```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
These tools are accessing Google Earth Engine API
Author: Alexandra Christensen
Created: November 24, 2022
Updated: 12/01/2022

"""

import fnmatch
import os
import numpy as np
import rasterio
from shapely.geometry import Polygon, LineString
import geopandas as gpd
import math
import ee
from ee import batch


def maskS2clouds(image):
    qa = image.select('QA60');
    ##Bits 10 and 11 are clouds and cirrus, respectively.
    cloudBitMask = 1 << 10
    cirrusBitMask = 1 << 11
    ##Both flags should be set to zero, indicating clear conditions.
    cloudmask =
qa.bitwiseAnd(cloudBitMask).eq(0).And(qa.bitwiseAnd(cirrusBitMask).eq(0))
    #cloudmask =
qa.bitwiseAnd(cloudBitMask).eq(0)*(qa.bitwiseAnd(cirrusBitMask).eq(0))
    return image.updateMask(cloudmask)#.divide(10000)
    #return
image.updateMask(qa.bitwiseAnd(cloudBitMask).eq(0)).updateMask(qa.bitwise
And(cirrusBitMask).eq(0)).divide(100000)
def addNDWI(image):
  ndwi = image.normalizedDifference(['B8',
'B3']).rename('NDWI').toFloat();
  return image.addBands(ndwi)
def addNDVI(image):
  #ndvi = image.normalizedDifference(['B8',
'B3']).rename('NDVI').toFloat();
  ndvi =
(image.select(['B8']).subtract(image.select(['B3']))).divide(image.select
(['B8']).add(image.select(['B3']))).rename('NDVI')
  return image.addBands(ndvi)
def addMNDWI(image):
  mndwi = image.normalizedDifference(['B3', 'B11']).rename('MNDWI');
  return image.addBands(mndwi)


def get_S2_collection(y0,y1,m0,m1,clouds,polygon):
    collectionS2 = ee.ImageCollection("COPERNICUS/S2")\
        .filter(ee.Filter.calendarRange(y0,y1,'year'))\
        .filter(ee.Filter.calendarRange(m0,m1,'month'))\
        .filterBounds(polygon)\
        .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', clouds))\
        .map(addNDWI)\
        .map(addMNDWI)\
        .map(addNDVI)\
        .map(maskS2clouds)
    return collectionS2

def get_S1_collection(y0,y1,m0,m1,polygon):
    collectionVV = ee.ImageCollection('COPERNICUS/S1_GRD')\
        .filter(ee.Filter.eq('instrumentMode', 'IW'))\
        .filter(ee.Filter.calendarRange(y0,y1,'year'))\
        .filter(ee.Filter.calendarRange(m0,m1,'month'))\
        .filter(ee.Filter.listContains('transmitterReceiverPolarisation',
'VV'))\
        .filterMetadata('resolution_meters', 'equals' , 10)\
        .filterBounds(polygon)\
        .select('VV');
    collectionVH = ee.ImageCollection('COPERNICUS/S1_GRD')\
        .filter(ee.Filter.eq('instrumentMode', 'IW'))\
        .filter(ee.Filter.calendarRange(y0,y1,'year'))\
        .filter(ee.Filter.calendarRange(m0,m1,'month'))\
        .filter(ee.Filter.listContains('transmitterReceiverPolarisation',
'VH'))\
        .filterMetadata('resolution_meters', 'equals' , 10)\
        .filterBounds(polygon)\
        .select('VH');
    return collectionVV, collectionVH


def
submit_a_GEE_task(bands,ref_composite,method,polygon,AOI,resolution,save_
comp=False,save_mask=True):

    #bands = ['NDWI','VV','VH'];
    #ref_composite =
ee.Image.cat(ndwi_max.toFloat(),VV_ref.toFloat(),VH_ref.toFloat()).select
(bands).clip(ee_poly);
    #method = 'ndwimax_vvvhmax'
    print('Method is %s' %(method))
    folder_on_drive = 'gee_' + AOI
    if save_comp:
        # Save composite
        comp_out1 = batch.Export.image.toDrive(ref_composite, folder =
folder_on_drive, description='%s_%s' %(AOI,method), fileFormat =
'GeoTiff',scale = resolution,maxPixels=10000000000000 )
        process = batch.Task.start(comp_out1)
        print('Saved composite to Google Drive at %s/%s_%s.tif'
%(folder_on_drive,AOI,method))

    ## Unsupervised Clustering using S2 max, VV, and VH composite
    training =
ref_composite.sample(region=polygon,scale=resolution,numPixels=500,tileSc
ale = 16)
    classifier = ee.Clusterer.wekaKMeans(2).train(training,bands)
    water = ref_composite.select(bands).cluster(classifier);

    if save_mask:
        ## Save clusters
        mask_out1 = batch.Export.image.toDrive(water,
region=polygon,folder = folder_on_drive, description='%s_%s_clustered'
%(AOI,method),fileFormat = 'GeoTiff',scale
=resolution,maxPixels=10000000000000 )
        process = batch.Task.start(mask_out1)
        print('Saved clusters to Google Drive at %s/%s_%s_clustered.tif'
%(folder_on_drive,AOI,method))
    print('')
```