GOOGLE EARTH ENGINE FOR MULTISPECTRAL REMOTE SENSING

เทพชัย ศรีน้อย

นิสิตชั้นปีที่ 4 ภาควิชาวิศวกรรมสำรวจ

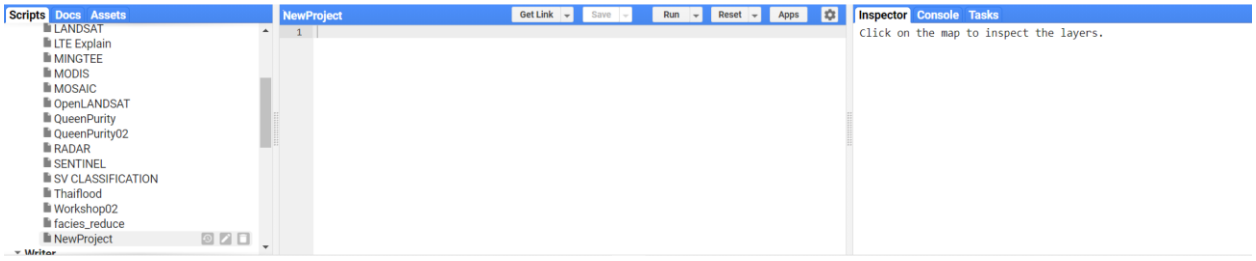
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

เอกสาร Google Earth Engine for Remote Sensing นี้คัดลอกมาจากรายงานการฝึกงานทางวิศวกรรมของผู้เขียนกับทางฝ่ายผลิตชั้นข้อมูลภูมิสารสนเทศ สำนักผลิตภัณฑ์ภูมิสารสนเทศ สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (องค์การมหาชน) เมื่อกลางปี พ.ศ. 2564 ที่ผ่านมา

เครื่องมือ Google Earth Engine (GEE) เป็นระบบการประมวลผลและแสดงผลข้อมูลทางภูมิสารสนเทศตามเงื่อนไขที่เรากำหนด ทั้งเรื่องของประเภทดาวเทียม วันที่ทำการบันทึกข้อมูล ขอบเขตพื้นที่ที่ต้องการ แบนด์ของดาวเทียม และเงื่อนไขอื่นๆ ซึ่งเกี่ยวข้องกับลักษณะทางเรขาคณิตหรือเชิงรังสีของภาพถ่าย เช่น เรื่องของการปกคลุมของเมฆ ความถูกต้องทางตำแหน่ง ระยะทางจากดวงอาทิตย์ เป็นต้น นำข้อมูลที่ค้นมาแสดงผลเป็นชั้นข้อมูลแผนที่ และสามารถนำมาประมวลผลเพื่อสร้างเป็นสารสนเทศทางภูมิศาสตร์ เพื่อเข้าใจในปัญหา ช่วยในการตัดสินใจแก้ไขปัญหาต่อไป ด้วยการเขียนโปรแกรมผ่านระบบอินเทอร์เน็ต

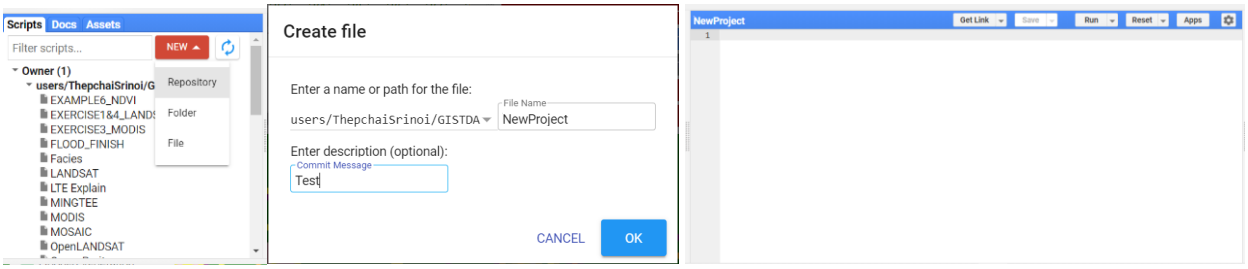
2.1 การทำความรู้จักหน้าต่าง GEE และการเริ่มต้นสร้างไฟล์งานใหม่

หน้าต่าง GEE จะพบส่วนซ้ายสุดโปรแกรม มี Tab ชื่อ Scripts สำหรับค้นไฟล์งาน Docs สำหรับการค้นฟังก์ชันเขียนโปรแกรม และ Assets สำหรับการค้นไฟล์ที่นำเข้าจากคอมพิวเตอร์ ส่วนกลางคือไฟล์งานปัจจุบันของเรา เป็นพื้นที่เขียนโปรแกรม และส่วนขวาของโปรแกรมมี Tab ชื่อ Inspector ใช้ในการแสดงข้อมูลภายในชั้นแผนที่เมื่อกดลงไปในชั้นข้อมูล Console แสดงข้อความ (หากมีการเขียนด้วย print (...)) และแสดงข้อความแจ้งความผิดพลาดของโปรแกรม และ Tasks เป็นส่วนแสดง process การนำเข้าและการส่งออกข้อมูล



รูปที่ 1 ลักษณะหน้าต่างและแถบหลักของ Google Earth Engine

เริ่มต้นงานด้วยการสร้าง Script ใหม่ด้วยการกด Tab ชื่อ Scripts ที่แถบซ้ายบน แล้วกดปุ่ม NEW ซึ่งอยู่ข้างกับ Filter Scripts แล้วเลือกที่ File จากนั้นทำการ Create file ด้วยการพิมพ์ชื่อไฟล์ที่ช่องว่าง ณ File Name แล้วกด OK จะได้ไฟล์เปล่าขึ้นมา ตรงกลางหน้าจอ GEE สามารถดำเนินการเขียนโปรแกรมได้เลย

