

**ĐẠI HỌC QUỐC GIA HÀ NỘI**  
**ĐẠI HỌC CÔNG NGHỆ**



**BÁO CÁO MÔN HỌC**  
**MÔN: TRÍ TUỆ NHÂN TẠO**  
**TUẦN 3**

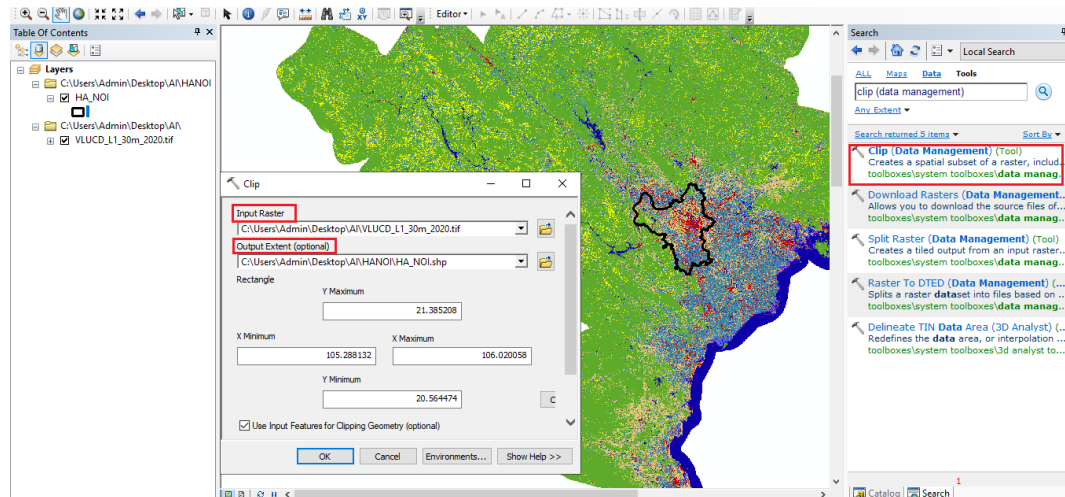
**Thành viên:**  
**Hoàng Quốc Anh**  
**Nguyễn Bá Chung**  
**Trần Hữu Quốc Đông**  
**Phạm Quang Hùng**  
**Ngô Thị Ngọc Quyên**

# MỤC LỤC

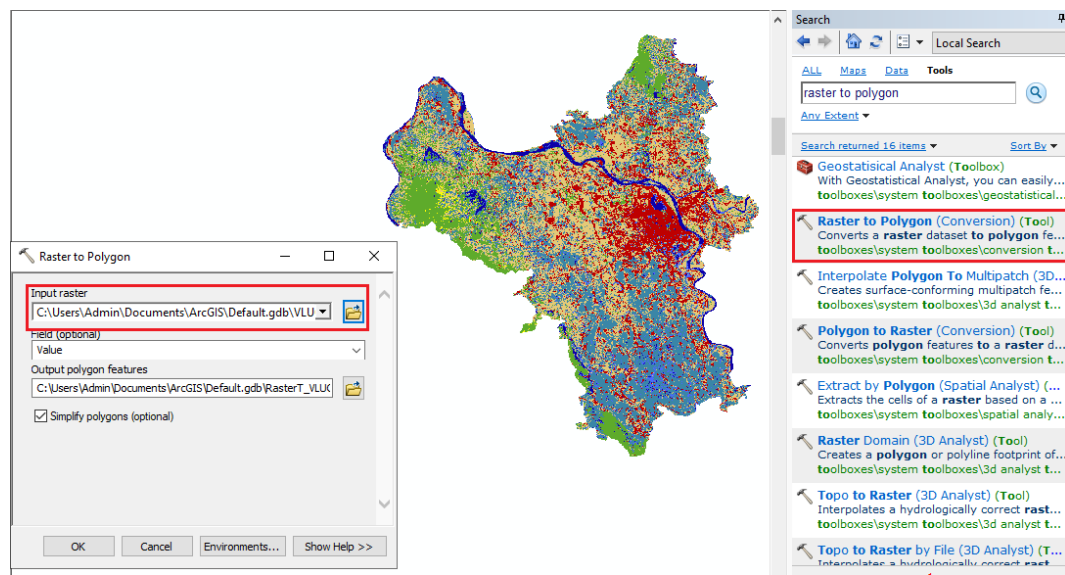
<b>I. Lấy lớp phủ của Hà Nội trên phần mềm Arcmap .....</b>	<b>3</b>
<b>II. Tạo các điểm mẫu cho từng loại lớp phủ .....</b>	<b>3</b>
1. Tạo các điểm mẫu ngẫu nhiên trên phần mềm Arcmap .....	4
2. Sử dụng Google Earth Pro để kiểm tra các điểm mẫu .....	5
3. Xuất các điểm mẫu dưới dạng file csv .....	9
<b>III. Matching nhận các điểm lấy mẫu với bản đồ ảnh vệ tinh .....</b>	<b>9</b>
<b>IV. Kết quả .....</b>	<b>12</b>
1. Link code .....	12
2. Link các dữ liệu mẫu .....	12
3. Link file dữ liệu .....	12

# I. Lấy lớp phủ của Hà Nội trên phần mềm Arcmap

**Bước 1:** Sử dụng ảnh lớp phủ của Việt Nam và shapefile Hà Nội để tạo ảnh lớp phủ của Hà Nội. Sử dụng tool ‘Clip’ để cắt lấy lớp phủ của Hà Nội, với input là ảnh lớp phủ của Việt Nam, output là shape file Hà Nội.



