Enhancing Water Body Detection with Satellite Image Feature Engineering

Team Members:
Laxmi Narayana Mangilipally - 11605829
Pranitha Pothuguntla - 11656849
Aashritha Sandiri - 11712554
Shasank Reddy Ganta - 11605399

Github

1. Introduction:

Water bodies play a critical role in various ecological, societal, and economic aspects, making their accurate detection and monitoring essential. Leveraging the capabilities of the Sentinel-2 satellite, our team aims to enhance water body detection through advanced feature engineering techniques applied to satellite imagery. This project holds significant promise for applications in water resource management, disaster response, climate change monitoring, and ecosystem assessment.
Feature engineering plays a critical role in enhancing the performance of water body detection algorithms. By extracting and transforming information from satellite imagery, feature engineering can effectively capture the unique characteristics of water bodies and distinguish them from other land-cover types. This report explores various feature engineering techniques that can be employed to improve water body detection from satellite imagery.
Water indices are a widely used class of features for water body detection. These indices exploit the specific spectral properties of water, such as its high reflectance in the near-infrared (NIR) band and its low reflectance in the green band. By combining multiple bands, water indices can effectively highlight water bodies and suppress the influence of other land-cover types.

Commonly used water indices include:

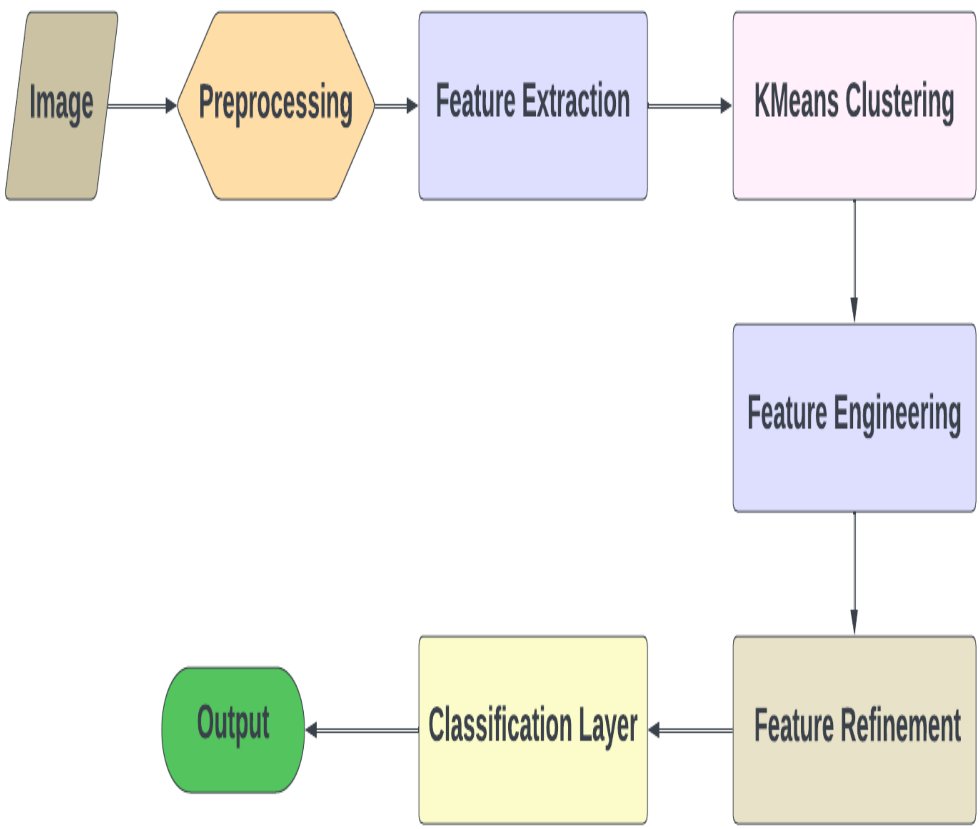
1. Normalized Difference Water Index (NDWI)
2. Modified Normalized Difference Water Index (MNDWI)
3. Automated Water Extraction Index (AWEI)

2. Background:

Dataset preparation involves collecting and preprocessing satellite images containing water bodies. Transforming the images from RGB color space to HSV color space helps to differentiate between the water bodies and other objects. Applied Gaussian Blur to reduce noise and smoothen the HSV images and Canny Edge Detection was used to identify edges and boundaries. By applying the median smoothing technique we can eliminate any remaining noise present in feature maps. Channel analysis assesses the effectiveness of the H, S, and V channels of the HSV image in identifying water bodies. Thresholding is applied to each channel to extract regions corresponding to water bodies. Segmentation mask generation combines the threshold feature maps to create the final segmentation masks, delineating the water bodies in images.

