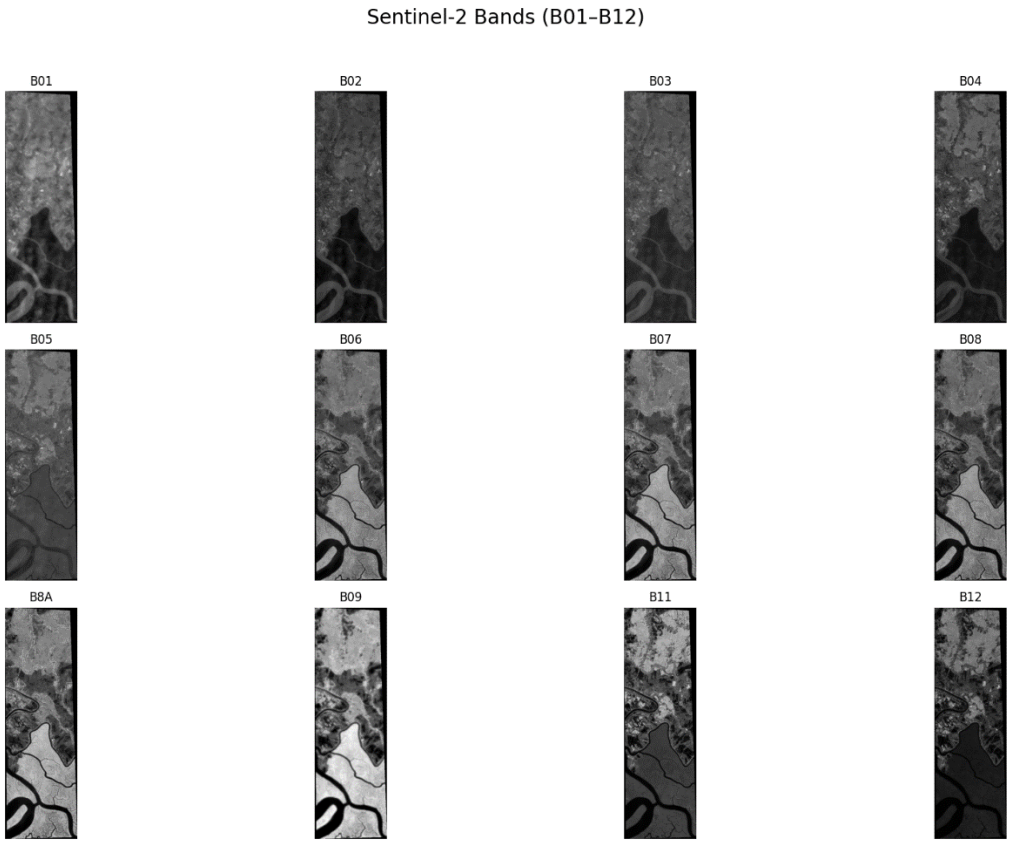
WASALATHILAKA.N.S.

