# Aerial Imagery and OpenStreetMap Retrieval

Geospatial Vision and Visualisation

Haoyu Wei
Tzu-jui Liu
Yuchao Wang

# Overview

## Objective

To download the satellite image, retrieve road network map, and overlay the map on to the satellite image

## Code:

https://github.com/whywww/Aerial-Imagery-and-OpenStreetMap-Retrieval

## Execution:

### Requirements (tested on):

- matplotlib==3.2.1
- numpy==1.18.3
- opencv-python==4.2.0.34
- overpy==0.4

### Run:

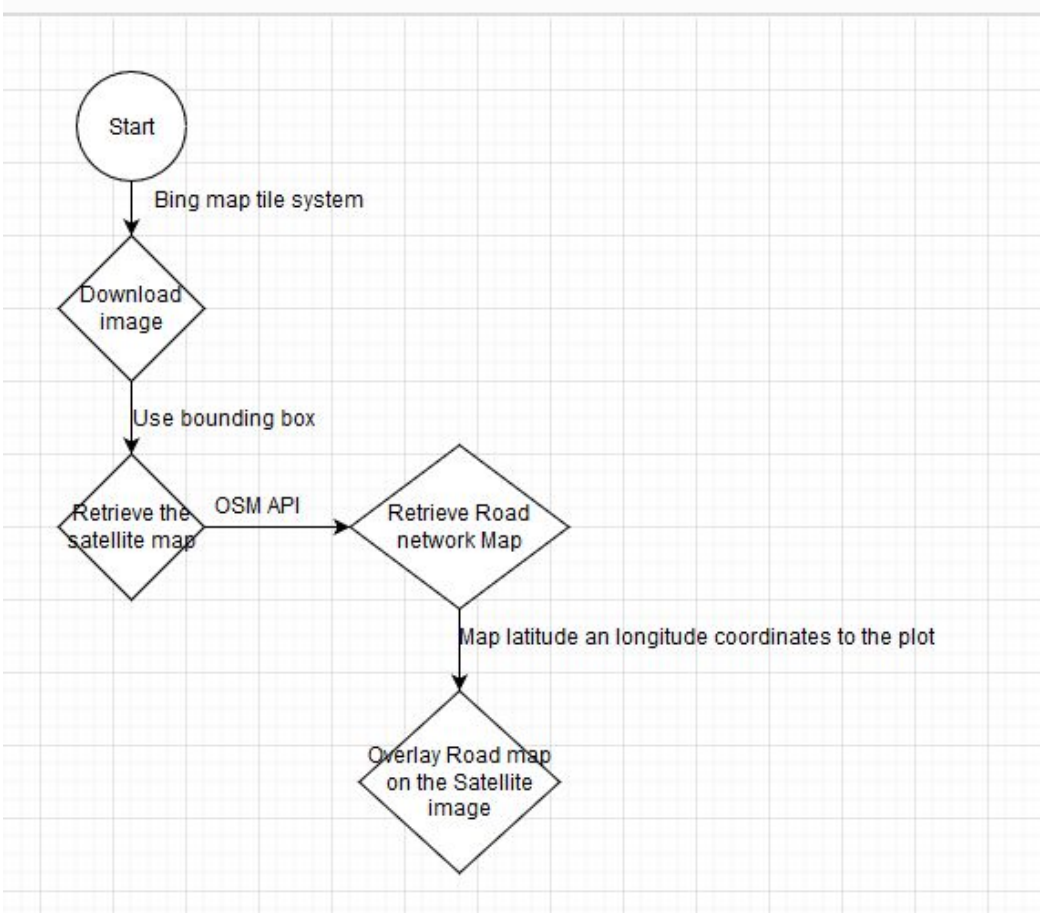
```
python3 code/get_aerial.py
python3 code/OSM_retrieval.py
```

# Our Approach

We use the location of **Northwestern University** as our satellite image.

- **Image Download**: firstly specify the bounding box ([south latitude, west longitude, north latitude, east longitude]) and highest map resolution (2000*1500) to download the aerial image from Bing maps tile system.
- **Cropping to fit**: Cropping the highest resolution map downloaded from the first step to fit the bounding box area.
- **Retrieve Road Network**: Write OSM API Query language to retrieve highway longitudes and latitudes in the Bounding Box area. Here we chose highway with values: ‘primary’|‘secondary’|‘tertiary’|‘residential’|‘cycleway’|‘path’ to display.
- **OverLay the road network map on the satellite image**: Map latitude and longitude of the nodes on the highways to the aerial map image and plot.