3. Model

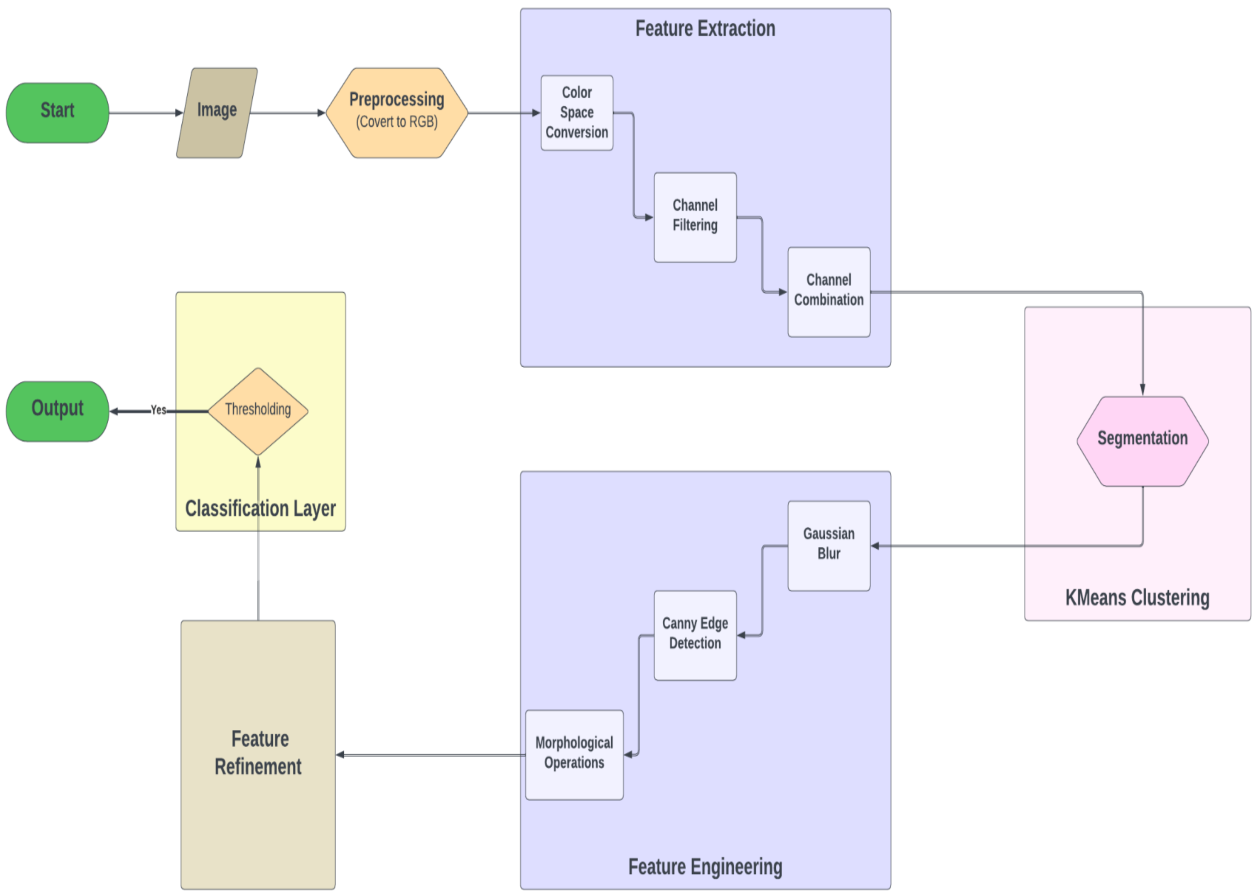
3.1. Architecture Diagram:



In our project, as shown in the architecture flow image first undergoes preprocessing, and then features are extracted from that preprocessed image, next we applied K-means clustering for segmentation and went through feature engineering layers to remove noise from the image. Then we refined the image and masked the image in binary terms which helped us to detect water bodies for non-water body regions.

3.2. WorkFlow Diagram:

As shown, waterbody detection pipelines the combination of feature extraction, K-means clustering, and feature engineering techniques.



This flow begins with a **Preprocessing** stage, which converts the input satellite image to RGB color space. The next stage is **Feature Extraction**, which has 3 levels

1. Color Space Conversion: RGB to HSV color space conversion.
2. Channel Filtering: At this level, we threshold each channel of the HSV image to isolate sensitive regions in the image.

3. Channel Combination: Hold filtered channel masks together so that we can easily create feature maps.

In the next stage, we apply **K-means Clustering** to the output of feature extraction to detect unique regions. This stage includes segmentation as well.

Next comes **Feature engineering layers**, we have performed Gaussian blurring to reduce noise and the edges have been smoothened. After removing noise and smoothing the edges we have identified, for which we have used the Canny Edge detection technique. After detecting edges morphological operations, which include dilation and erosion to refine the borders of water bodies.

