



Chula
Chulalongkorn University

Open Remote Sensing II

Lab : Introduction to Google Earth Engine (again ?)

2108-421 Modern Integrated Surveying Technology
Semester 2/2022

Thepchai Srinoi

(thepchairsrinoi@gmail.com)

Department of Survey Engineering Chulalongkorn University

Bangkok Thailand



Google Earth Engine with Java programming

<https://code.earthengine.google.com/>



เขียน code ข้างบน แสดงภาพข้างล่าง

```
1 var AOI =
2   ee.Geometry.Polygon(
3     [[[101.1, 12.7],
4       [101.1, 12.6],
5       [101.2, 12.6],
6       [101.2, 12.7]]], null, false);
7
8 // Download Image
9 function maskS2clouds(image) {
10   var qa = image.select('QA60');
11
12   // Bits 10 and 11 are clouds and cirrus, respectively.
13   var cloudBitMask = 1 << 10;
14   var cirrusBitMask = 1 << 11;
15
16   // Both flags should be set to zero, indicating clear conditions.
17   var mask = qa.bitwiseAnd(cloudBitMask).eq(0)
18     .and(qa.bitwiseAnd(cirrusBitMask).eq(0));
19
20   return image.updateMask(mask).divide(10000);
21 }
22
23 var image = ee.ImageCollection('COPERNICUS/S2_SR')
24   .filterDate('2020-12-21', '2020-12-24')
25   .filterBounds(AOI)
26   .map(maskS2clouds)
27   .mosaic()
28 // Image Visualization
29 Map.centerObject(AOI, 10)
30 Map.addLayer(image, {min: 0.0,max: 0.2,bands: ['B4', 'B3', 'B2']}, 'S2A');
```

```
# Install Function for Cloud Mask
```

```
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.
```

```
    mask = (qa.bitwiseAnd(cloudBitMask).eq(0)) and (qa.bitwiseAnd(cirrusBitMask).eq(0))
```

```
    return image.updateMask(mask).divide(10000)
```

```
# Map Visualization : Google Earth Engine
```

```
def add_ee_layer(self, ee_image_object, vis_params, name):
```

```
    map_id_dict = ee.Image(ee_image_object).getMapId(vis_params)
```

```
    folium.raster_layers.TileLayer(
```

```
        tiles=map_id_dict['tile_fetcher'].url_format,
```

```
        attr='Map Data &copy; <a href="https://earthengine.google.com/">Google Earth Engine</a>',
```

```
        name=name,
```

```
        overlay=True,
```

```
        control=True
```

```
    ).add_to(self)
```

```
[ ] # Open the image
```

```
    lat_min, lat_max, lon_min, lon_max =(12.6, 12.7, 101.1, 101.2)
```

```
    AOI = ee.Geometry.Polygon(
```

```
        [[[lon_min, lat_max],
```

```
            [lon_min, lat_min],
```

```
            [lon_max, lat_min],
```

```
            [lon_max, lat_max]]])
```

```
    fromdate = '2020-12-21'
```

```
    todate = '2020-12-24'
```

```
    image = ee.ImageCollection('COPERNICUS/S2_SR').filterDate(fromdate,todate).\
```

```
        filterBounds(AOI).map(maskS2clouds).mosaic()
```

```
    # Visualization
```

```
    folium.Map.add_ee_layer = add_ee_layer
```

```
    c = (AOI.centroid().getInfo())['coordinates']
```

```
    map_matched = folium.Map(location=[c[1], c[0]], zoom_start=13)
```

```
    map_matched.add_ee_layer(image,{'min':0, 'max':0.2, 'bands':['B4','B3','B2']}, 'RGB_L2A')
```

```
    display(map_matched.add_child(folium.LayerControl()))
```

สำหรับ python programming สามารถทำได้ผ่าน Google Colab

<https://colab.research.google.com/>



แสดงแผนที่ด้วย folium

Welcome to Colab Laboratory for Google Earth Engine

login เข้า google earth engine

```
import ee
# Authenticate to the Earth Engine servers
ee.Authenticate()
# Initialize the API
ee.Initialize()
```

... To authorize access needed by Earth Engine, open the following URL in a web browser and follow the instructions. If the web browser does not start automatically, please manually
https://code.earthengine.google.com/client-auth?scopes=https%3A//www.googleapis.com/auth/earthengine%20https%3A//www.googleapis.com/auth/devstorage.full_control&request_id=Mk