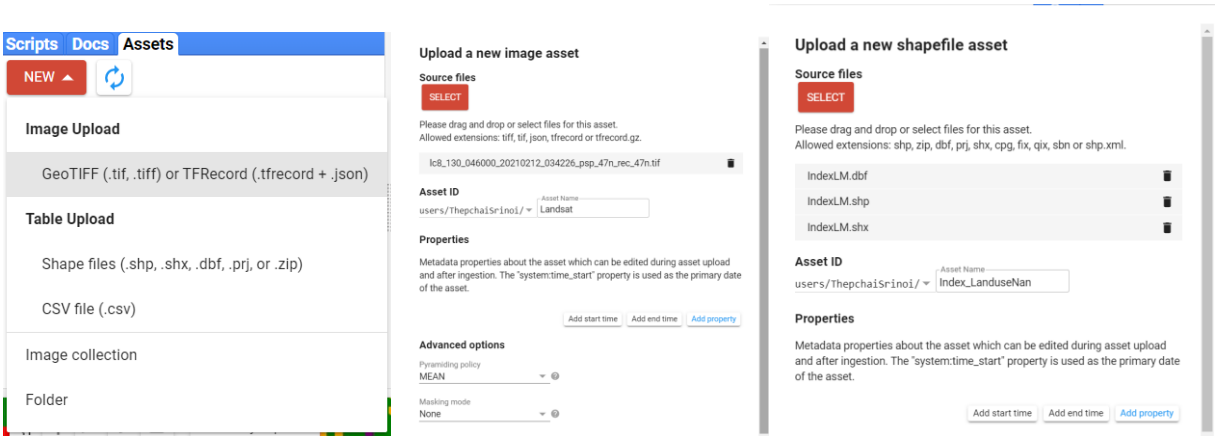


รูปที่ 2 ลำดับการสร้างไฟล์งานใหม่ของ Google Earth Engine

2.2 การนำเข้าข้อมูล

วิธีการนำเข้าข้อมูลทางด้านภูมิสารสนเทศ สามารถทำได้ทั้งการนำเข้าจากคอมพิวเตอร์ของตนเอง และการเรียกใช้ผ่าน Google Earth Engine ด้วยการเขียนโปรแกรม

2.2.1 การนำเข้าจากคอมพิวเตอร์ หากมีไฟล์ข้อมูลในคอมพิวเตอร์ สามารถนำเข้าระบบ GEE ได้ด้วยการนำเข้าผ่าน Assets ด้วยการกดที่ Tab ชื่อ Assets จากนั้นกดปุ่ม NEW แล้วจะมีตัวเลือกว่านำข้อมูลชนิดใดเข้ามายังระบบ



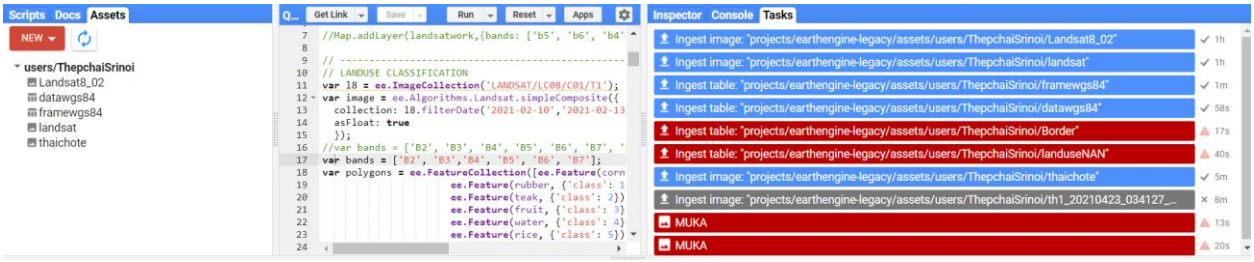
รูปที่ 3 ลำดับการนำเข้าไฟล์ภาพ GeoTiFF และไฟล์ Shapefile สำหรับ Google Earth Engine

ในกรณีนี้ยกตัวอย่างการนำเข้าข้อมูลเฉพาะ 2 รูปแบบ ตามรายละเอียดดังต่อไปนี้

- ข้อมูลราสเตอร์ ทั้งภาพถ่ายทางดาวเทียม หรือแบบจำลองระดับ บันทึกลงไฟล์ Geo TIFF เลือก Image Upload --> Geo TIFF แล้วกดปุ่ม SELECT เพื่อเลือกไฟล์จากในคอมพิวเตอร์ ทำการเปลี่ยนชื่อไฟล์เมื่อนำเข้าเรียบร้อยแล้วที่ Asset Name แล้วเลื่อนลงด้านล่างกด Upload

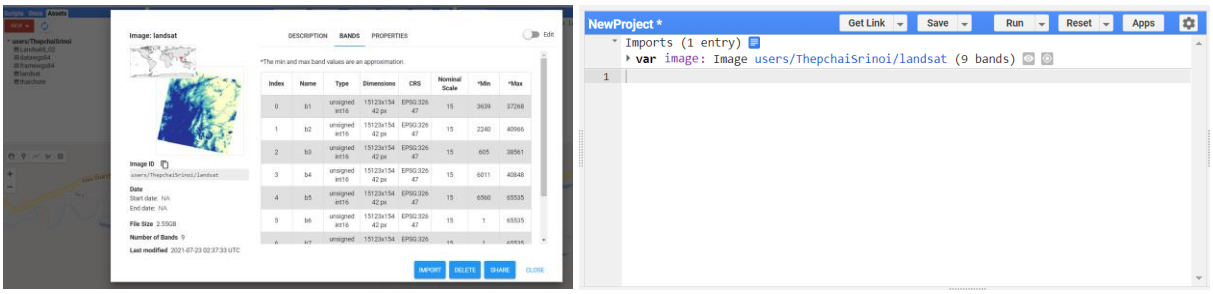
- ข้อมูลเวกเตอร์ เลือก Table Upload --> Shapefiles แล้วกดปุ่ม SELECT เพื่อเลือกไฟล์จากในคอมพิวเตอร์ โดยต้องนำเข้าไฟล์นามสกุล .shp .shx และ .dbf แล้วเลือกระบบพิกัด EPSG:4326 จากนั้นเปลี่ยนชื่อไฟล์ เมื่อนำเข้าเรียบร้อยแล้วที่ Asset Name แล้วเลื่อนลงด้านล่างกด Upload