After going through noise reduction and refining boundaries of water bodies, the output will pass through the Feature **Refinement layer.** This layer combines the original feature map with blurred ones to take the highest value of each pixel.

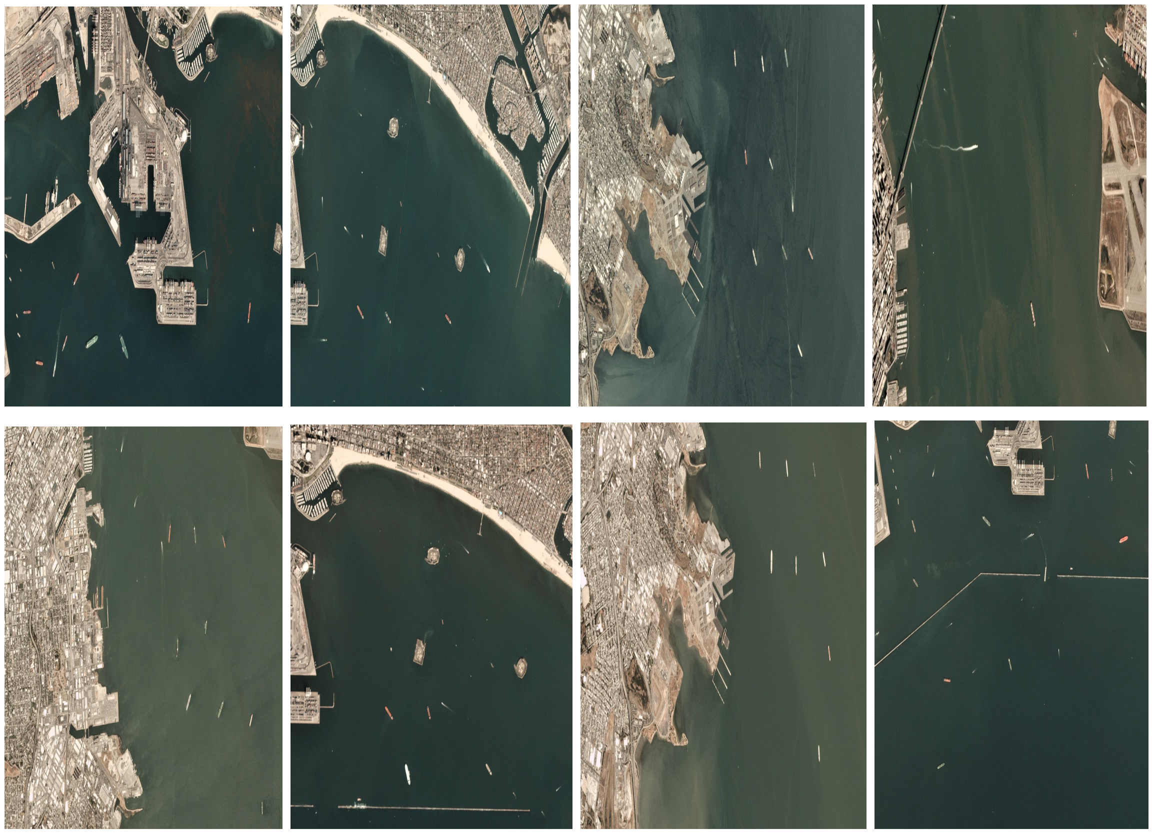
After getting high pixel values, we need to apply a threshold to apply feature maps to give a proper binary mask of water body detection. This level is named the **Classification Layer**.

As a result, we get the final output which is produced in the form of a binary water body detection mask. Where each pixel is labeled as 1 or 0, namely water or non-water respectively. This architecture is enhanced with k-means clustering to yield more accurate detection.

