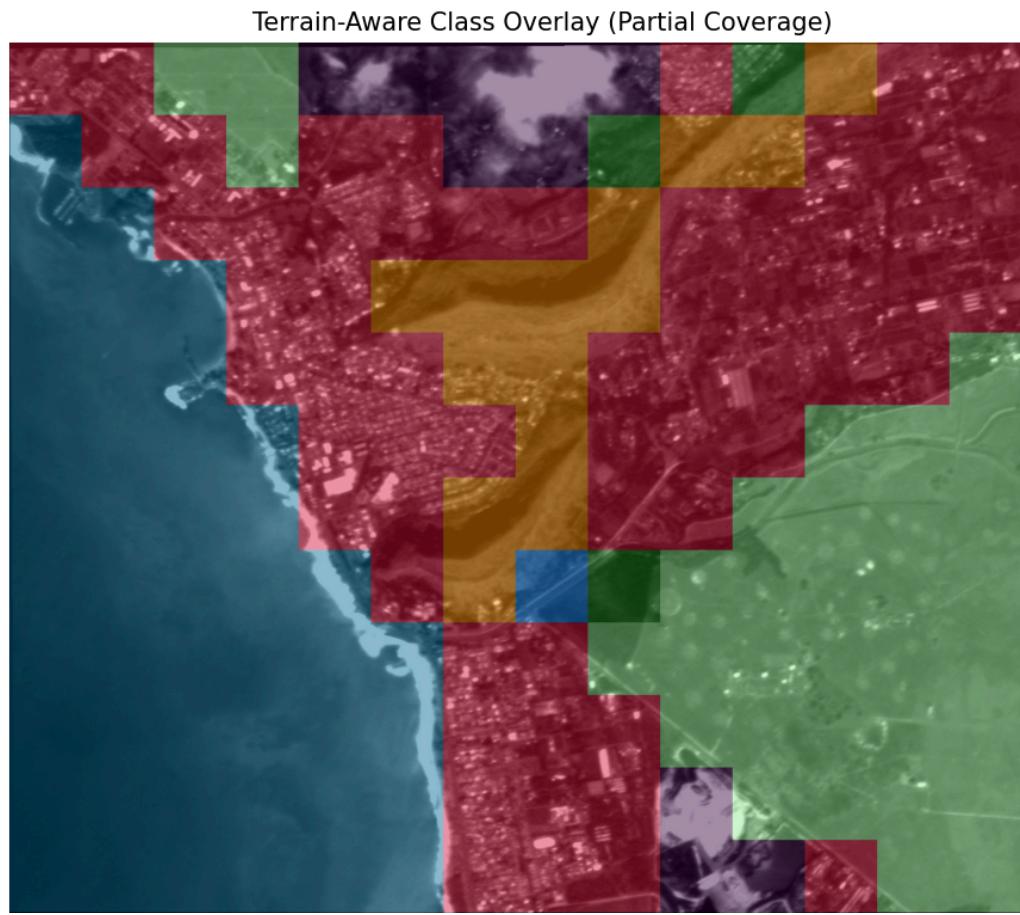
Our classification task essentially mirrors the process of **Land Use and Land Cover (LULC) mapping**, a fundamental application in remote sensing. LULC maps are used in urban planning, agriculture monitoring, environmental assessment, and disaster management.

Land-use classification schemes typically address both land use and land cover. A major land-use classification system developed by the United States Geological Survey (USGS) has multiple levels of classification. The categories within these levels are arranged in a

nested hierarchy. The most general or aggregated classification (level I) includes broad land-use categories, such as ‘agriculture’ or ‘urban and built-up’ land (Table 1). This level of classification is commonly used for regional and other large-scale applications.

Traditional land use classification uses local or regional information of the land, such as pixel signature or texture of remote sensing images. Due to the emergence of big data, land use classification now has new data sources. Mobile phone records, taxi trajectories, POI and check-in data have been used to extract the temporal variations of activities in a certain parcel and thus to derive the classes or functions of land parcels.

By using EfficientNet to classify image patches into land cover types, we replicated this process in a **supervised learning framework**. Each class we used—forest, urban, water, etc., is a standard category in LULC analysis. Thus, our model serves as a lightweight LULC mapping tool capable of performing semantic segmentation at patch-level resolution.



LULC Based on our classes

Our model with 168 patches was able to classify land with a sufficient degree of accuracy.

From the nature of the classification, we observe that a higher number of patches will lead to better and better classification and a more smooth transition between segments.

6. Conclusion

The objective of this project was to perform a comprehensive multi-sensor satellite image analysis for a selected Area of Interest (Hawaii), combining traditional remote

sensing techniques with modern machine learning approaches. Through this, we aimed to extract, analyze, and classify land surface features using both optical and radar data.

We began by acquiring high-resolution PlanetScope imagery and Sentinel-1 SAR data. For PlanetScope, we applied cloud cover filters and used Docker-based access via Planet's API to download a suitable scene. The Sentinel-1 data was obtained from ASF Data Search and processed extensively using ESA's SNAP toolbox to correct geometric distortions, remove noise, calibrate radiometrically, and enhance the data for visual interpretation. A specialized SAR edge detection pipeline—involving speckle reduction, adaptive filtering, contrast enhancement, and morphological operations—was used to extract meaningful edges from radar data.

A key component of the project was the comparison of Sentinel-1 intensity values and PlanetScope reflectance. This allowed us to observe how each sensor captured different surface properties. PlanetScope provided intuitive color information that highlighted vegetation and land cover, while Sentinel-1 was more effective at capturing structural features like terrain and built-up areas, especially in cloudy or low-light conditions.

Building upon this, we created 168 image patches from the AOI, labeled them into meaningful land use and land cover (LULC) classes—such as forest, ocean, urban, agricultural, mountain, and canal—and trained a set of EfficientNet models (B0 to B4) to classify these patches. Among these, EfficientNet-B3 emerged as the best-performing model, achieving the highest accuracy on the validation set. The use of deep learning in this context demonstrated the potential of modern architectures for automated, scalable LULC classification.

One of the key highlights was the reconstruction of a segmentation-style classification map, where the class predictions were overlaid on the original satellite image using transparency. This provided a visually intuitive and geospatially accurate way to represent the model's understanding of the terrain, preserving both class information and natural features.

6.1 Ups and Downs

The project was not without its challenges. Working with SAR data required careful preprocessing due to the presence of speckle noise and geometric distortions, and mastering SNAP's complex workflow was initially time-consuming. Ensuring alignment and consistent patching across different data sources also required precision and multiple iterations. On the deep learning side, balancing model complexity, preventing overfitting, and fine-tuning hyperparameters posed typical but meaningful hurdles.

On the upside, the integration of classical remote sensing techniques with deep learning yielded rich insights. We successfully explored how combining multi-sensor data, image processing, and neural networks can provide a comprehensive understanding of surface features, which would be hard to achieve using a single technique alone.

6.2 Final Thoughts

In conclusion, this project successfully met its objectives of acquiring, processing, analyzing, and classifying multi-sensor satellite imagery. It demonstrated the value of combining traditional GIS tools with AI-based models to generate land cover intelligence. With further refinement, such as expanding the labeled dataset, incorporating time-series data, or applying segmentation networks, this workflow could be scaled to perform real-time LULC monitoring and support a variety of environmental and planning applications.