ติดตามการดำเนินการ Upload ไฟล์เข้าระบบได้จากการกด Tasks ที่มุมขวาบนของระบบ ระยะเวลาขึ้นกับขนาดของข้อมูล เมื่อทำสำเร็จแล้วให้กด Refresh ที่วางทางขวาของปุ่ม NEW ตอนที่จะนำเข้าข้อมูล ไฟล์ที่แสดงด้วยชื่อ Asset Name ที่ตั้งไว้จะปรากฏ หากไม่พบปัญหาขัดข้องใด ๆ แล้วดังแสดงในรูปที่ 4



รูปที่ 4 ลักษณะของหน้า Assets (ทางซ้าย) และ Tasks (ทางขวา) เมื่อนำไฟล์เข้าสำเร็จและไม่สำเร็จ

ตอนนี้สามารถนำข้อมูลของเราเข้าไฟล์งานได้ด้วยการเลือกไฟล์ใน Assets แล้ว Import เข้ามาในไฟล์งาน ขณะนี้ ข้อมูลของเราจะถูกเก็บเป็นตัวแปรหนึ่ง สามารถเปลี่ยนชื่อตัวแปรได้ เพื่อนำไปเขียนโปรแกรมต่อไป



รูปที่ 5 การนำไฟล์จาก Asset ทำการ Import เข้ามาในไฟล์งาน

ตัวอย่าง code เก็บภาพที่นำเข้ามาจากคอมพิวเตอร์

```
var image = ee.Image("users/ThepchaiSrinoi/landsat")
```
[1]

2.2.2 การนำเข้าด้วยการเขียนโปรแกรม Google Earth Engine ได้เก็บข้อมูลทางภูมิศาสตร์ไว้แล้วโดยเฉพาะภาพถ่ายทางดาวเทียมไว้จำนวนมหาศาล ทั้งดาวเทียม Landsat 1-8, Sentinel 1-5 และ MODIS นอกจากนี้ยังมีแบบจำลองระดับ DEM SRTM ข้อมูลเขตการปกครอง Large Scale International Boundary (LSIB) และข้อมูลอื่นๆ ซึ่งการค้นภาพตามชื่อดาวเทียมทำผ่านฟังก์ชัน

ee.ImageCollection(PATH ของดาวเทียม)

ตัวอย่างเช่น

สำหรับ ชุดภาพจากดาวเทียม Landsat-8

```
var landsat8 = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
```
[2]

สำหรับ ชุดภาพจากดาวเทียม Sentinel-2

```
var sentinel2 = ee.ImageCollection('COPERNICUS/S2')
```
[3]

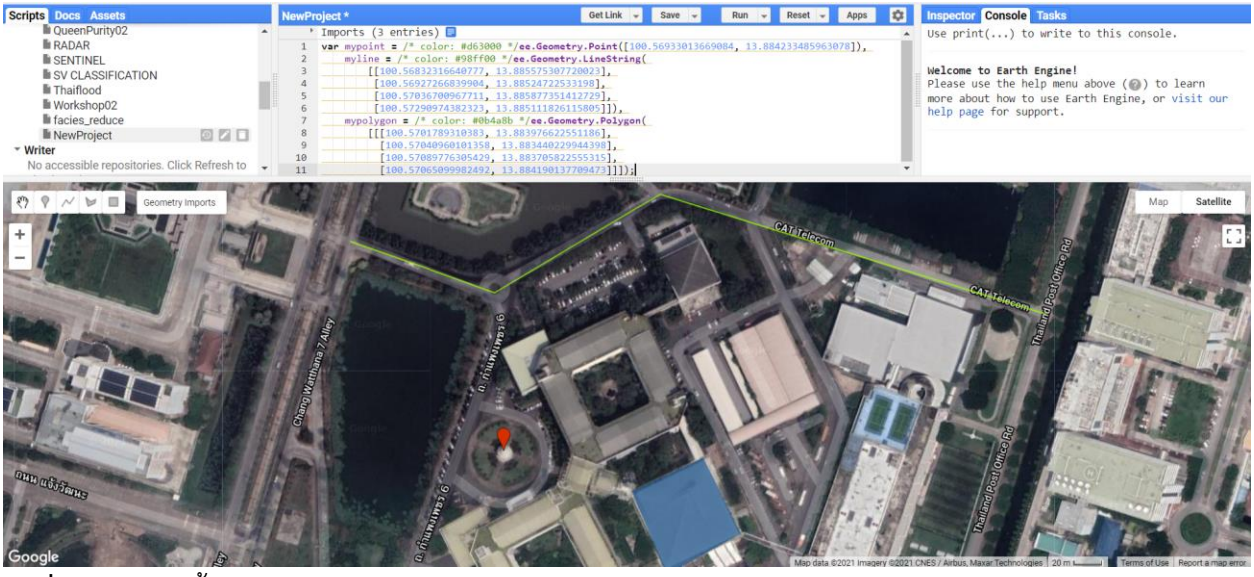
การนำเข้าในตอนแรกจะเป็นชุดของภาพจากดาวเทียมที่เรียกค้นด้วยการเขียนโปรแกรม จึงต้องจำเป็นต้องทำการกรองด้วยเงื่อนไขที่กำหนดเพื่อแสดงผลหรือวิเคราะห์ต่อจากนี้ไป