E/20/425

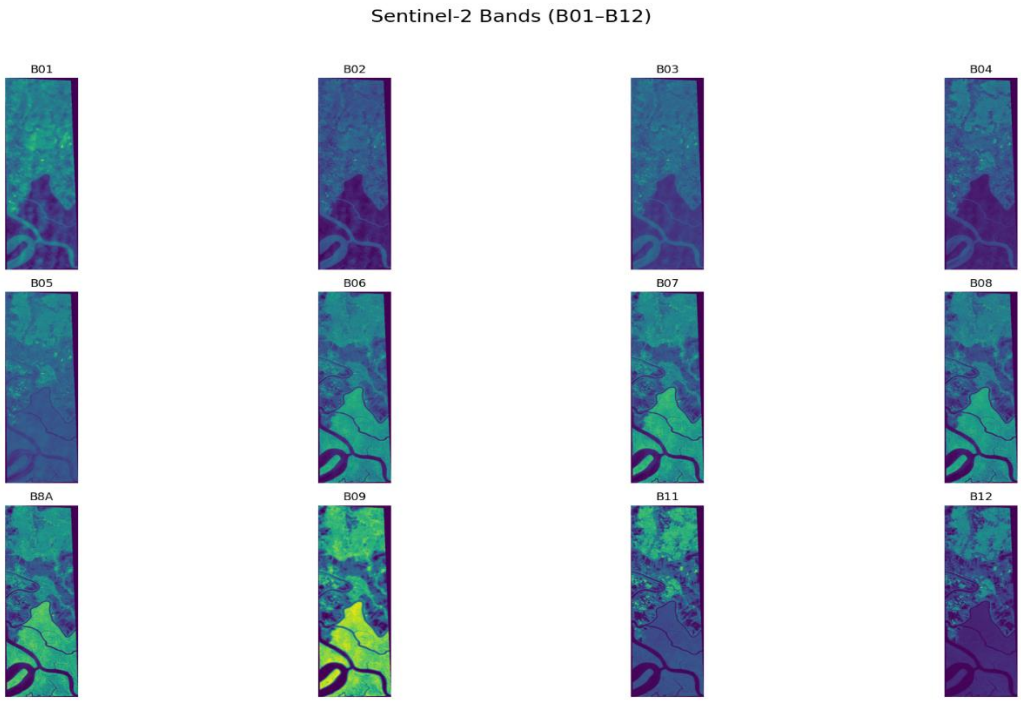
Getting hands on experience with sundarabans dataset

VISUALIZATION OF BANDS

```
!pip install rasterio matplotlib
import rasterio
import matplotlib.pyplot as plt
band_files = {
    "B01": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B01_(Raw).tiff",
    "B02": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B02_(Raw).tiff",
    "B03": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B03_(Raw).tiff",
    "B04": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B04_(Raw).tiff",
    "B05": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B05_(Raw).tiff",
    "B06": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B06_(Raw).tiff",
    "B07": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B07_(Raw).tiff",
    "B08": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B08_(Raw).tiff",
    "B8A": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B8A_(Raw).tiff",
    "B09": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B09_(Raw).tiff",
    "B11": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B11_(Raw).tiff",
    "B12": "2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B12_(Raw).tiff"
}
# Plot bands
fig, axes = plt.subplots(3, 4, figsize=(20, 12))
fig.suptitle("Sentinel-2 Bands (B01-B12)", fontsize=20)

for i, (band_name, file) in enumerate(band_files.items()):
    row = i // 4
    col = i % 4
    ax = axes[row, col]

    try:
        with rasterio.open(file) as src:
            band = src.read(1)
            ax.imshow(band, cmap='gray')
            ax.set_title(f'{band_name}')
            ax.axis('off')
    except Exception as e:
        ax.set_title(f"Error\n{band_name}")
        ax.axis('off')
        print(f"Could not load {file}: {e}")
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
```



VISUALIZATION OF BANDS WITH A COLOUR MAP



TRUE COLOUR COMPOSITE

```
#True Color Composite (B4, B3, B2)
red = load_band(get_band_path('B04'))
green = load_band(get_band_path('B03'))
blue = load_band(get_band_path('B02'))

rgb = np.stack([red, green, blue], axis=-1)
rgb /= np.max(rgb)

plt.figure(figsize=(10, 10))
plt.imshow(rgb)
plt.title("True Color Composite (B4, B3, B2)")
plt.axis('off')
plt.show()
```

NDVI AND NDWI CALCULATIONS

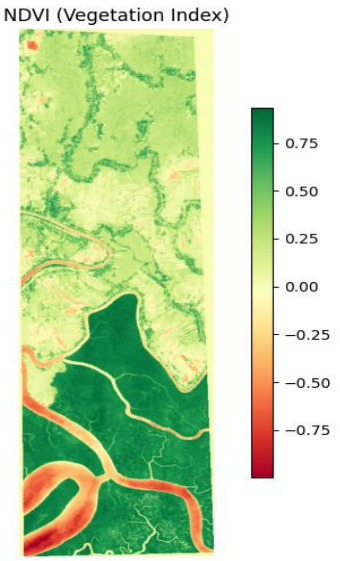
$$\text{NDVI} = \frac{B8 - B4}{B8 + B4}$$