4. Dataset:

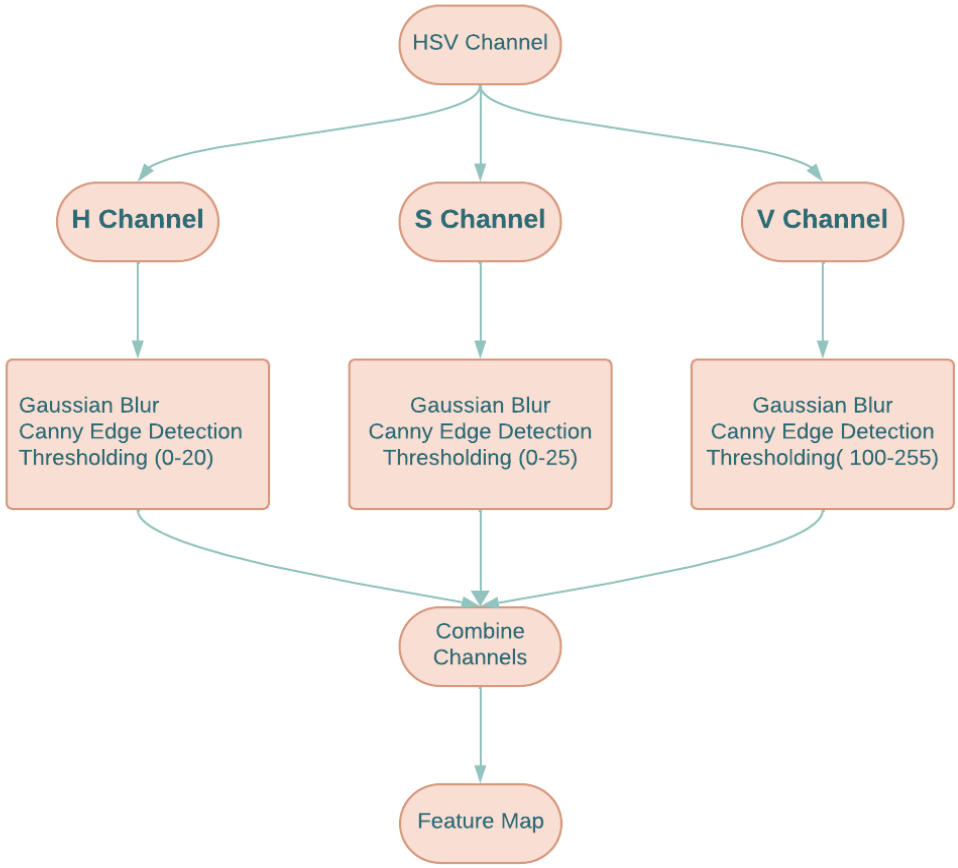
4.1. Description:

A collection of water bodies images captured by the Sentinel-2 Satellite. Each image comes with a black and white mask where black represents water and white represents something else but water. The dataset is the collection of images stored in the specified path i.e. **/content/Dataset/**. The images are loaded, processed, and analyzed to extract features that are related to edges, objects, and specific color channels in the HSV color space. The main objective is to identify and differentiate water bodies from other objects in the images based on the visual features extracted through various image processing techniques.
https://www.kaggle.com/datasets/franciscoescobar/satellite-images-of-water-bodies/data



4.2. Feature Design:

Color channels: Individual channels (H, S, V) of the HSV spectrum are analyzed for water body identification.
Edge Detection: Gaussian Blur and Canny edge detection are applied on HSV images this helps in separating water bodies from other objects in the image and is also applied on individual channels to identify object boundaries.
Custom Smoothing: A “median_smooth” function is implemented for potential future noise reduction in feature maps or predictions this removes noise if present.
Thresholding: Ranges are defined for H, S, and V channels this filters out irrelevant features and potentially highlights water bodies.
Combined Feature Maps: Outputs from edge detection and thresholding for each channel are combined to create a final feature map.