ตัวอย่างการกรองจากชุดภาพที่เป็น Multispectral Remote Sensing เช่น landsat-8 หรือ Sentinel-2 แต่ถ้าเป็นชุดภาพจาก Microwave Remote Sensing เช่น Sentinel-1 จะมีรูปแบบการกรองภาพอีกแบบหนึ่งตามคุณลักษณะของการบันทึกข้อมูล จากนี้จะยกตัวอย่างสำหรับชุดภาพจาก landsat-8 ดังนี้

จากการค้นภาพด้วยชื่อดาวเทียมใน code [2] เก็บในตัวแปร landsat-8 จากนั้นทำการค้นตามวันที่ด้วย filter Date (start,end) สมมุติว่าอยากได้ภาพที่ถ่ายในช่วงวันที่ 1 มกราคม 2564 จนถึง 30 มกราคม 2564 เก็บชุดภาพที่สอดคล้องในตัวแปร landsat8

```
var landsat8 = landsat8.filterDate('2021-01-01','2021-01-30')
```
[4]

เมื่อค้นภาพในช่วงเวลาที่กำหนดใน code [4] หากเราต้องการเลือกภาพที่ครอบคลุมบริเวณที่เราสนใจสร้างรูปเรขาคณิตผ่านพื้นที่ แล้วทำการเก็บลงในตัวแปร



รูปที่ 5 การสร้างชั้นข้อมูลรูปทางเรขาคณิต

ตัวอย่างรูปแบบการเก็บรูปร่างทางเรขาคณิต พิมพ์ longitude ก่อนแล้วตามด้วย latitude (EPSG: 4326)

ข้อมูลจุด var mypoint = ee.Geometry.Point([long, lat])

ข้อมูลเส้น var myline = ee.Geometry.LineString([[long1,lat1] , [long2,lat2]])

ข้อมูลพื้นที่ var mypolygon = ee.Geometry.Polygon([[[lon1,lat1], [lon2,lat2], [lon3,lat3]]])

วิธีการสร้างชั้นรูปร่างทางเรขาคณิต นอกจากการเขียนโปรแกรมตรงๆ สามารถสร้างได้จากหน้าต่างแผนที่ด้านล่าง มุมซ้ายบนจะเห็นปุ่มมือ สำหรับการเลื่อนแผนที่ ปุ่ม Marker สำหรับปักหมุด (สร้างข้อมูลจุด Point) ปุ่มสร้างสายเส้น (สร้างข้อมูลเส้น Line String) ปุ่มสร้างรูปร่าง (สร้างข้อมูลรูปปิด Polygon) และปุ่มสร้างสี่เหลี่ยมมุมฉาก กดปุ่มแล้วสร้างรูปได้เลย รูปที่สร้างถูกนำมาเก็บในตัวแปรเรียบร้อย ดูได้จากด้านบนหน้าต่างการทำงาน

สมมติว่าเราต้องภาพครอบคลุมจุด mypoint ที่เราทำการปักไว้ การให้ค้นภาพที่ครอบคลุมบริเวณที่เราต้องการทำได้นำตัวแปรเก็บภาพก่อนหน้าทำผ่าน filterBounds(geometry) เก็บในตัวแปร

var landsat8 = landsat8.filterBounds(mypoint) [5]

ถ้าไม่มีการ filterbound กำหนดพื้นที่ครอบคลุม ระบบจะนำภาพที่ถ่ายทั้งโลกมาบริการเรา

นอกจากนี้เป็นเครื่องมือการกรองภาพเพิ่มเติมตาม Metadata ของภาพ รูปแบบการใช้งานจะเป็นแบบ Namevariable.filterMetadata(Properties, Operator, Value) สำหรับ landsat8 มีดังนี้

Name	Type	Description
CLOUD_COVER	INT	Percentage cloud cover, -1 = not calculated
CLOUD_COVER_LAND	INT	Percentage cloud cover over land, -1 = not calculated
IMAGE_QUALITY	INT	Image quality, 0 = worst, 9 = best, -1 = quality not calculated
GEOMETRIC_RMSE_MODEL	DOUBLE	Combined RMSE (Root Mean Square Error) of the geometric residuals (meters) in both across-track and along-track directions. (Obtained from raw Landsat metadata)
GEOMETRIC_RMSE_MODEL_X	DOUBLE	RMSE (Root Mean Square Error) of the geometric residuals (meters) measured on the GCPs (Ground Control Points) used in geometric precision correction in the along-track direction. (Obtained from raw Landsat metadata)
GEOMETRIC_RMSE_MODEL_X	DOUBLE	RMSE (Root Mean Square Error) of the geometric residuals (meters) measured on the GCPs (Ground Control Points) used in geometric precision correction in the across-track direction. (Obtained from raw Landsat metadata)
EARTH_SUN_DISTANCE	DOUBLE	Earth-Sun distance (AU)
ESPA_VERSION	STRING	Internal ESPA image version used to compute SR
LANDSAT_ID	STRING	Landsat Product Identifier (Collection 1)
LEVEL1_PRODUCTION_DATE	INT	Date of production for raw Level 1 data as ms since epoch
PIXEL_QA_VERSION	STRING	Version of the software used to produce the 'pixel_qa' band
SATELLITE	STRING	Name of satellite
SENSING_TIME	STRING	Time of the observations as in ISO 8601 string
SOLAR_AZIMUTH_ANGLE	DOUBLE	Solar azimuth angle
SR_APP_VERSION	STRING	LaSRC version used to process surface reflectance
WRS_PATH	INT	WRS path number of scene
WRS_ROW	INT	WRS row number of scene

รูปที่ 6 รายละเอียด Metadata ของภาพที่เราสามารถใช้กรองภาพเพิ่มเติมได้

สมมติว่าเราต้องการกรองภาพให้มีเพียงภาพที่มีเมฆปกคลุมน้อยกว่าร้อยละ 20 นำตัวแปรเก็บภาพ [5] ใช้ properties ‘CLOUD_COVER’ ทำ operator ‘less_than’ ด้วย value 20 เก็บลงตัวแปร

```
var landsat8 = landsat8.filterMetadata(‘CLOUD_COVER’, ‘less_than’, 20)    [6]
```