**Bước 2:** Chuyển ảnh lớp phủ Hà Nội từ **Raster** sang **Polygon**, sử dụng tool ‘Raster to Polygon’ với Input raster là ảnh lớp phủ thu được ở trên.



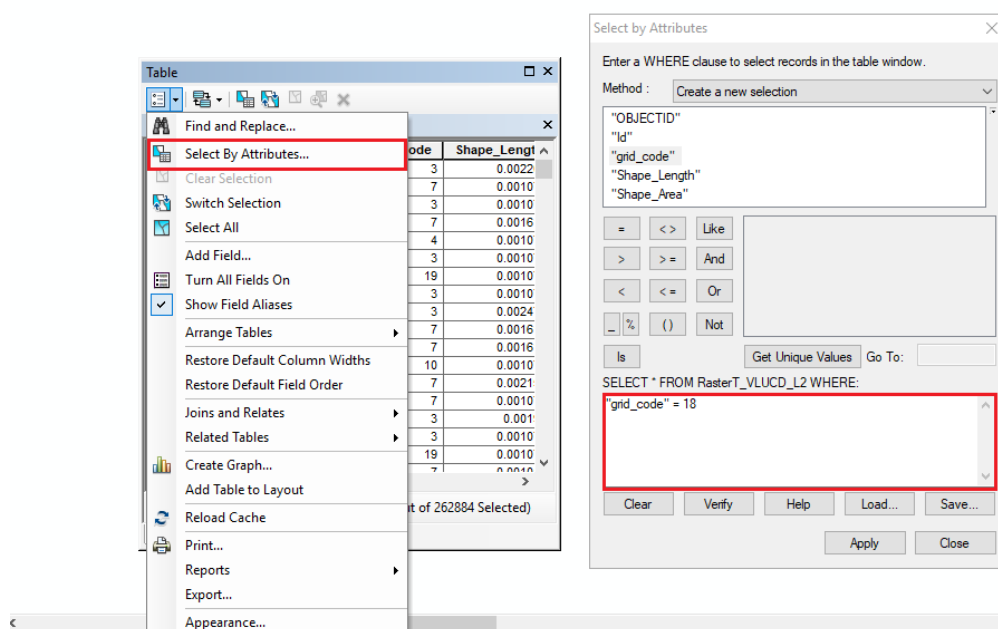
## II. Tạo các điểm mẫu cho từng loại lớp phủ

Bản đồ lớp phủ có phân loại các lớp phủ như sau:

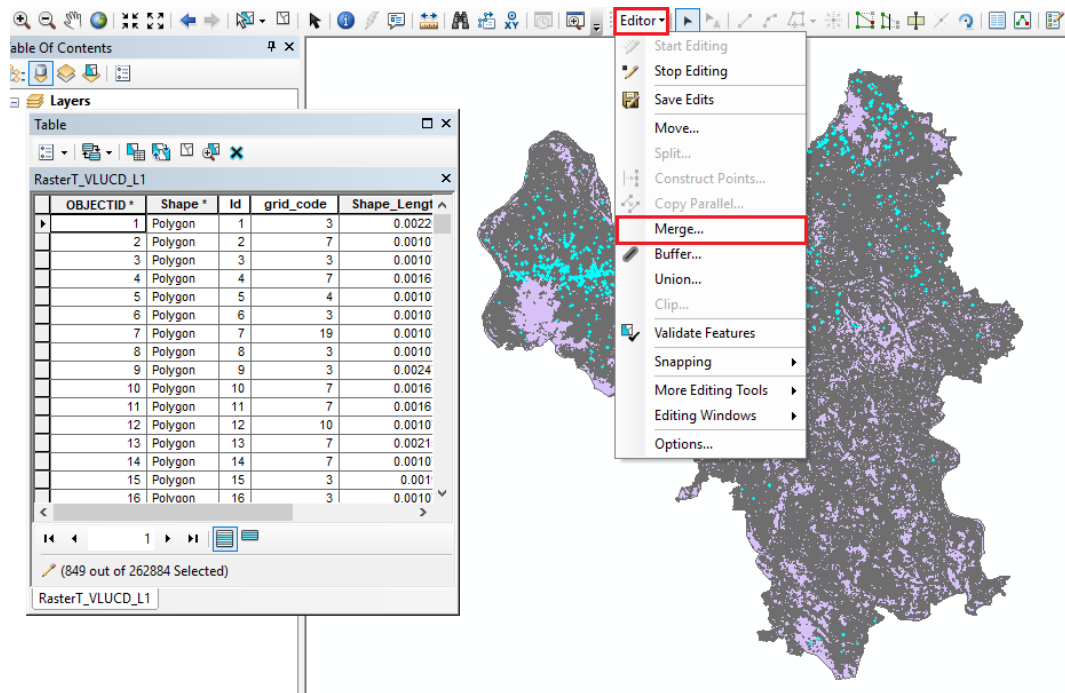
Tên	Residential Land	Rice Paddies	Croplands	Grassland	Barren Land
Grid Code	1, 2	3	4,5,6	7	8
Tên	Scrub	Forests	Wetlands	Open water	Aquaculture
Grid Code	9	10,11,12,14,20	15	18	19