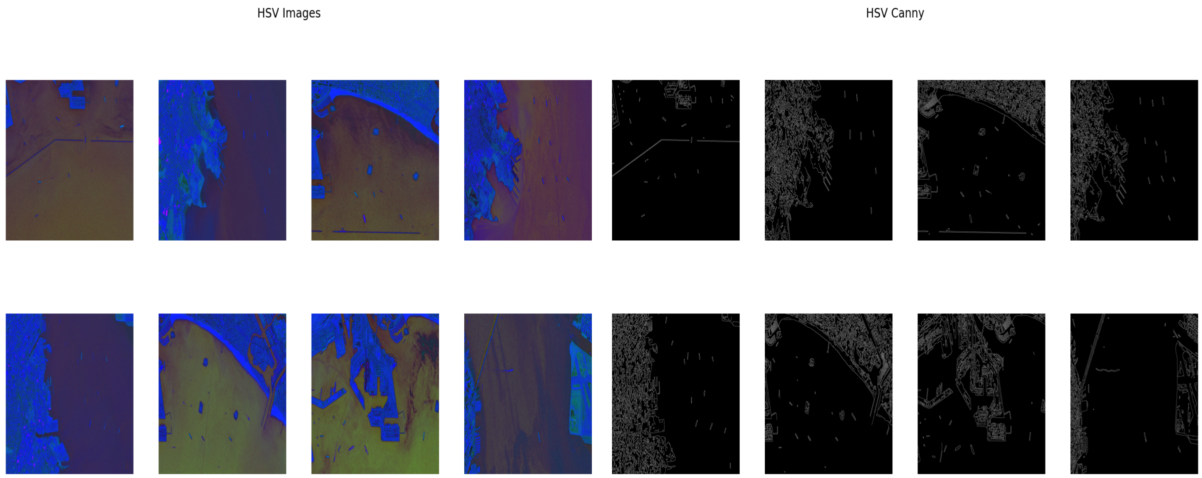


5. Analysis of Data:

5.1. Data Pre-processing:

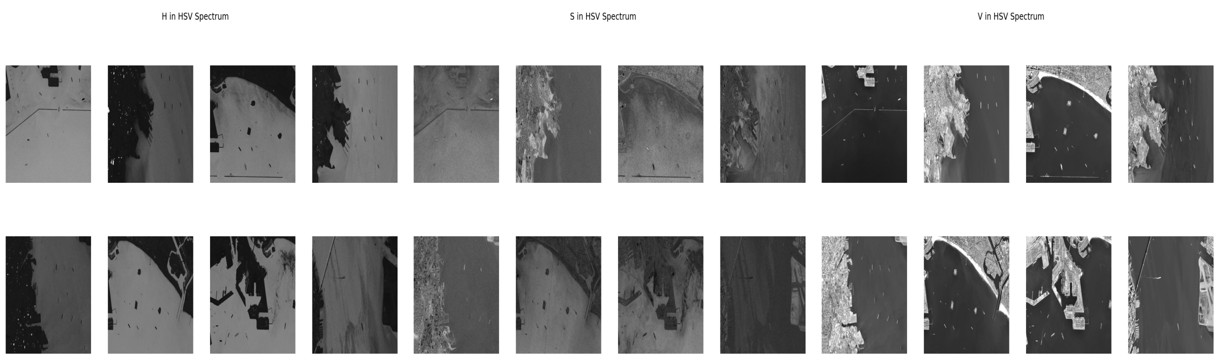
Loading Images:
Imported necessary modules and defined a helper function for plotting images. Images are loaded from specified paths in different color spectrums such as RGB, HSV, and Gray, and then resized to 1000 x 1000 pixels.

Canny Edge Detection Using HSV Spectrum:
Loading images in HSV color space to compare them with RGB images. HSV images play a better role in differentiating water bodies from other objects from images. Gaussian Blur and Canny Edge detection are applied to the HSV images, particularly to identify edges and objects in the images. The resulting images are displayed here:

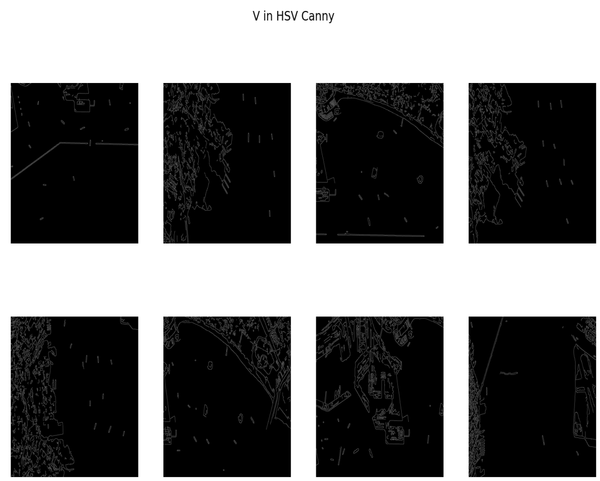


H, S, V Channels in HSV:
To gain insights into the information contained in each channel we plot individual channels on these channels Gaussian blur and Canny edge detection are independently implemented.

H, S, and V Images before applying edge detection techniques:



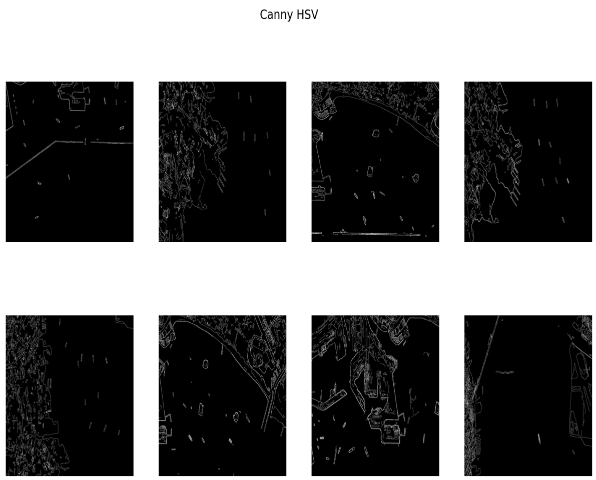
H, S, and V channels individually after applying edge detection techniques:



Combining Feature Maps:

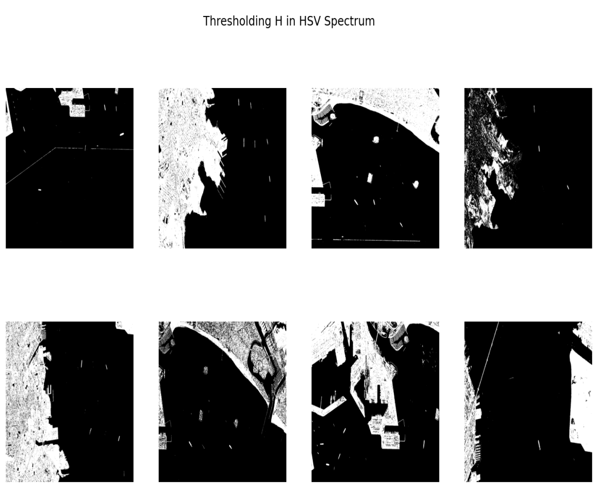
The outputs after the Edge detection using Gaussian Blur and Canny Edge detection for each channel are combined to create a final feature map. This provides the final edge detection map shown below.