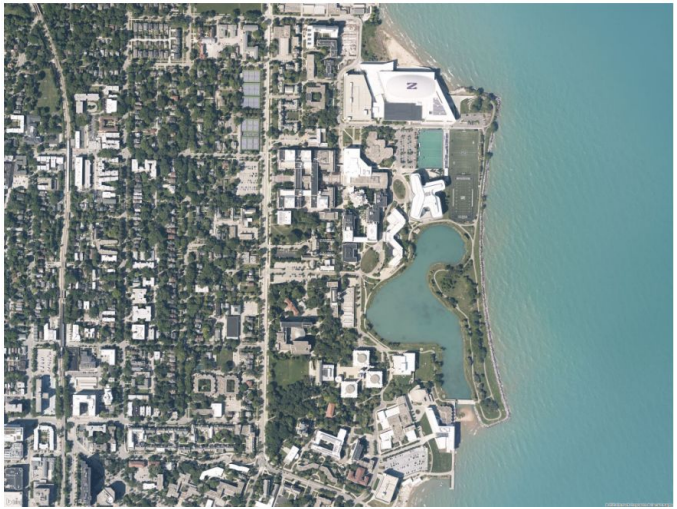
# Flow Chart


Start
Bing map tile system
Download image
Use bounding box
Retrieve the satellite map
OSM API
Retrieve Road network Map
Map latitude an longitude coordinates to the plot
Overlay Road map on the Satellite image

# Our Approach

```
bounding_box = [42.047000, -87.682300, 42.062000, -87.668500]

# [South Latitude, West Longitude, North Latitude, East Longitude]
```



Downloaded Satellite Image

Satellite image within bounding box

# OSM API Query

```
api = overpy.Overpass()
result = api.query("""
   way(42.047000, -87.682300, 42.062000, -87.668500) ["highway"~"primary|secondary|tertiary|residential|cycleway|path"] ;
   (._;>;);
   out body;
   """)
```

**Partial Results:**

Highway: primary
- Lat: 42.049451, Lon: -87.678119
- Lat: 42.049537, Lon: -87.678076
- Lat: 42.050086, Lon: -87.677798

Highway: secondary
- Lat: 42.046703, Lon: -87.681510
- Lat: 42.046891, Lon: -87.681425
- Lat: 42.047050, Lon: -87.681362

Highway: secondary_link
- Lat: 42.049974, Lon: -87.681640
- Lat: 42.049950, Lon: -87.681697
- Lat: 42.049918, Lon: -87.681735
- Lat: 42.049869, Lon: -87.681767
- Lat: 42.049796, Lon: -87.681811

Highway: tertiary
- Lat: 42.049978, Lon: -87.681814
- Lat: 42.049974, Lon: -87.682597

Highway: path
- Lat: 42.052445, Lon: -87.671132
- Lat: 42.052532, Lon: -87.671241

Highway: residential
- Lat: 42.051271, Lon: -87.681845
- Lat: 42.051278, Lon: -87.682830

Highway: tertiary
- Lat: 42.048573, Lon: -87.674734
- Lat: 42.048595, Lon: -87.674671
- Lat: 42.048615, Lon: -87.674584
- Lat: 42.048632, Lon: -87.674494

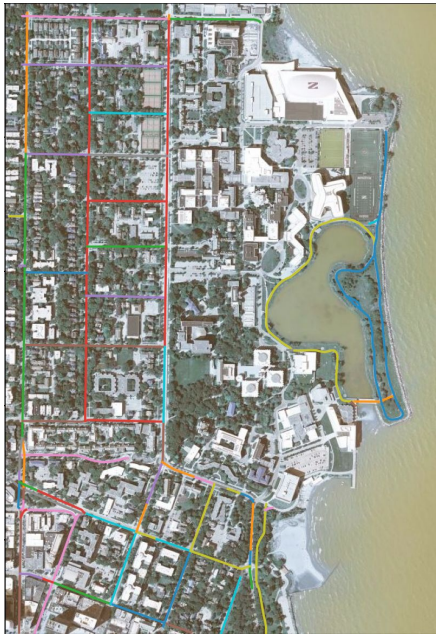
# Overlay Map On Satellite Image

Satellite image within bounding box

+

Road Network Map

=



Final Image

# Cool Place on Earth - Grand Prismatic Spring

[44.523640, -110.840288, 44.526883, -110.836061]

# Challenges and Findings

- Challenges:

  The most challenging part of this project is to read documents to get familiar with the API.

- Conclusions:

  We use the map of the Northwestern University in this project. From what we showed in the slides before, we are able to get a highly accurate road network map from the satellite image and also successfully overlay the road map on the satellite image.

# Reference

[1] Bing Map API: https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system

[2] OpenStreetMap API: https://python-overpy.readthedocs.io/en/latest/