## 1. Tạo các điểm mẫu ngẫu nhiên trên phần mềm Arcmap

Trên ảnh lớp phủ dạng polygon, mở bảng thuộc tính, chọn polygon cùng một lớp phủ theo bảng trên, sau đó ‘Export data’ tạo một file dạng polygon của từng lớp phủ:

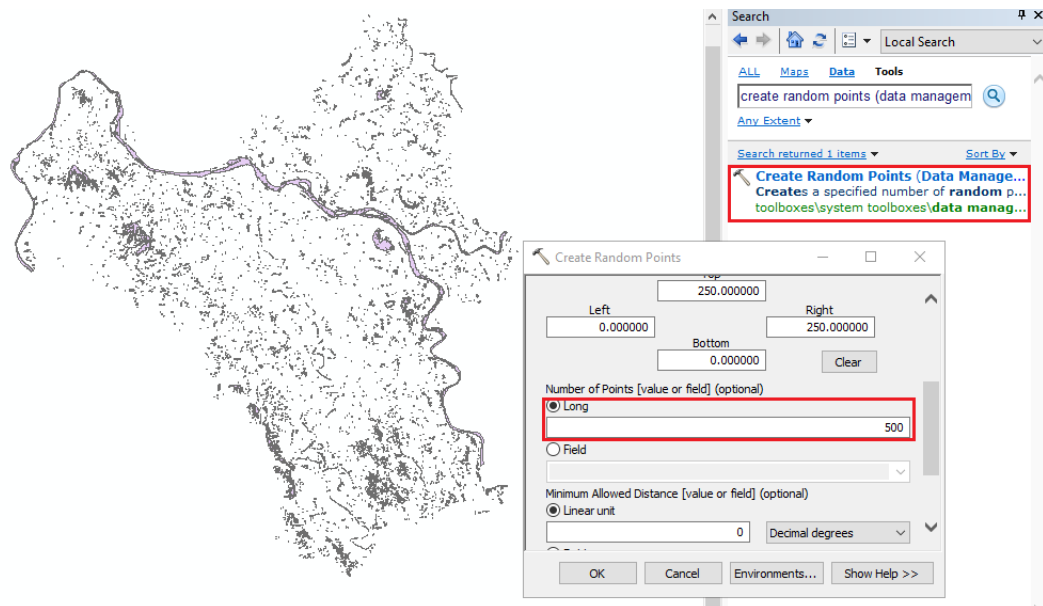


Các lớp phủ đang tồn tại ở dạng polygon rời rạc vì vậy cần phải merge chúng lại thành một polygon cho mỗi loại lớp phủ. Mở bảng thuộc tính, chọn ‘Select all’. Sau đó chọn tool ‘Editor’ bắt đầu để ‘Merge’ các polygon đã chọn ở bước trên. Các polygon của mỗi lớp phủ sẽ được gộp vào thành một polygol.