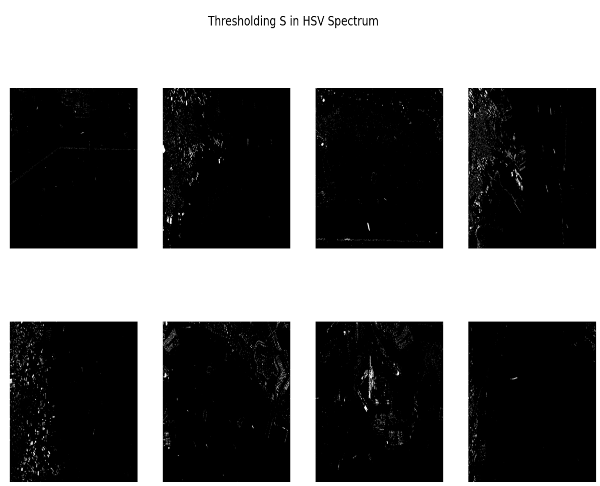
Feature map of H, S, and V channels after applying edge detection:



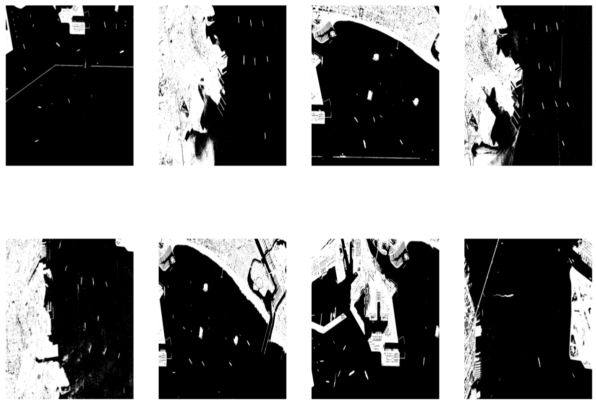
Thresholding:

Thresholding is applied to individual channels (H, S, V) in the HSV spectrum. Different thresholds are used for each channel and then thresholded feature maps are combined to generate final feature maps; those images are displayed below.

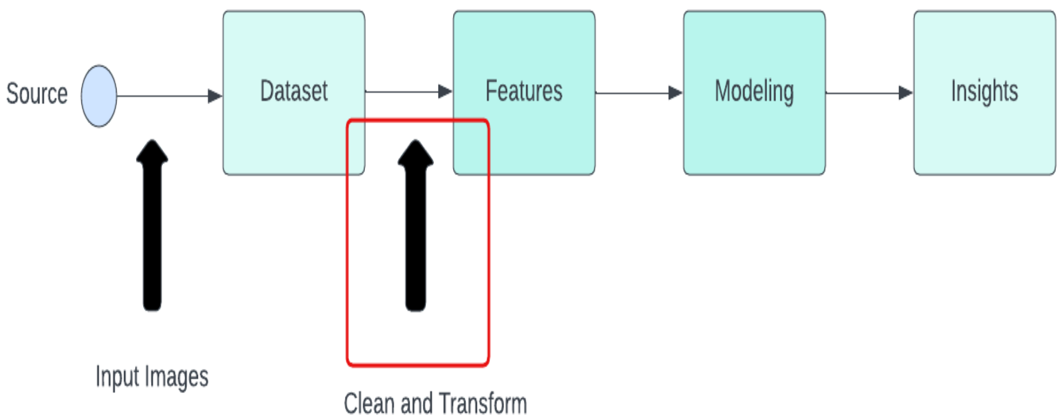




Thresholding V in HSV Spectrum



5.2. Graph Model:



Load Dataset:
Images are loaded from the specified path ('/content/Dataset').
Load and Plot Original Images:
RGB Images are loaded using OpenCV and plotted them for visualization and for future analysis.
HSV Conversion and Plotting:
RGB images are converted to the HSV color space.
HSV images are plotted for comparison with the RGB images
Edge Detection on HSV:
Gaussian Blur and Canny Edge Detection are applied to the HSV Images.
Channel Analysis:
Individual channels (Hue, Saturation, Value) in the HSV images are analyzed.
Gaussian Blur and Canny Edge Detection are applied to each channel.