ท้ายสุดคือการเลือกผสมสีภาพ Color Composite ทำได้ผ่านการ band select โดยมีรูปแบบทั่วไปของการเขียน Code ดังนี้ Namevariable.select(['แบนด์ใส่ช่องแดง','แบนด์ใส่ช่องเขียว','แบนด์ใส่ช่องน้ำเงิน']) เช่น

```
var landsat8 = landsat8.select(['B4','B3','B2'])                    [7]
```

บัดนี้ภาพทั้งหมดที่ผ่านการกรองตามคุณสมบัติที่เรากำหนดเก็บไว้ในตัวแปร landsat8 เรียบร้อย เราจะนำตัวแปรนี้ไปแสดงผลเป็นชั้นข้อมูลในแผนที่ต่อไป

เพิ่มเติมเกี่ยวกับการเขียนโปรแกรม เราสามารถเขียน Code เหล่านี้

```
var mypoint = ee.Geometry.Point([100.56933, 13.88423]);          //จุดแสดงบริเวณที่สนใจ
var landsat8 = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA');   //ค้นภาพ landsat8
var landsat8 = landsat8.filterDate('2021-01-01','2021-01-30');   //ค้นวันที่
var landsat8 = landsat8.filterBounds(mypoint);                   //ค้นภาพครอบคลุมจุดที่สนใจ
var landsat8 = landsat8.filterMetadata(‘CLOUD_COVER’, ‘less_than’, 20);          //กรองเมฆ
var landsat8 = landsat8.select(['B4','B3','B2']);                //ใส่สี สร้าง color composite
```

ในรูปแบบนี้

```
var mypoint = ee.Geometry.Point([100.56933, 13.88423]);
var landsat8 = ee.ImageCollection('LANDSAT/LC08/C01/T1_TOA')
                    .filterDate('2021-01-01','2021-01-30')
                    .filterBounds(mypoint)
                    .filterMetadata(‘CLOUD_COVER’, ‘less_than’, 20)
                    .select(['B4','B3','B2']);
```

2.3 การแสดงผลชั้นข้อมูลภูมิสารสนเทศ

Google Earth Engine มีหน้าต่างแสดงแผนที่ไว้ด้านล่าง การแสดงผลชั้นข้อมูลลงในแผนที่ด้านล่าง จะต้องทำการกำหนดศูนย์กลางพื้นที่แสดงผลแผนที่ จากนั้นทำการแสดงชั้นข้อมูล

การกำหนดศูนย์กลางพื้นที่แสดงผลแผนที่ ทำได้ 2 แบบด้วยกัน

แบบที่ 1 กำหนด longitude, latitude และระบุจำนวนเท่าการ zoom ไปยังพื้นที่ รูปแบบทั่วไปของ code คือ

Map.setCenter(longitude,latitude, zoom);

แบบที่ 2 กำหนดจากศูนย์กลางของรูปเรขาคณิตที่กำหนด และระบุจำนวนเท่าของการ zoom ไปยังพื้นที่

Map.centerObject(geometry,zoom);

ต่อไปเป็นการแสดงชั้นข้อมูล รูปแบบทั่วไปคือต้องใส่ชื่อตัวแปรชั้นข้อมูล ตัวแปรการมองเห็น ชื่อชั้นข้อมูล

Map.addLayer(layer variable, visibility, layer name)

สำหรับภาพ landsat-8 ค่าการมองเห็น Visibility ให้กำหนดแบบนี้ {min:0.0, max:0.4}; ทั้งนี้สามารถปรับได้ตาม Scale ช่วงของ Digital Number ใน Raster Grid ของภาพถ่ายว่าค่าสูงสุดเก็บที่เท่าใด

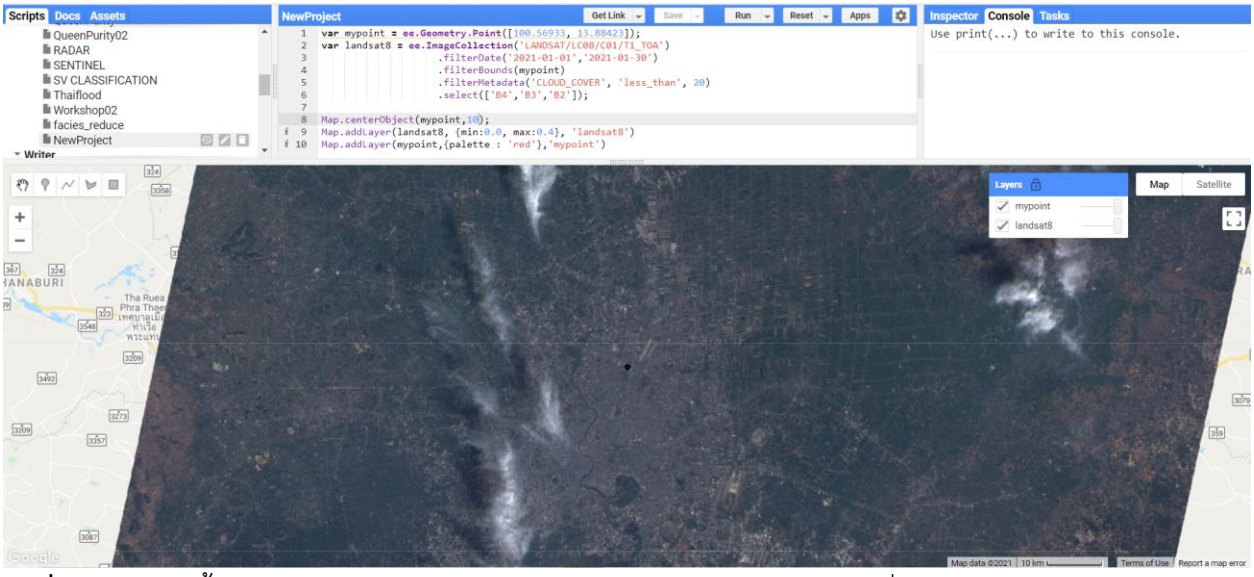
ส่วนการแสดง geometry ค่าการมองเห็น Visibility สามารถใส่เป็น color name หรือ color code