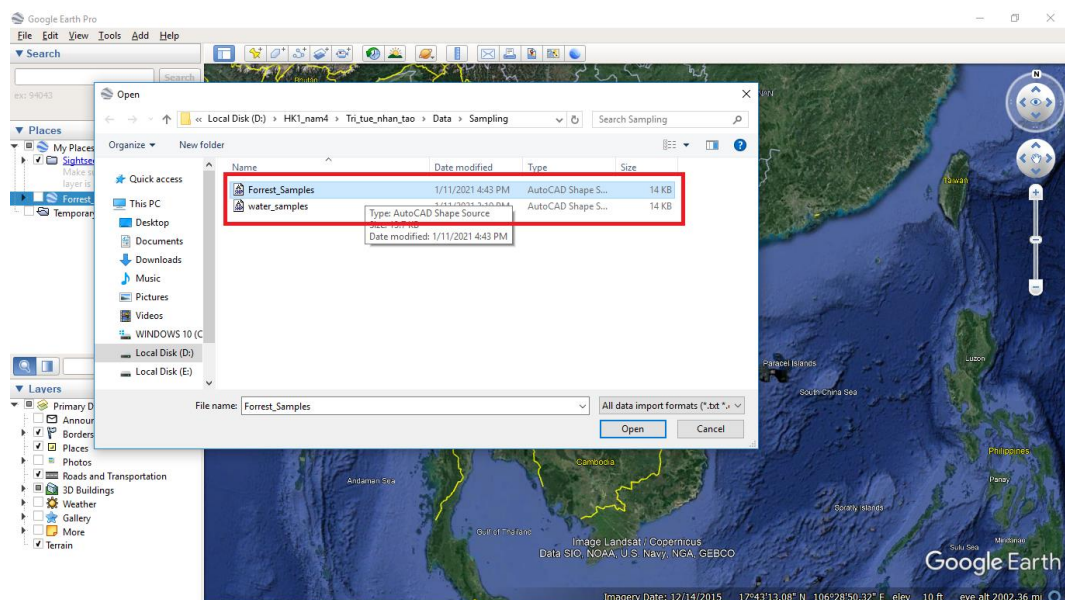
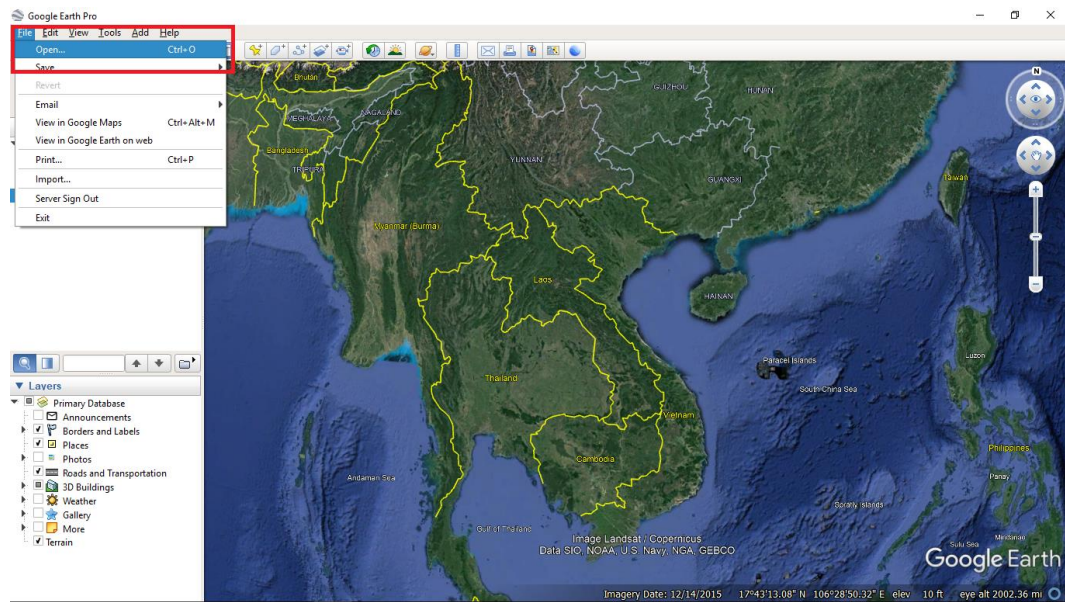
Thực hiện tương tự với các lớp phủ còn lại.

Tiếp theo dùng tool 'Create random points' để tạo ra các điểm ngẫu nhiên trên từng lớp phủ. Export file vừa tạo ra dưới dạng '.shp'.



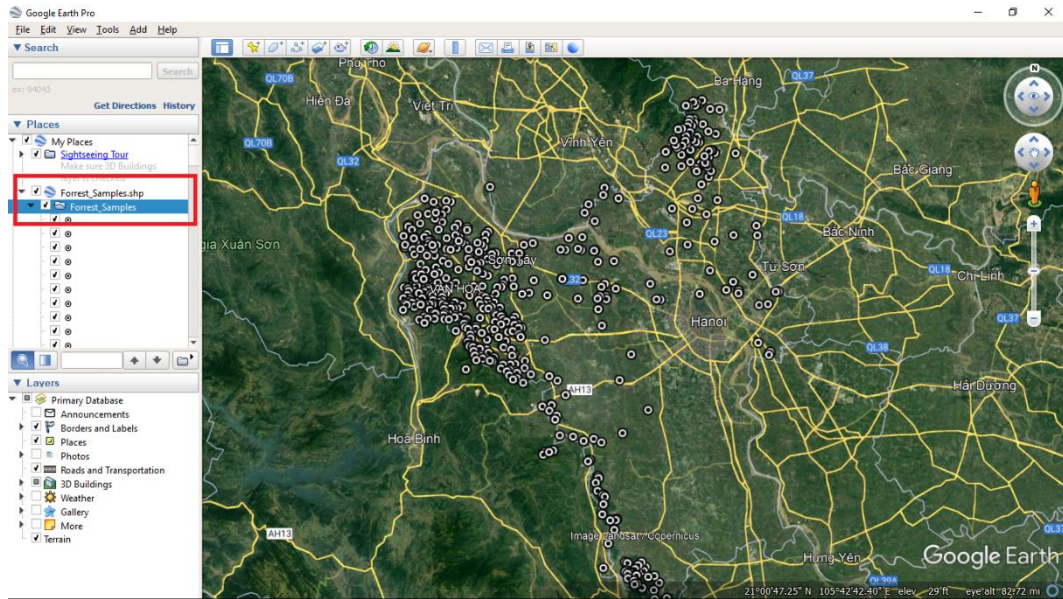
## 2. Sử dụng Google Earth Pro để kiểm tra các điểm mẫu