The processed channels are plotted to understand their characteristics.
Combine Feature Maps:
Edge-detected channels are merged to generate feature maps.
Thresholding:
Applied to individual channels. These are merged to generate a single thresholded feature map.
KMeans Clustering:
This is performed on the HSV data for various values of K(2 to 5).
These are blurred based on KMeans cluster assignments.
Final Predictions:
The final combined feature maps are plotted.

6. Implementation:

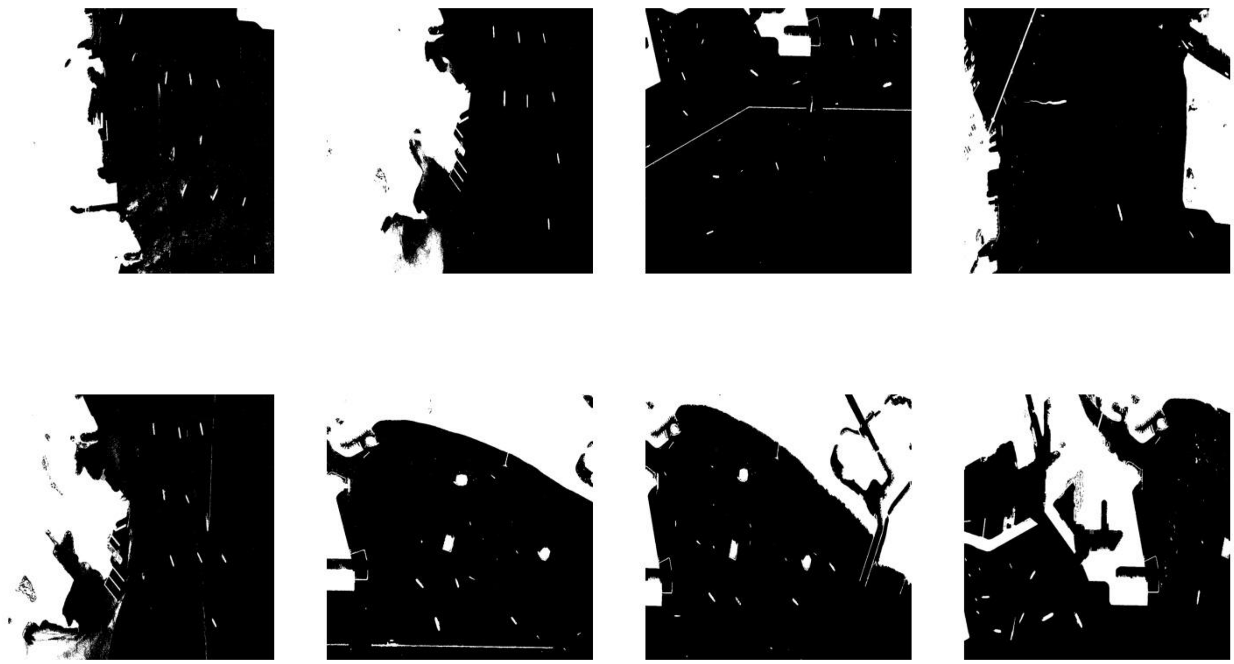
```
from sklearn.cluster import KMeans
kmeans = dict()
data = np.array(hsv_images)
data = data.reshape((-1, 3))
preds = dict()
for i in range(2, 6):
    kmeans[i] = KMeans(n_clusters=i)
    kmeans[i].fit(data)
    preds[i] = kmeans[i].predict(data)
```

This algorithm we used here is used for performing K-Means clustering on the flattened HSV images with the assignment of clusters ranging from 2 to 5 and storing them for further analysis and visualization. Several clusters can be adjusted based on the desired level of granularity in clustering as this is a hyperparameter.

7. Results:

By blending the results from thresholding, K-Means clustering helps enhance certain features and reduce the noise in the final predictions. From the below final predictions image we see the highlighted features or regions where the entire black mask represents the water body and white represents other objects, and from this, we can see that the black masking is more visible because the rest of the noise is detected and reduced.