Map.addLayer(geometry,{palette : 'color name or code'}, layer name)

สมมติว่าจะแสดงผลภาพ landsat8 ที่ได้เตรียมกันมา โดยแสดงแผนที่ตรงกลางที่จุด mypoint โดยแสดงผลขยาย zoom เข้าไป 10 เท่า พร้อมแสดงจุด mypoint สีแดง เขียนโปรแกรมเพิ่มเติมได้ดังนี้

```
Map.centerObject(mypoint,10);
Map.addLayer(landsat8, {min:0.0, max:0.4}, ‘landsat8’)
Map.addLayer(mypoint, {palette : 'red'}, 'mypoint')
```

แสดงผลชั้นข้อมูลตามที่ได้เขียนโปรแกรมได้ดังนี้



รูปที่ 7 การแสดงชั้นข้อมูลภาพถ่าย landsat-8 (RGB : 432) บริเวณโดยรอบจุดที่สนใจ (สีแดง)

2.4 ตัวอย่างการประมวลผลข้อมูล

นอกจากการแสดงชั้นข้อมูลต่าง ๆ ที่ Google Earth Engine สามารถนำภาพถ่ายทางอากาศที่เก็บไว้จำนวนมหาศาล กรองมาแสดงผลให้เราเห็นเป็นชั้นข้อมูลแล้ว Google Earth Engine สามารถทำการประมวลผลข้อมูลภูมิสารสนเทศได้ด้วยการเขียนโปรแกรมอีกเช่นกัน

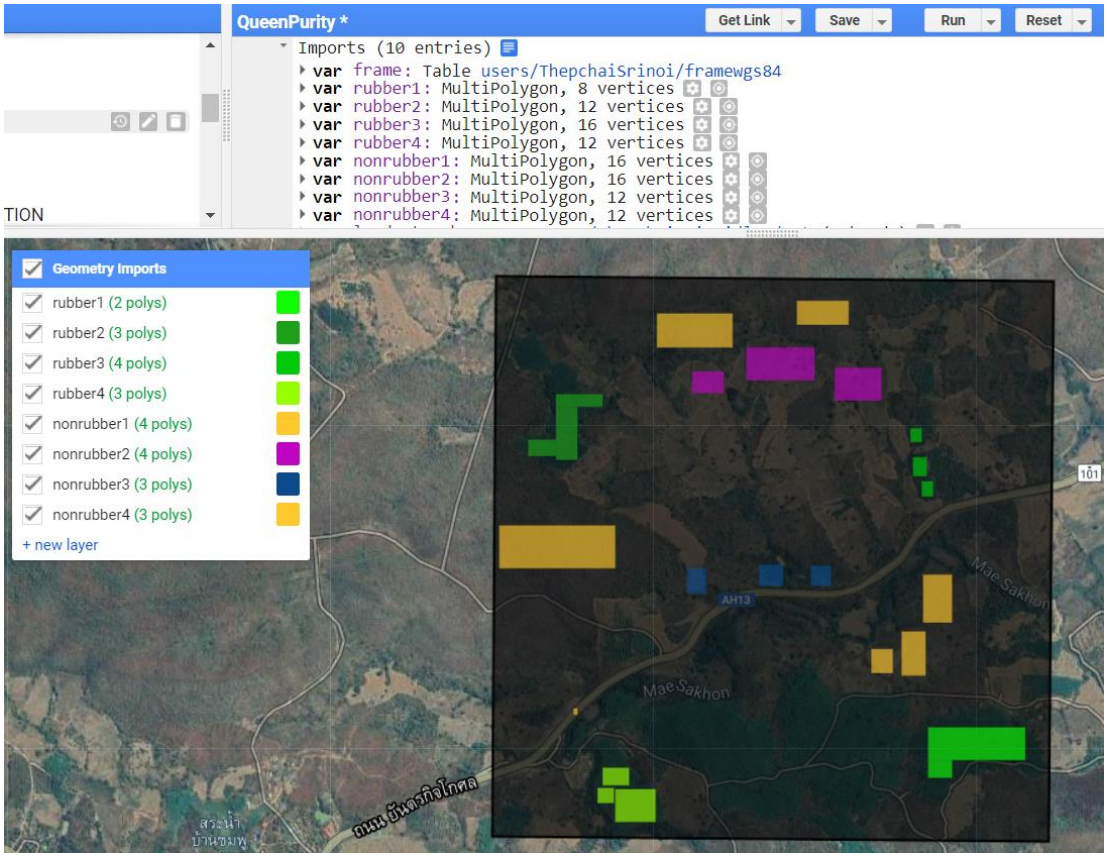
ภาระงานสำคัญของฝ่ายผลิตชั้นข้อมูลภูมิสารสนเทศ คือการจัดทำชั้นข้อมูลภูมิสารสนเทศให้ได้มาตรฐาน มีความความถูกต้อง และแม่นยำมากยิ่งขึ้น นำข้อมูลที่ได้มาให้นำมาตรวจสอบ และปรับปรุงข้อมูลเพิ่มเติมด้วยการแปลตีความสายตาจะทำให้ข้อมูลมีความถูกต้องยิ่งขึ้น โดยดาวน์โหลดภาพถ่ายจากดาวเทียมจาก Google Earth Engine มาเก็บในคอมพิเตอร์ เพื่อมาแปลตีวความด้วยโปรแกรม ArcGIS ปัจจุบันมีความพยายามในการค้นหาเทคนิคในการแปลตีความการใช้ประโยชน์ที่ดินแบบอัติโนมัติ ให้มีความถูกต้องสูง หากสามารถทำได้จะทำให้การวิเคราะห์ข้อมูลได้พื้นที่ขนาดใหญ่มีความรวดเร็ว และลบต้นทุนงบประมาณในการดำเนินงาน ทั้งนี้เรื่องเหล่านี้ยังคงเป็นประเด็นศึกษาวิจัยทาง Remote Sensing ในปัจจุบันทั้งจากการใช้ Hyperspectral Remote Sensing การใช้เทคนิควิธีการ Feature Selection หรือ Feature Transformation การเลือกตัวจำแนกที่เหมาะสม เพื่อให้ได้ Overall Accuracy สูงสุด Google Earth Engine สามารถเข้ามาเป็นเครื่องมือช่วยในการแปลตีความอัตโนมัติได้ และคาดว่าจะเข้ามาเป็นเครื่องมือในการทำวิจัยทางด้าน Remote Sensing ต่อไป