The authorization workflow will generate a code, which you should paste in the box below.
Enter verification code:

login เข้า google earth engine

```
import ee
# Authenticate to the Earth Engine servers
ee.Authenticate()
# Initialize the API
ee.Initialize()
```

To authorize access needed by Earth Engine, open the following URL in a web browser and follow the instructions. If the web browser does not start
<https://code.earthengine.google.com/client-auth?scopes=https%3A//www.googleapis.com/auth/earthengine%20https%3A//www.googleapis.com/auth/devsto>

The authorization workflow will generate a code, which you should paste in the box below.
Enter verification code: 4/1AwTgzH5M4Mf0fLNRp1FzIYCZGJ_OgvUA05T4YfAJdWPTbFC9wKH7k_sQk1Y

Successfully saved authorization token.

Google Earth Engine

Notebook Authenticator

Active account: ThepchaiSrinoi@gmail.com SWITCH ACCOUNT

Cloud Project: seniorproject2022 CHOOSE PROJECT

Data access: ☐ Use read-only scopes

WARNING: ONLY PROCEED IF YOU NEED TO ACCESS EARTH ENGINE FROM A NOTEBOOK

The token that you generate here will allow access to your Google account. Ensure that you understand the notebook that you are running.

Any code that you include in the notebook (and anyone with access to the notebook kernel) will be able to copy or **change** your data. Enable read-only scopes above to prevent data changes.

If you are not running a notebook, or you don't understand these warnings, then the link that sent you here may be trying to trick you. Do not proceed!

GENERATE TOKEN

Earth Engine Notebook Client -
ThepchaiSrinoi@gmail.com

ต้องการเข้าถึงบัญชี Google ของคุณ

thepchaisrinoi@gmail.com

เลือกสิ่งที่

Earth Engine Notebook Client -
ThepchaiSrinoi@gmail.com
เข้าถึงได้

- ☒ ดูและจัดการข้อมูลใน Google Earth Engine ดูข้อมูลเพิ่มเติม
- ☒ Manage your data and permissions in Cloud Storage and see the email address for your Google Account ดูข้อมูลเพิ่มเติม

ตรวจสอบว่าคุณเชื่อถือ Earth Engine Notebook Client - ThepchaiSrinoi@gmail.com ได้

คุณอาจกำลังแชร์ข้อมูลที่มีความละเอียดอ่อนกับเว็บเบราว์เซอร์หรือแอปนี้ คุณดูหรือยกเลิกสิทธิ์เข้าถึงได้ทุกเมื่อในบัญชี Google

ดูวิธีที่ Google ช่วยคุณแชร์ข้อมูลอย่างปลอดภัย

ดูนโยบายความเป็นส่วนตัวและข้อกำหนดในการให้บริการของ Earth Engine Notebook Client - ThepchaiSrinoi@gmail.com

รหัสการให้สิทธิ์

โปรดคัดลอกโค้ดนี้ไว้ สลับไปที่แอปพลิเคชันของคุณแล้ววางโค้ดลงไป:

