RS_Expt7

April 27, 2025

[36]:
```
import ee
import geemap
ee.Authenticate()
ee.Initialize(project='ee-nandiinii')
```

<IPython.core.display.HTML object>

[37]:
```
# Bands considered as
band_names = ee.List(['B2', 'B3','B4','B5','B6','B7','B10','B11'])
```

<IPython.core.display.HTML object>

[38]:
```
# Load a landst 8 image and select the bands of intrest
image = ee.Image('LANDSAT/LC08/C02/T1/LC08_044034_20140318').select(band_names)
```

<IPython.core.display.HTML object>

[39]:
```
# Display the input image and the region
n = geemap.Map()
n.center_object(region, 10)
n.add_layer(ee.Image().paint(region, 0, 2), {}, 'Region')
n.add_layer(
    image,
    {'bands': ['B5', 'B4', 'B2'], 'min': 0, 'max': 20000},
    'Original Image',
)
display(n)
```

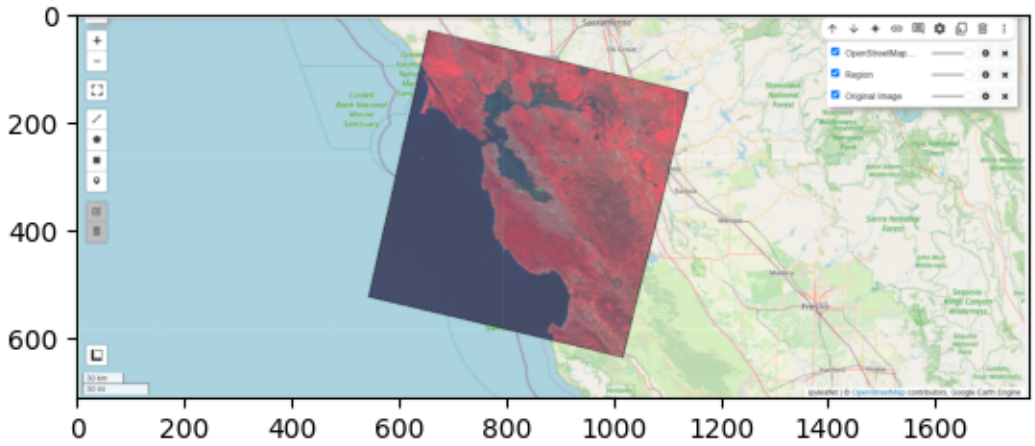
<IPython.core.display.HTML object>

```
Map(center=[37.47164678275328, -122.14450014746849],␣
 ↪controls=(WidgetControl(options=['position', 'transparent…
```

[40]:
```
import matplotlib.pyplot as plt
img=plt.imread('/content/drive/MyDrive/Colab Notebooks/Screenshot 2025-04-27␣
 ↪204509.png')
plt.imshow(img)
```

<IPython.core.display.HTML object>

[40]: <matplotlib.image.AxesImage at 0x7c3b22f1c610>



[41]:
```
# Set an appropriate scale for the landsat data
scale=30
```

<IPython.core.display.HTML object>

[42]:
```
# Mean center the data to enable faster covariance calculation and SD stretch
mean_dict = image.reduceRegion(
    reducer=ee.Reducer.mean(),
    geometry=region,
    scale=scale,
    maxPixels=1e9
)
means = mean_dict.toImage(band_names)
centered = image.subtract(means)
```

<IPython.core.display.HTML object>

[43]:
```
# This helper function returns a list of new band names
def get_new_band_names(prefix):
    seq = ee.List.sequence(1, band_names.length())
    def add_prefix_and_number(b):
        return ee.String(prefix).cat(ee.Number(b).int())
    return seq.map(add_prefix_and_number)
```

<IPython.core.display.HTML object>

[44]:
```
# This function accepts mean centered imagery, a scale, and a region
# Performs PCA and returns principal components as a new image
def get_principal_components(centered, scale, region):
```

```
    # Collapse bands into 1D array
    arrays = centered.toArray()

    # Compute the covariance of the bands within the region
    covar = arrays.reduceRegion(
        reducer=ee.Reducer.centeredCovariance(),
        geometry=region,
        scale=scale,
        maxPixels=1e9
    )

    # Get the covariance 'array' object
    covar_array = ee.Array(covar.get('array'))

    # Compute eigenvalues and eigenvectors
    eigens = covar_array.eigen()
    eigen_values = eigens.slice(1, 0, 1)
    eigen_vectors = eigens.slice(1, 1)

    # Project centered image into the space of eigenvectors
    array_image = arrays.toArray(1)
    principal_components = ee.Image(eigen_vectors).matrixMultiply(array_image)

    # Standard deviation image
    sd_image = (
        ee.Image(eigen_values.sqrt())
        .arrayProject([0])
        .arrayFlatten([get_new_band_names('sd')])
    )

    # Return principal components normalized by their standard deviations
    return (
        principal_components
        .arrayProject([0])
        .arrayFlatten([get_new_band_names('pc')])
        .divide(sd_image)
    )
```

```
<IPython.core.display.HTML object>
```

[45]:
```
# Get the principal components image
pc_image = get_principal_components(centered, scale, region)
```

```
<IPython.core.display.HTML object>
```

[46]:
```
# Add principal components layers to the map
for i in range(1, band_names.length().getInfo() + 1):
```

```
    pc_band = 'pc' + str(i)
    n.add_layer(
        pc_image.select(pc_band),
        {'min': -2, 'max': 2},
        pc_band
    )
```

```
<IPython.core.display.HTML object>
```

[47]:
```
# Display the final map
display(n)
```

```
<IPython.core.display.HTML object>
```

```
Map(center=[37.47164678275328, -122.14450014746849],␣
↪controls=(WidgetControl(options=['position', 'transparent…
```

[48]:
```
import matplotlib.pyplot as plt
img=plt.imread('/content/drive/MyDrive/Colab Notebooks/Screenshot 2025-04-27␣
↪205100.png')
plt.imshow(img)
```

```
<IPython.core.display.HTML object>
```

[48]:
```
<matplotlib.image.AxesImage at 0x7c3b22f997d0>
```