PCA Best

April 18, 2023

[ ]:
```
import rasterio
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:
```
import rasterio
import numpy as np
from sklearn.decomposition import PCA

# Input file path
file_path = "H:/Term Paper/VINIR1_stacked.tif"

# Read the image data and metadata
with rasterio.open(file_path) as src:
    img = src.read()
    meta = src.meta

# Reshape the image array to a 2D array
n_pixels = img.shape[1] * img.shape[2]
n_bands = img.shape[0]
img_2d = img.transpose(1, 2, 0).reshape(n_pixels, n_bands)

# Apply PCA to the image data
pca = PCA(n_components=n_bands)
PC = pca.fit_transform(img_2d)
```

[ ]:
```
import rasterio
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt

# Input file path
file_path = "H:/Term Paper/VINIR1_stacked.tif"

# Read the image data and metadata
with rasterio.open(file_path) as src:
    img = src.read()
    meta = src.meta
```

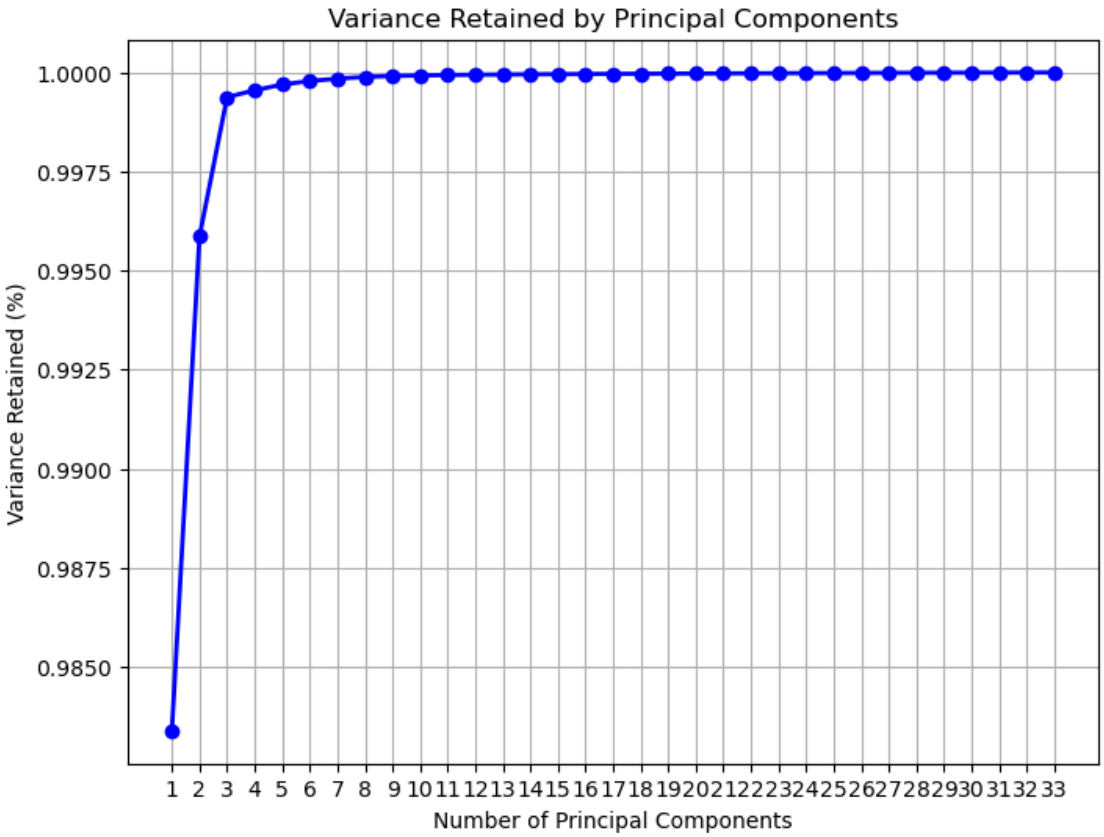
```
# Reshape the image array to a 2D array
n_pixels = img.shape[1] * img.shape[2]
n_bands = img.shape[0]
img_2d = img.transpose(1, 2, 0).reshape(n_pixels, n_bands)

# Apply PCA to the image data
pca = PCA(n_components=n_bands)
PC = pca.fit_transform(img_2d)

# Print the explained variance ratios for each principal component
print(pca.explained_variance_ratio_)
```

```
[9.83385356e-01 1.24758329e-02 3.51407477e-03 1.76412751e-04
 1.42104890e-04 9.22144741e-05 5.64793531e-05 4.65735198e-05
 2.23032357e-05 1.42569023e-05 1.01021618e-05 7.25545519e-06
 5.82354326e-06 5.20475982e-06 4.12998174e-06 3.94785745e-06
 3.69156752e-06 3.18678237e-06 2.84958329e-06 2.65582771e-06
 2.64178564e-06 2.51192455e-06 2.33935720e-06 2.18873932e-06
 2.14563906e-06 2.03137548e-06 1.93725890e-06 1.86670914e-06
 1.78804500e-06 1.70436231e-06 1.64388577e-06 1.55554447e-06
 1.18898851e-06]
```

[ ]:
```
# Plot the graph
plt.figure(figsize=(8,6))
plt.plot(range(1, n_bands+1), pca.explained_variance_ratio_.cumsum(), 'bo-',␣
 ↪linewidth=2)
plt.xlabel('Number of Principal Components')
plt.ylabel('Variance Retained (%)')
plt.title('Variance Retained by Principal Components')
plt.xticks(range(1, n_bands+1))
plt.grid()
plt.show()
```


Variance Retained by Principal Components
Variance Retained (%)
1.0000
0.9975
0.9950
0.9925
0.9900
0.9875
0.9850
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
Number of Principal Components