4/1AwTgzH5M4Mf0fLNRp1FzIYCZGJ_OgvUA05T4YfAJdWPTbFC9wKH7k_sQk1Y

อย่าลงชื่อเข้าใช้หรือให้สิทธิ์เข้าถึงแก่ Earth Engine Notebook Client - ThepchaiSrinoi@gmail.com


```
for month in range(1,13) :
    print(month)
    fromdate = '2021-'+str(month)+'-01'
    todate = '2021-'+str(month)+'-28'
    if month == 1 :
        myimage2 = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_NO2')\
            .select('NO2_column_number_density')\
            .filterDate(fromdate,todate)\
            .filterBounds(AOI)\
            .mean()\
            .rename('NO2_'+str(month))

    else :
        image = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_NO2')\
            .select('NO2_column_number_density')\
            .filterDate(fromdate,todate)\
            .filterBounds(AOI)\
            .mean()\

        myimage2 = myimage2.addBands(image.rename('NO2_'+str(month)))

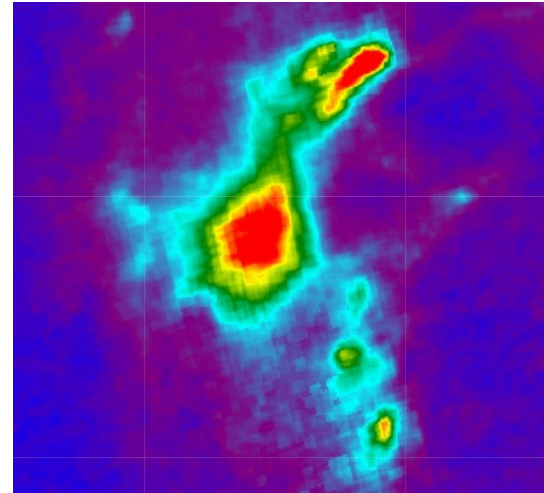
band_viz = {
    'min': 0,
    'max': 0.0002,
    'palette': ['black', 'blue', 'purple', 'cyan', 'green', 'yellow', 'red']
}

folium.Map.add_ee_layer = add_ee_layer

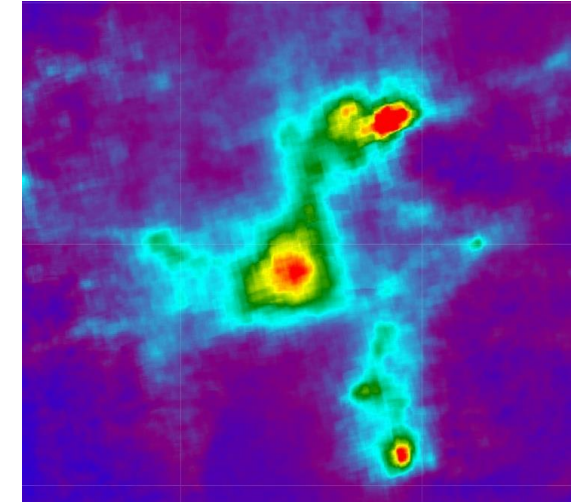
c = (AOI.centroid().getInfo())['coordinates']

map_matched = folium.Map(location=[14,100], zoom_start=9)
for month in range(1,13) :
    map_matched.add_ee_layer(myimage2.select('NO2_'+str(month)),band_viz,'NO2_'+str(month))
display(map_matched.add_child(folium.LayerControl()))
```

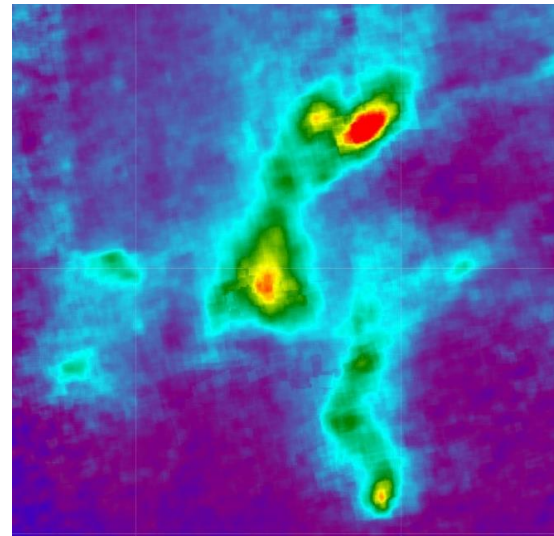
January



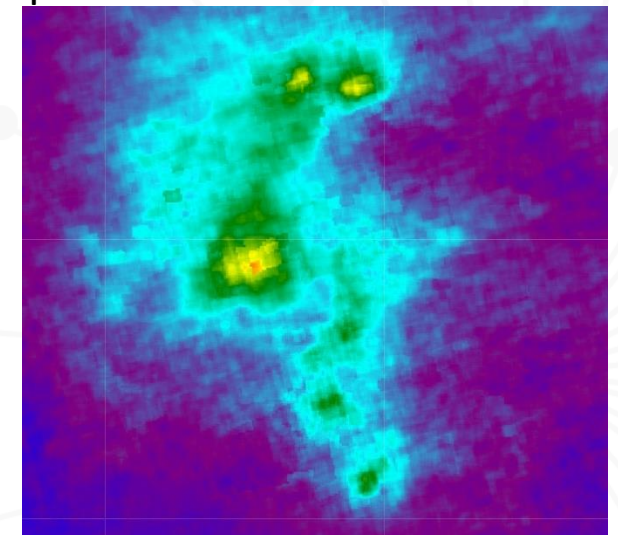
February



March



April



Import External Data : Landcover Classification with ML in GEE

df

	X	Y	class
0	100.533771	13.924564	0
1	100.516601	13.935989	0
2	100.511406	13.930158	0
3	100.512136	13.922949	0
4	100.542498	13.934334	0
...
175	100.439966	13.932960	2
176	100.435115	13.940927	2
177	100.415455	13.947197	2
178	100.429210	13.947346	2
179	100.427057	13.967969	2

