Pixel based_SVM

March 28, 2024

1 Project title:-

Advancing Earth Observation Data and ResUNet-Deep Learning Model for Irrigated Area Mapping: The Case of Along the Awash Valley, Ethiopia

2 Pixel Based Image Classification (PBIC) using Support Vector Machine (SVM) classifier

This Jupyter notebook demonstrates how to apply PBIC using RF classifier with the ESA EO-Africa inovation lab cloud computing environment.

Prerequisites for running this notebook

Several packages need to be installed and/or imported for running this script:

The `rasterio, geopandas,sklearn,and numpy` modules should be installed first to apply PBIC based RF classifier ;

2.0.1 Import Relevant Packages

[1]:

```
import rasterio
import numpy as np
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import geopandas as gpd
```

2.0.2 Load Sentinel-2 image

[ ]:

```
sentinel_image_path = "/home/eoafrica/Sentinel2_AWbasin/outputs_rgb/stacked_rgb.tif"
with rasterio.open(sentinel_image_path) as src:
    sentinel_image = src.read()
```

2.0.3 Load ground truth GCP shapefile data

[ ]:

```
gcp_shapefile_path = "/home/eoafrica/Sentinel2_AWbasin/GCP_LULCawash/lulcgcp.shp"
gcp_data = gpd.read_file(gcp_shapefile_path)
```

2.0.4 Extract the values of the image pixels at the locations of the GCPs

[ ]:

```
gcp_points = gcp_data.geometry.apply(lambda geom: (geom.x, geom.y)).tolist()
gcp_values = []
for point in gcp_points:
    row, col = src.index(point[0], point[1])
    gcp_values.append(sentinel_image[:, row, col])

gcp_values = np.array(gcp_values)
```

2.0.5 Extract corresponding class labels from the GCP data

[ ]:

```
class_labels = gcp_data["class_labe"].values.astype(int)
```

2.0.6 Split data into training and testing sets

[ ]:

```
X_train, X_test, y_train, y_test = train_test_split(gcp_values, class_labels, test_size=0.3, random_state=42)
```

2.0.7 Train Support Vector Classifier

[ ]:

```
clf = SVC(kernel='linear', C=1.0)
clf.fit(X_train, y_train)
```

2.0.8 Predict on the test set

[ ]:

```
y_pred = clf.predict(X_test)
```

2.0.9 Calculate accuracy

[ ]:

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

2.0.10 Classify the whole image

[ ]:

```
predicted_image = np.zeros_like(sentinel_image[0])
for i in range(sentinel_image.shape[1]):
    for j in range(sentinel_image.shape[2]):
        pixel_values = sentinel_image[:, i, j].reshape(1, -1)
        predicted_class = clf.predict(pixel_values)
        predicted_image[i, j] = predicted_class
```

2.0.11 Save the classified image

[ ]:
```
classified_image_path = "/home/eoafrica/Sentinel2_AWbasin/RF_LULCresult/
 ↪classified_image.tif"  # Path to save the classified image
with rasterio.open(classified_image_path, 'w', **src.profile) as dst:
    dst.write(classified_image.astype(rasterio.uint8))
```

[ ]:
```

```