**Bước 1:** Chọn *File – Open* để mở file .shp

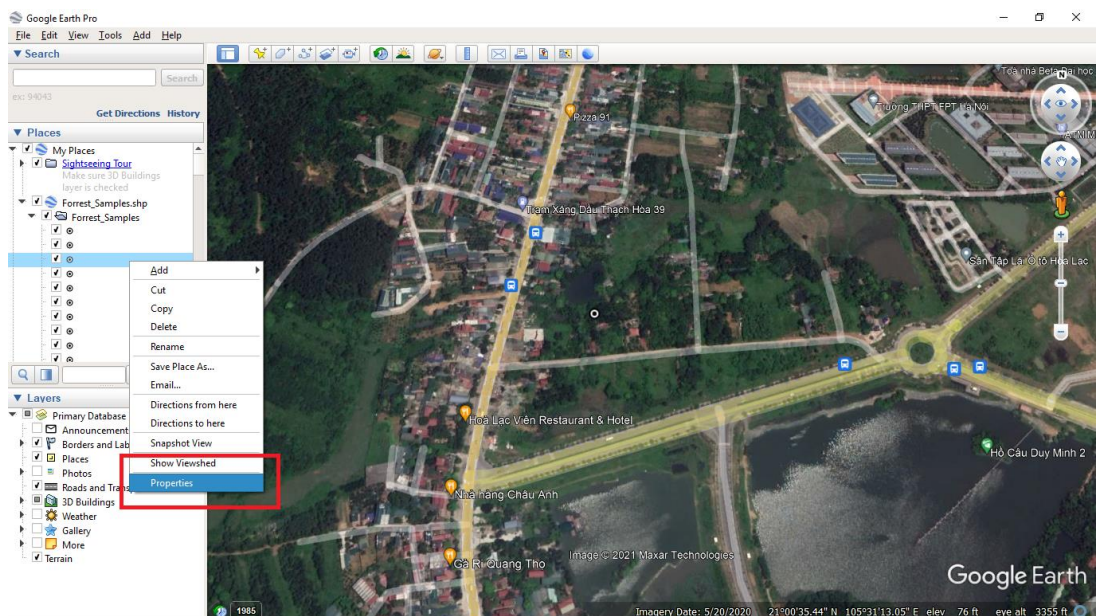


**Bước 2:** Từ bảng Places bên trái, để hiện các điểm đã lấy mẫu, ta tích vào ô vuông bên cạnh tên file đã mở ở bước 1.



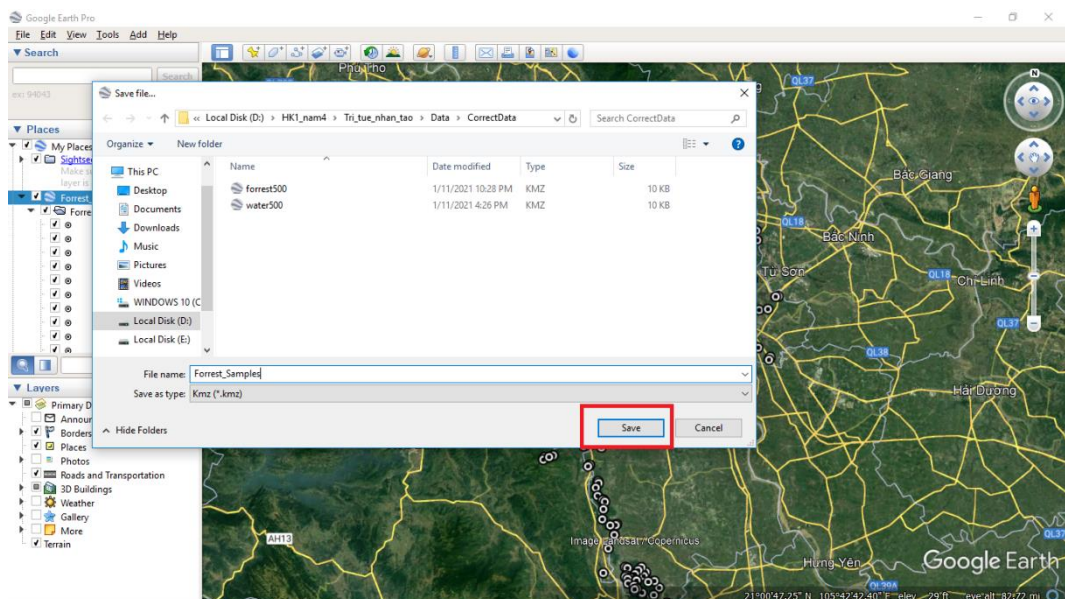
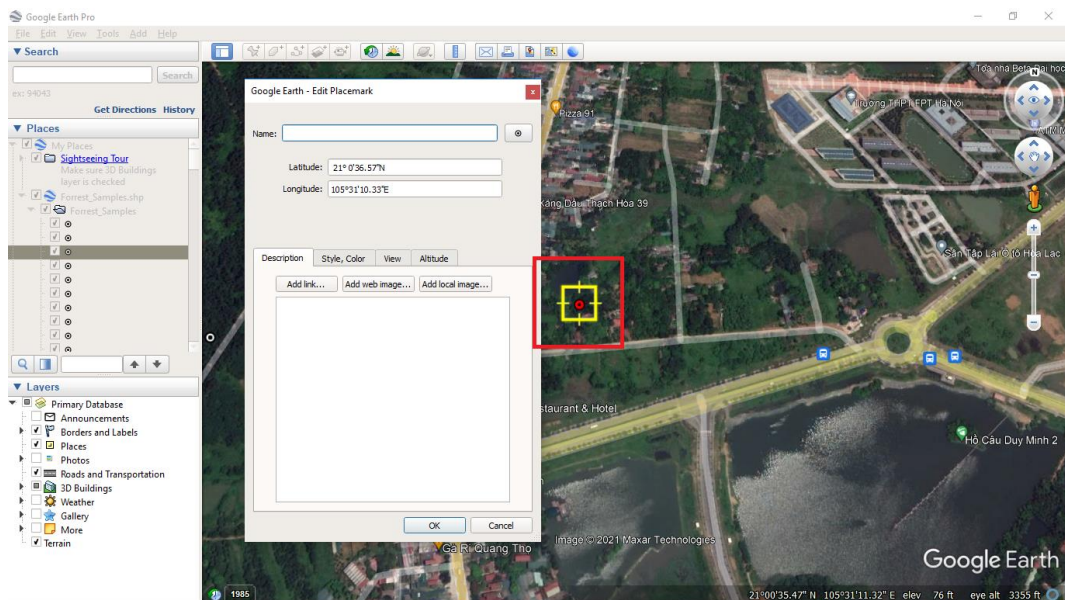