180 rows × 3 columns

```
!pip install geemap
```

```
from sklearn.model_selection import train_test_split  
X = df[df.columns[:2]]  
Y = df[df.columns[2]]
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, stratify=Y, test_size=0.50, random_state = 25)  
df_train = X_train.join(Y_train)
```

#Convert from pandas to featurecollection

```
train_points = geemap.pandas_to_ee(df_train, latitude='Y', longitude='X')
```

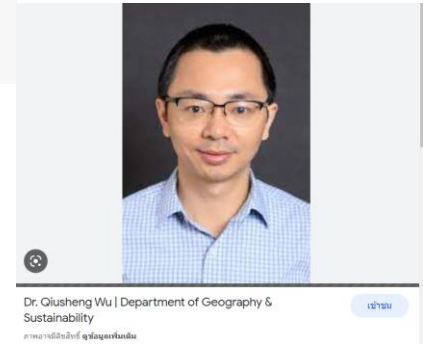
```
#Choose bands  
bands = ['B2', 'B3', 'B4', 'B12']
```

```
# Overlay the points on the imagery to get training.  
training = myimage.sampleRegions(  
    collection=train_points,  
    properties=['class'],  
    scale=30)
```

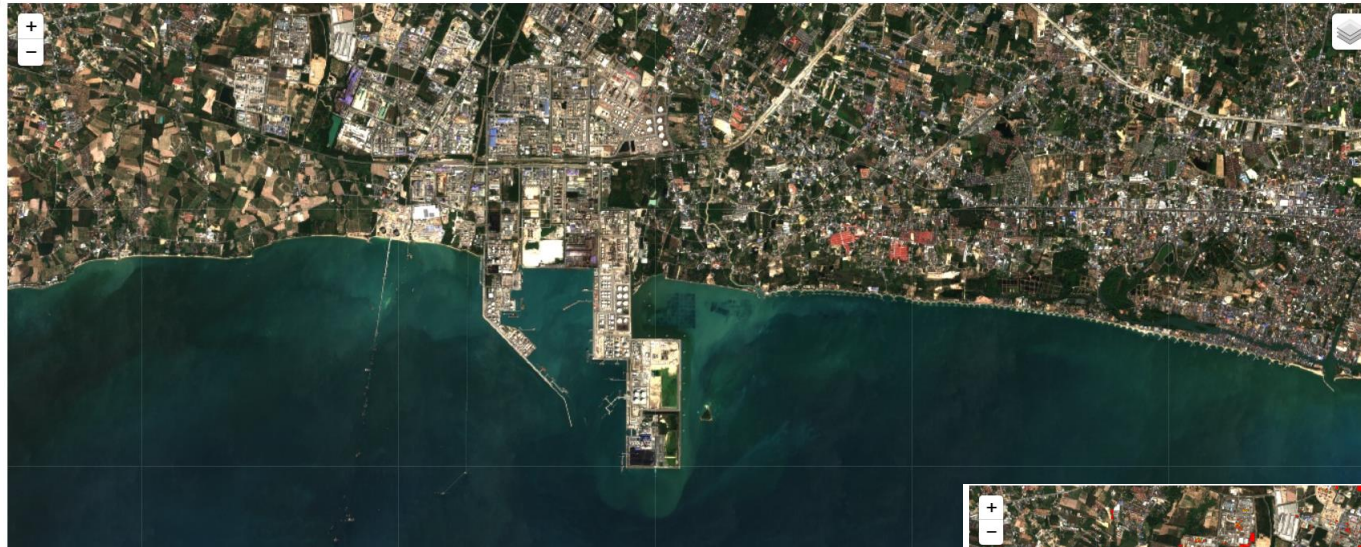
```
# Train a CART classifier with default parameters.  
trained = ee.Classifier.smileCart().train(training, 'class', bands)
```

```
# Classify the image with the same bands used for training.  
classified = (myimage.select(bands)).classify(trained)
```