เบื้องต้นขอยกตัวอย่างการใช้ Google Earth Engine ในการจำแนกยางพาราในพื้นที่ด้วยการจำแนกแบบ Supervised Classification โดย Machine Learning : CART (Classification and Regression Trees)

//กำหนดกรอบสี่เหลี่ยมล้อมรอบพื้นที่ที่สนใจ นำเข้า Shapefile เข้ามา ตั้งชื่อ frame

var frame = ee.FeatureCollection("users/ThepchaiSrinoi/framewgs84")

//กำหนด polygon ล้อมรอบพื้นที่ป่ายางพารา นำมาใช้เป็น training data สำหรับ machine learning

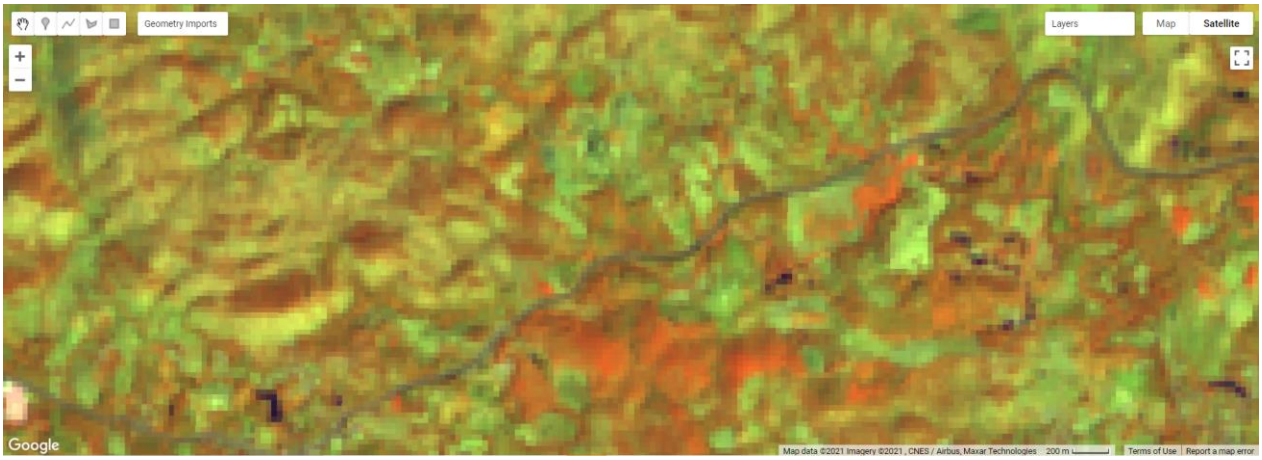


รูปที่ 8 polygon กรอบพื้นที่ที่สนใจ และ training polygon สีเขียวแทนยางพารา สีอื่นไม่ใช่ยางพารา