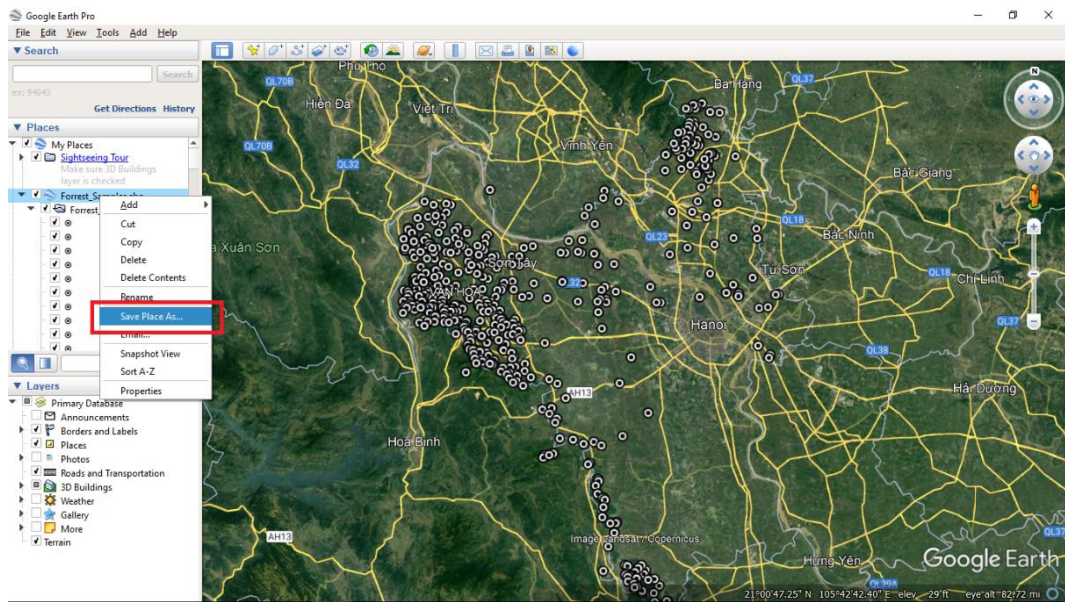


**Bước 3:** Với các điểm bị lệch so với thực tế, ta chuột phải vào điểm đó chọn Properties, sau đó có thể di chuyển điểm được chọn tới vị trí đúng.



**Bước 4:** Khi các điểm tại vị trí đúng, ta có thể xuất ra file .kmz bằng cách chuột phải vào file .shp đã mở chọn *Save Place As*.