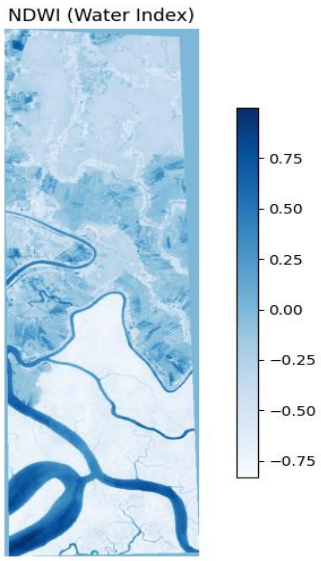
```
band3_path = '2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B03_(Raw).tiff'
band4_path = '2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B04_(Raw).tiff'
band8_path = '2020-01-27-00_00_2020-01-27-23_59_Sentinel-2_L2A_B08_(Raw).tiff'
with rasterio.open(band3_path) as b3:
    green = b3.read(1).astype('float32')
with rasterio.open(band4_path) as b4:
    red = b4.read(1).astype('float32')
```

```
    meta = b4.meta.copy()

with rasterio.open(band8_path) as b8:
    nir = b8.read(1).astype('float32')
ndvi = (nir - red) / (nir + red + 1e-5)
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.title("NDVI (Vegetation Index)")
plt.imshow(ndvi, cmap='RdYlGn')
plt.colorbar(shrink=0.7)
plt.axis('off')
```

$$\text{NDWI} = \frac{B3 - B8}{B3 + B8}$$

```
ndwi = (green - nir) / (green + nir + 1e-5)
plt.subplot(1, 2, 2)
plt.title("NDWI (Water Index)")
plt.imshow(ndwi, cmap='Blues')
plt.colorbar(shrink=0.7)
plt.axis('off')

plt.tight_layout()
plt.show()
```





NDVI near +1 → dense green vegetation

NDVI near 0 → bare soil

NDWI near +1 → open water

NDWI near 0 or negative → dry land or vegetation

Using NDVI and NDWI to classify the Sundarbans image into dense vegetation,sparse vegetation,Bare soil and water bodies

```
# NDVI thresholds
def classify_ndvi(ndvi):
    classification = np.zeros(ndvi.shape, dtype=np.uint8)

    # 0 = No vegetation
    classification[(ndvi < 0)] = 0  # Water/urban/shadows
    # 1 = Bare soil
    classification[(ndvi >= 0) & (ndvi < 0.2)] = 1
    # 2 = Sparse vegetation
    classification[(ndvi >= 0.2) & (ndvi < 0.5)] = 2
    # 3 = Dense vegetation
    classification[(ndvi >= 0.5)] = 3

    return classification

ndvi_classified = classify_ndvi(ndvi)
import matplotlib.colors as mcolors
cmap = mcolors.ListedColormap(['black', 'sandybrown', 'yellowgreen', 'forestgreen'])
bounds = [0, 1, 2, 3, 4]
norm = mcolors.BoundaryNorm(bounds, cmap.N)

plt.figure(figsize=(8, 6))
plt.title("NDVI Classified Land Cover")
plt.imshow(ndvi_classified, cmap=cmap, norm=norm)
cbar = plt.colorbar(ticks=[0.5, 1.5, 2.5, 3.5])
cbar.ax.set_yticklabels(['No Veg', 'Bare Soil', 'Sparse Veg', 'Dense Veg'])
plt.axis('off')
plt.show()
# NDWI thresholding: water > 0.3
water_mask = (ndwi > 0.3).astype(np.uint8)
plt.figure(figsize=(6, 5))
plt.title("Water Mask from NDWI")
plt.imshow(water_mask, cmap='Blues')
plt.colorbar(label='Water Presence')
```

```
plt.axis('off')
plt.show()
```