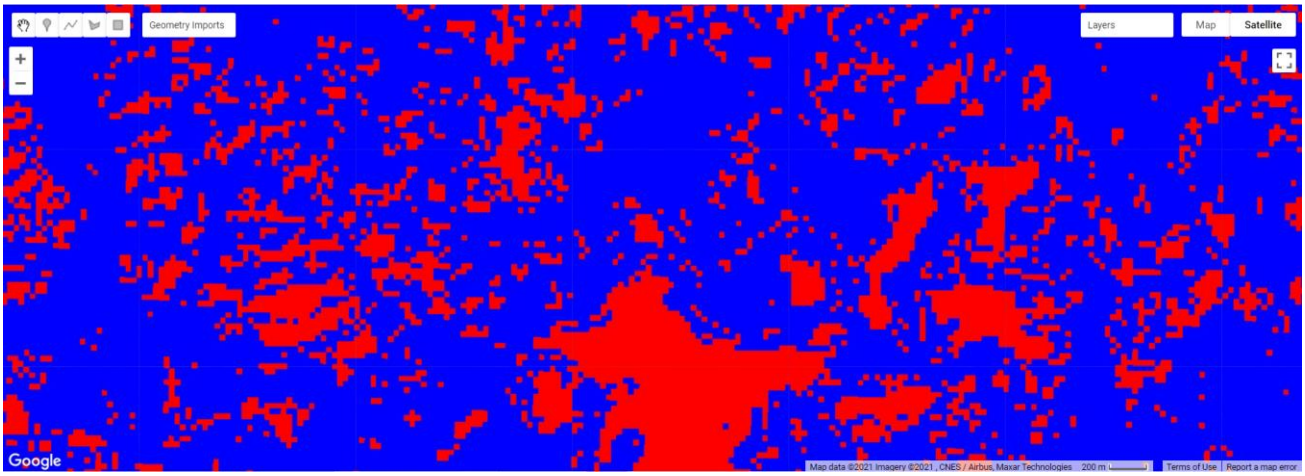
```
//เลือกภาพ landsat8 กรองวันที่ เลือกที่ครอบคลุมกรอบพื้นที่ที่ต้องการ เลือก Bands มาจำแนก
var l8 = ee.ImageCollection('LANDSAT/LC08/C01/T1');
var image = ee.Algorithms.Landsat.simpleComposite({
        collection: l8.filterDate('2021-02-10','2021-02-13').filterBounds(frame),
        asFloat: true
        });
var bands = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B10', 'B11'];

//ส่วนของการ Training Data สร้าง Polygon ครอบคลุมพื้นที่ที่เป็นยางพารา (จำแนกเป็น class 0) กับ
Polygon ที่ไม่เป็นยางพารา (จำแนกเป็น class 1)
var polygons = ee.FeatureCollection([ee.Feature(rubber1, {'class': 0}),
                ee.Feature(rubber2, {'class': 0}),
                ee.Feature(rubber3, {'class': 0}),
                ee.Feature(rubber4, {'class': 0}),
                ee.Feature(nonrubber1, {'class': 1}),
                ee.Feature(nonrubber2, {'class': 1}),
                ee.Feature(nonrubber3, {'class': 1}),
                ee.Feature(nonrubber4, {'class': 1}),]);

//กำหนด training data ในตัวแปร polygons กำหนดขนาด pixel เท่ากับ 30 เมตร
var training = image.sampleRegions({collection: polygons,properties: ['class'],scale:30});

//นำ training data ไปสอนตัวจำแนก CART แยก class 0 และ 1 โดยใช้ bands ตามที่กำหนดไว้แล้ว
var trained = ee.Classifier.smileCart().train(training, 'class', bands);

//เมื่อสอนแล้ว ให้ CART ไปจำแนกยางพาราในพื้นที่ที่เหลือในภาพ LANDSAT-8 ที่กรองกันมา
var classified = image.select(bands).classify(trained);
```

```
//ส่วนของการแสดงผลชั้นข้อมูล
//การกำหนดศูนย์กลางของแผนที่และจำนวนเท่าของการ Zoom เข้า
Map.centerObject(frame, 15);
//การแสดงผลภาพ LANDSAT-8 แบบ False Color Composite แบบ 5-6-4
Map.addLayer(image, {bands: ['B5', 'B6', 'B4'], max: 0.4}, 'image');
//การแสดงผลภาพการจำแนกป่ายางพารา กำหนดให้ สีแดงแสดงยางพารา สีน้ำเงินไม่เป็นยางพารา
Map.addLayer(classified, {min: 0, max: 1, palette: ['red','blue']},'classification');
```

กด Save แล้วกด Run ได้ผลการจำแนกดังนี้



รูปที่ 9 ภาพถ่ายทางดาวเทียมจาก LANDSAT-8 False Color Composite (5-6-4)



รูปที่ 10 ภาพการ Supervised Classification (CART) เพื่อจำแนกยางพารา (สีแดง)

การทดสอบความถูกต้องของการจำแนก

```
var confusionMatrix = ee.ConfusionMatrix(validation.classify(trained)
                                .errorMatrix({
                                        actual:'class',
                                        predicted : 'classification'
                                        }));
print('Confusionmatrix', confusionMatrix);       //รายงานตารางแสดงความถูกต้อง
//ถัดจากนี้เป็นการรายงานค่าความถูกต้อง (รายงานเป็นค่า float ตั้งแต่ 0 ถึง 1)
print('Consumers Accuracy',confusionMatrix.consumersAccuracy());
print('Producers Accuracy',confusionMatrix.producersAccuracy());
print('Overall Accuracy',confusionMatrix.accuracy());
```

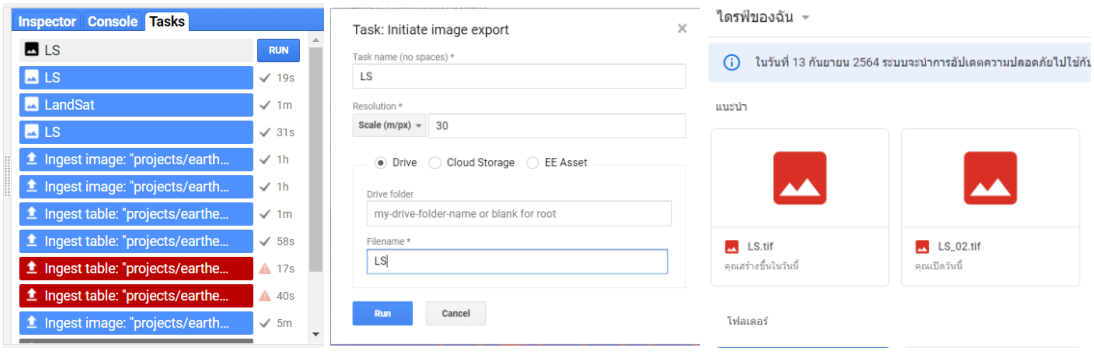
2.5 การส่งออกข้อมูล

ข้อมูลที่ได้สร้าง ทำการค้นคว้า ประมวลผล ใน Google Earth Engine สามารถทำการส่งออกไปยัง Google Drive ของเราได้ทั้งในรูปแบบของ GeoTiFF หรือ Shapefile แล้วแต่ประเภทของข้อมูล ขอยกตัวอย่างกรณีการส่งออกภาพถ่ายทางดาวเทียม landsat-8 ที่ 30 m per pixel

Export.image.toDrive({image: imagelayer, description: "name", scale: m per px, region:geometry});

ตัวอย่างเช่น หากต้องการส่งออก Landsat-8 ภายในพื้นที่ frame ที่เราสนใจออกมา ใช้ code นี้

```
Export.image.toDrive({image:image, description: "LS", scale:30, region:frame});
```



รูปที่ 11 ลำดับของการส่งออกไฟล์ Landsat-8 มายัง Google Drive ของเรา