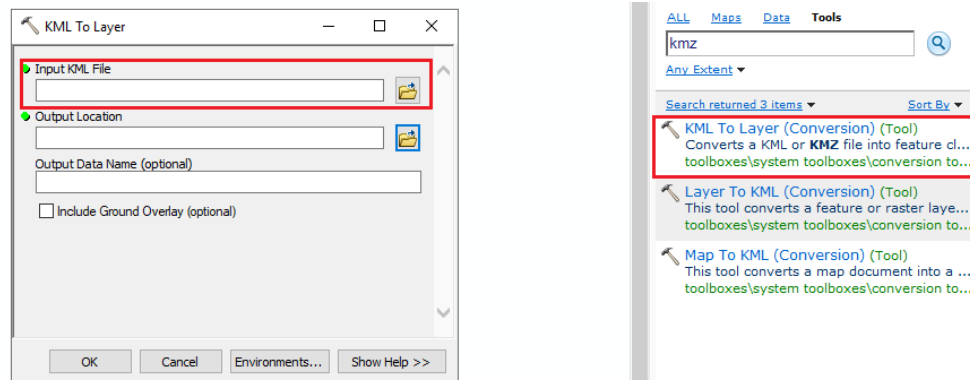




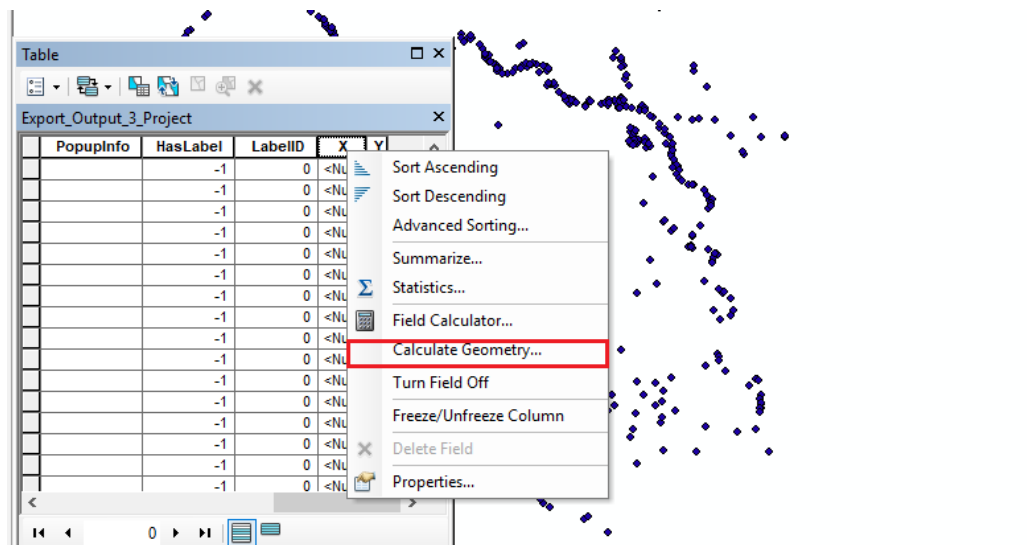


### 3. Xuất các điểm mẫu dưới dạng file csv

Đọc file ‘.kmz’ vừa xuất từ Google Earth Pro sử dụng tool ‘KML to layer’ trên phần mềm Arcmap.



Export data vừa đọc từ file kmz, sau đó dùng tool ‘Project (data management)’ để chuyển sang hệ quy chiếu UTM. Mở bảng thuộc tính, thao tác ‘Add field..’ để thêm hai trường X, Y dạng *Float*. Bật chế độ ‘Editor’ để thêm giá trị vào hai trường vừa tạo với ‘Calculate Geometry..’



Sau khi thao tác xong với hai trường X,Y, copy bảng thuộc tính vào excel để tạo tọa độ và nhãn cho từng lớp phủ.

### III. Matching nhãn các điểm lấy mẫu với bản đồ ảnh vệ tinh

Quá trình thực hiện bằng chương trình code Python với đầu vào là các điểm lấy mẫu và ảnh vệ tinh hệ tọa độ UTM, đầu ra là file Excel chứa tọa độ cũng như giá trị các band của các điểm mẫu. Chương trình bắt đầu với các đoạn lệnh nhập thư viện như sau:

```
# Import Libraries
```

```
import os
import pandas as pd
from osgeo import gdal
import matplotlib.pyplot as plt
```

Trong đó, gdal là thư viện đọc ảnh vệ tinh, pandas là thư viện đọc và xuất file excel các điểm mẫu.

Các đầu vào được đọc và xử lý bằng các đoạn lệnh tiếp theo:

```
FILE_PATH = os.listdir('Datasets/WithoutCoords')
MAP_PATH = r'Input/LANDSAT_7_Bands_30M_UTM.tif'

ds = gdal.Open(MAP_PATH, gdal.GA_ReadOnly)

# ulx, uly: up left (x,y)
# xres, yres: x,y scale

ulx, xres, xskew, uly, yskew, yres = ds.GetGeoTransform()

# Read Excel File
for file in FILE_PATH:
    df = pd.read_excel('Datasets/WithoutCoords/' + file,
engine='openpyxl')
```

Trong đó, *ds* là ma trận ảnh vệ tinh, *df* là ma trận chứa các điểm lấy mẫu. Hàm `GetGeoTransform()` trả về tọa độ (x,y) phía trên bên trái của bản đồ thực (ulx, uly) và độ dài của mỗi pixel trên thực tế (xres, yres).