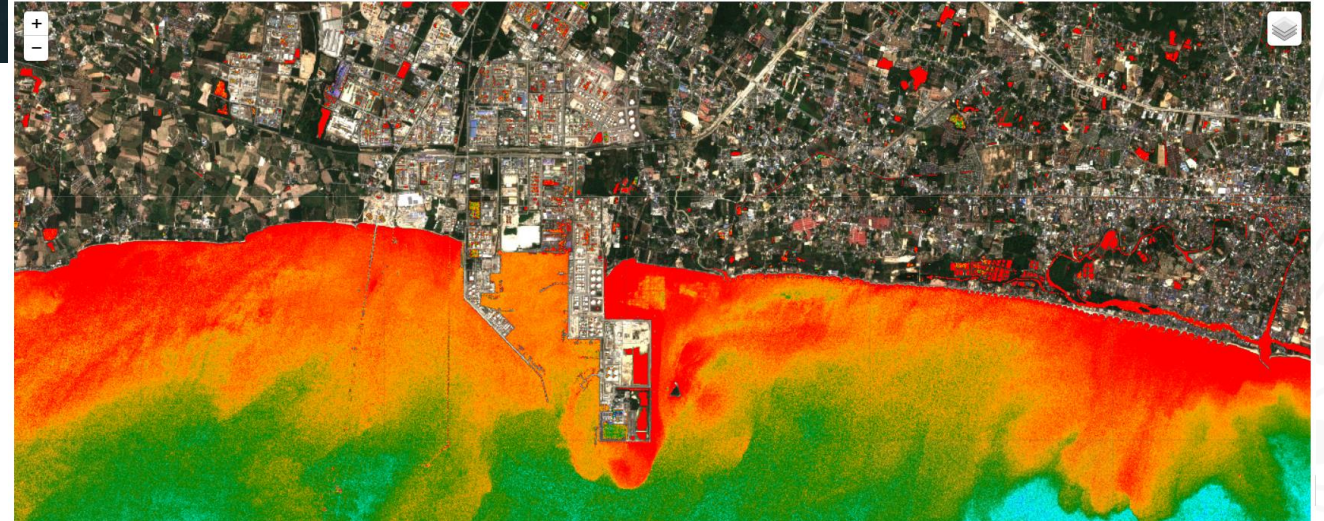
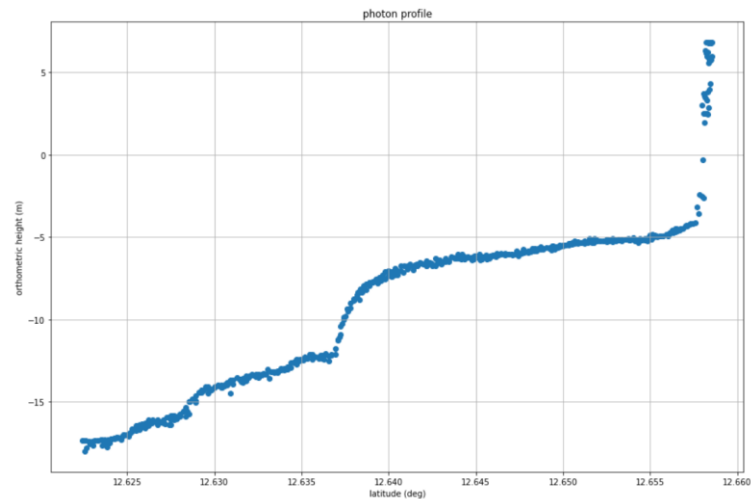
Coding คล้าย javascript version ...



Work With Big Data and ML : Satellite Derived Bathymetry Demonstration



lookmeebear Add files via upload		7abf3de 37 minutes ago 22 commits
Exercise_GEE_MIST2023.pdf	Optional Exercise	4 days ago
MIST_LAB_Lookmee_OpenRS_180120...	LAB Example 18012023	last week
MIST_Lookmee_Lab2_SDB.ipynb	Created using Colaboratory	3 hours ago
MIST_Lookmee_OpenRS_18012023.pdf	LectureSlide_18012023	last week
MIST_Lookmee_SDB_25012023.pdf	Lecture Slide 25 01 2023	1 hour ago
MIST_OpenRS_Lab1.js	Create MIST_OpenRS_Lab1.js	5 days ago
README.md	Update README.md	3 hours ago
dataset1.csv	Sample Data Bathymetry	5 days ago
introGEE_2021.pdf	Intro to GEE (GISTDA Internship)	last week
introSDB_2022.pdf	Add files via upload	37 minutes ago





Chula
Chulalongkorn University

Thepchai Srinoi

(thepchairsrinoi@gmail.com)

Department of Survey Engineering Chulalongkorn University

Bangkok Thailand