Final Predictions



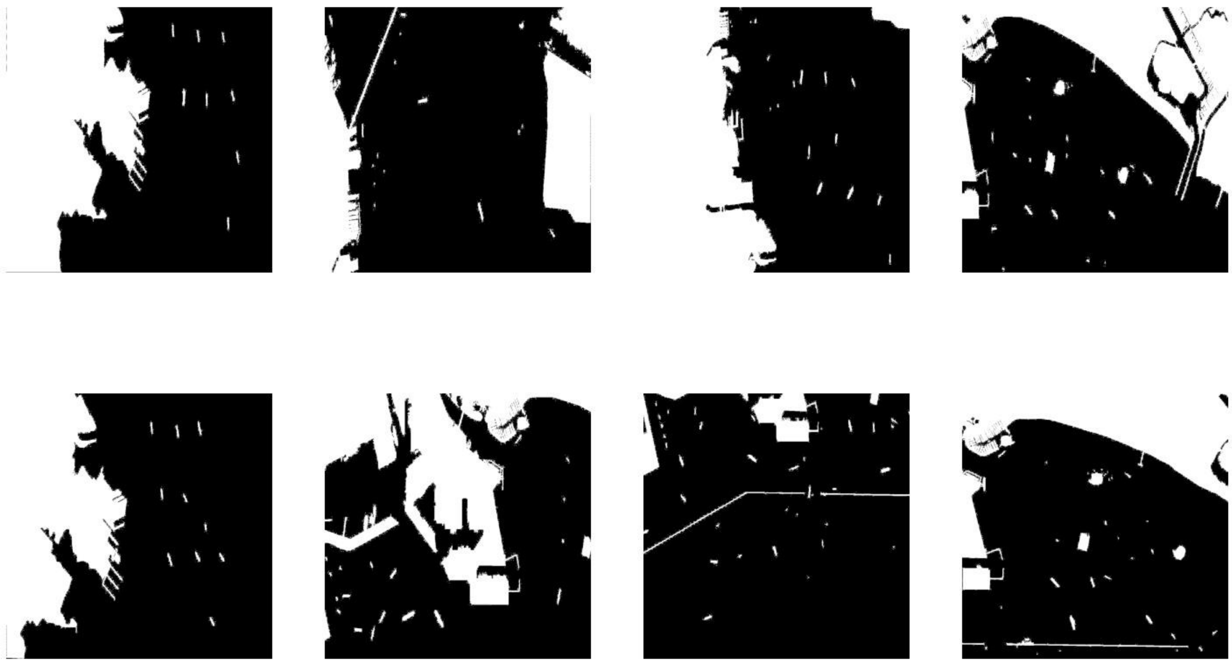
Ground Truth and its Importance in Water Body:

Ground truth refers to the actual, manually labeled information about the location of water bodies in an image. Obtaining accurate ground truth is essential for many reasons: Model Training and Evaluation, Identifying Model Bias and Errors, Choosing Hyperparameters and Algorithm Selection, etc.

It serves as the basis for calculating precision and recall scores, which measure the model's accuracy in identifying water bodies.

By using ground truth data we can effectively train and evaluate the water body segmentation model, leading to improved performance and real-world applicability.

Ground Truths



Evaluation Results of Water Bodies:

Precision and recall scores are calculated for each image to assess the segmentation accuracy.

```
Image 1 Scores:
Precision: 0.907109507948744     Recall: 0.9826370561774739
Image 2 Scores:
Precision: 0.9903656714802226    Recall: 0.9121162876484252
Image 3 Scores:
Precision: 0.984941538363287     Recall: 0.9800828776826023
Image 4 Scores:
Precision: 0.9743328118268598    Recall: 0.9315600069372182
Image 5 Scores:
Precision: 0.9178759490063394    Recall: 0.9578681993609163
Image 6 Scores:
Precision: 0.9910150738843098    Recall: 0.9220914241345503
Image 7 Scores:
Precision: 0.9925112694488876    Recall: 0.7790629101908041
Image 8 Scores:
Precision: 0.9905091239287003    Recall: 0.956658243663726
```

8. Project Management:

8.1 Implementation Status Report:

8.1.1 Work Completed:

Responsibilities and Contribution

Student	Task and Contribution
Laxmi Narayana Mangilipally	Contributed in code part in channel analysis and model building on K-Means Clustering, Ground Truth values also contributed in documentation - worked on the implementation and Results (25%)
Aashritha Sandiri	Data selection, Data visualization,Data Labeling and contributed to documentation - worked on Architecture and Workflow diagram (25%)
Pranitha Pothuguntla	Contributed in code part worked on Edge detection techniques and contributed in documentation on Dataset description and Feature design (25%)
Shasank Reddy Ganta	Contributed in code part and channel analysis, Smoothing technique, and Performance evaluation and also contributed to documenting the Data analysis i.e. Data preprocessing and Graph model. (25%)

9. References:

https://www.kaggle.com/datasets/franciscoescobar/satellite-images-of-water-bodies/data
https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123
https://ieeexplore.ieee.org/document/9498442
https://www.researchgate.net/publication/371498940_Deep_Learning_for_Automatic_Extraction_of_Water_Bodies_Using_Satellite_Imagery
https://www.mdpi.com/2072-4292/10/4/554
https://orbit.dtu.dk/en/publications/towards-automatic-monitoring-and-mapping-of-benthic-vegetation-us
https://www.mdpi.com/2072-4292/14/6/1429
https://www.mdpi.com/2073-4433/10/11/725