Chương trình lấy giá trị các band của các điểm lấy mẫu từ ma trận ảnh vệ tinh như sau:

```
x = df['X']
y = df['Y']

# Get Bands Value
for k in range(ds.RasterCount):
    band_k = ds.GetRasterBand(k + 1)
    band_k = band_k.ReadAsArray()
    b_k = []
```

```

for i in range(len(x)):
    pixel_x = (x[i] - ulx) / xres
    pixel_y = (y[i] - uly) / yres
    pixel_x = int(pixel_x)
    pixel_y = int(pixel_y)

    val = band_k[pixel_y][pixel_x]

    b_k.append(val)

df_b = pd.DataFrame(b_k)
df['Band_{}'.format(k + 1)] = df_b

```

Để biết được tọa độ trên ảnh mỗi điểm (pixel\_x, pixel\_y), ta lấy tọa độ của điểm ảnh đó trên thực tế trừ đi tọa độ (x,y) phía trên cùng bên trái rồi mới chia cho độ dài thực tế của một pixel.

Đến đây, với mỗi band ảnh, ta được các giá trị band tại vị trí điểm đó.

Sau đó, lưu các kết quả tìm được dưới dạng DataFrame rồi xuất ra file excel. Đoạn lệnh xuất kết quả ra file Excel như sau:

```

# Export file

df.to_excel('Datasets/Full/' + file + '_7_Bands.xlsx',
index=False)

```

Kết quả từ file Excel có dạng như sau:



X	Y	crop	Band_1	Band_2	Band_3	Band_4	Band_5	Band_6	Band_7
596561,4	2307390	crop	0,038535	0,04178	0,07907	0,059985	0,358113	0,19017	0,094195
580513,2	2280905	crop	0,030478	0,034341	0,062666	0,044434	0,273344	0,152509	0,074106
564875	2310640	crop	0,02682	0,035153	0,070848	0,070793	0,141303	0,065018	0,033255
600131,4	2329248	crop	0,033145	0,041643	0,083773	0,05861	0,407365	0,191848	0,09018
561014,8	2325250	crop	0,027865	0,040928	0,079813	0,059353	0,257903	0,132613	0,061443
578192,5	2303480	crop	0,04189	0,050223	0,089905	0,074258	0,249295	0,160635	0,087815
570048,6	2346948	crop	0,05179	0,059545	0,09612	0,091253	0,317138	0,233868	0,132695
575338,8	2318879	crop	0,031963	0,03749	0,070545	0,054898	0,301476	0,165338	0,083855
579960,8	2339171	crop	0,040763	0,046703	0,09403	0,06972	0,34538	0,18291	0,098843
582960,1	2291589	crop	0,045603	0,057441	0,104054	0,096588	0,230554	0,09733	0,048366
574912,9	2308826	crop	0,031316	0,042495	0,076403	0,072278	0,20931	0,100933	0,047954
585124,2	2343301	crop	0,023823	0,034245	0,078548	0,06411	0,300748	0,137618	0,06708
587153,5	2344906	crop	0,033104	0,039635	0,085904	0,058583	0,344019	0,181329	0,086908
566355,5	2307220	crop	0,031193	0,038728	0,07621	0,05993	0,23505	0,130055	0,062158
576887,3	2338376	crop	0,038948	0,047844	0,087719	0,072883	0,319833	0,191298	0,101909
571638,6	2325152	crop	0,02781	0,033956	0,068565	0,053729	0,319131	0,181686	0,08732
592255,4	2339934	crop	0,04079	0,045933	0,089878	0,060288	0,15376	0,068758	0,041423
563963,9	2319926	crop	0,03903	0,03936	0,082645	0,058143	0,250973	0,142843	0,062488
584151,8	2334623	crop	0,035785	0,046675	0,088585	0,077173	0,136573	0,049563	0,028305
533769	2330020	crop	0,020138	0,027246	0,063313	0,048751	0,284358	0,15607	0,066489
569338,9	2329797	crop	0,04442	0,053165	0,092353	0,083938	0,346178	0,197513	0,118285
564894,4	2317304	crop	0,033998	0,04453	0,097385	0,083553	0,277483	0,154833	0,074505
565336,4	2306645	crop	0,036005	0,043499	0,076224	0,062034	0,250409	0,156758	0,073955
571598,7	2316793	crop	0,034548	0,047253	0,08941	0,065705	0,237718	0,148178	0,064743

## IV. Kết quả

### 1. Link code

[https://github.com/Windrist/BaiTap\\_AI\\_UET](https://github.com/Windrist/BaiTap_AI_UET)

### 2. Link các dữ liệu mẫu

<https://drive.google.com/drive/folders/1pKa5C2HoLzZT0Ay79Z2Bo6elZwbJMZ-k?usp=sharing>

### 3. Link file dữ liệu

[https://github.com/Windrist/BaiTap\\_AI\\_UET/tree/master/Datasets](https://github.com/Windrist/BaiTap_AI_UET/tree/master/Datasets)