SimCenter NHERI
Center for Computational Modeling and Simulation

2021 Programming Bootcamp

BRAILS

Chaofeng Wang
University of California, Berkeley

NSF award: CMMI 1612843

Slides:

https://github.com/NHERI-SimCenter/SimCenterAI_Workshop2021/blob/master/presentations/day1/Part3.pdf

Demos:

https://github.com/charlesxwang/SimCenterAI_Workshop2021/tree/master/notebooks/day1

Outline

Part 3 BRAILS

Introduction

Architecture

Modules

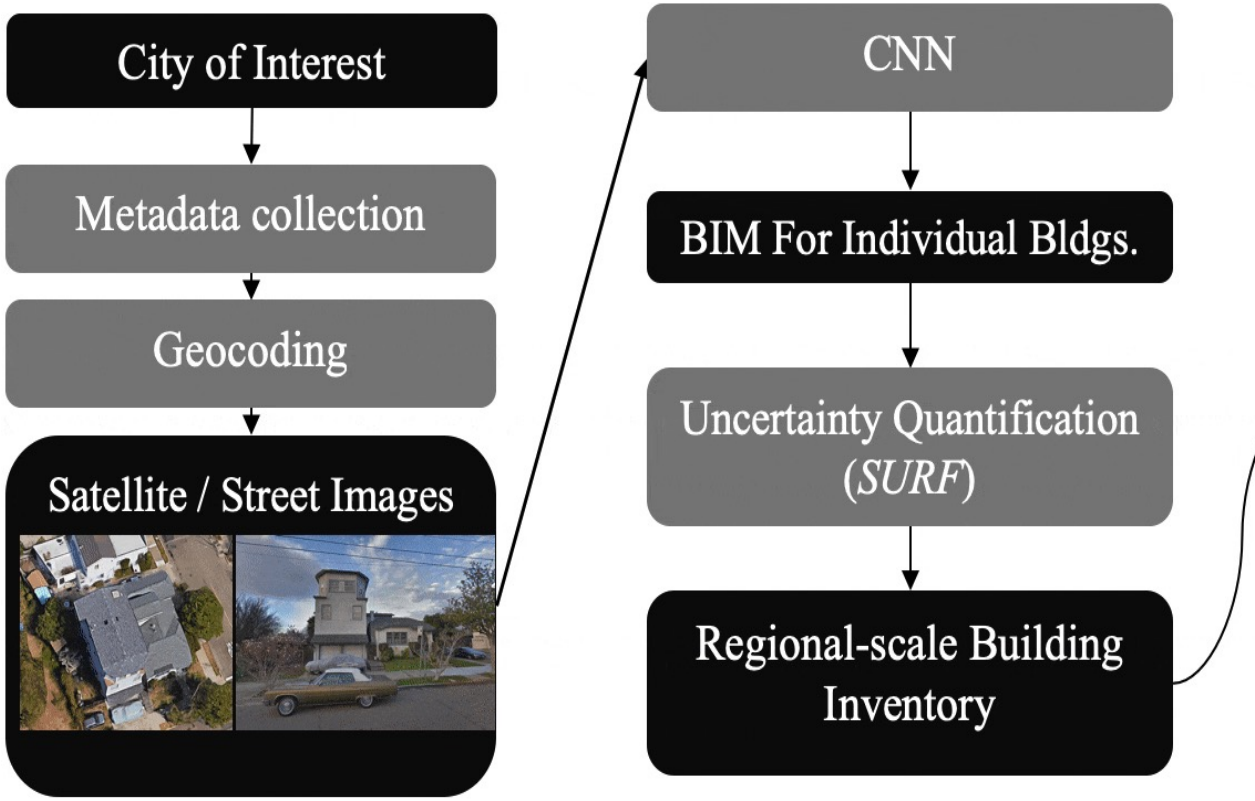
Workflow

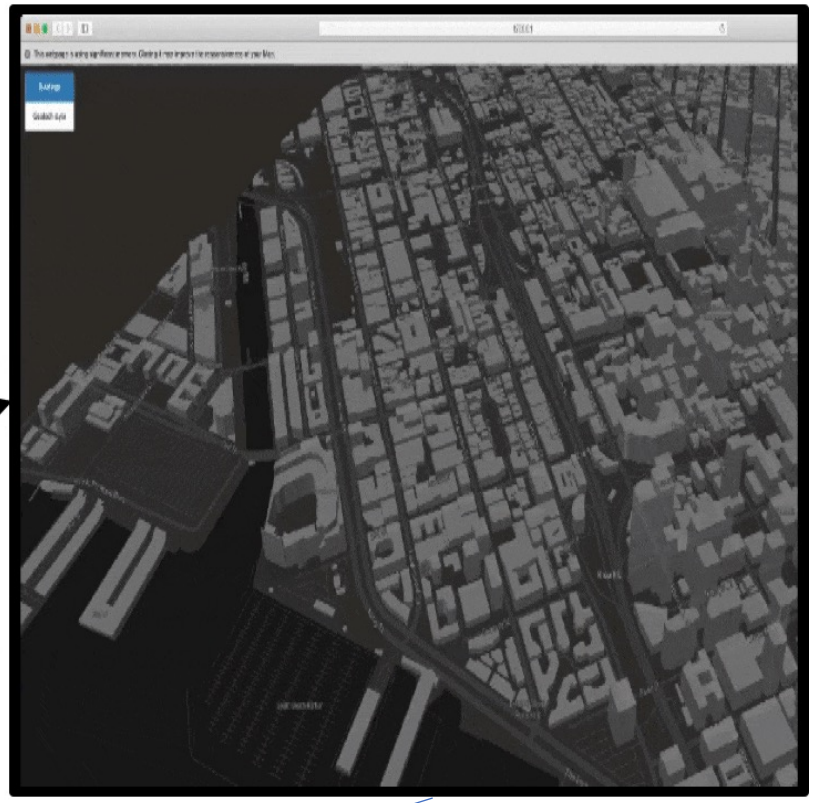
Demos & Exercises

Part 3
BRAILS

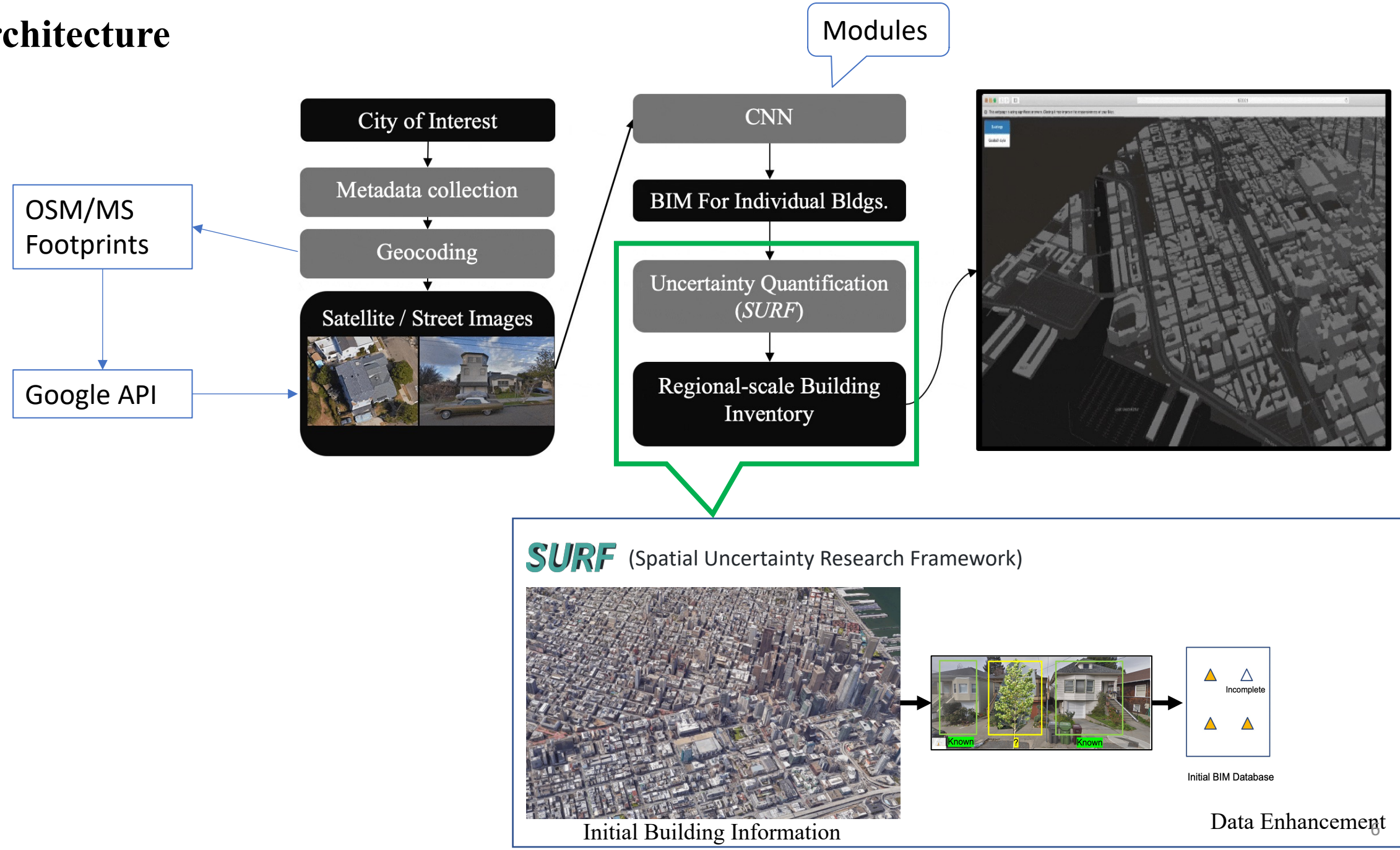
Introduction

BRAILS (Building Recognition using AI at Large-Scale)

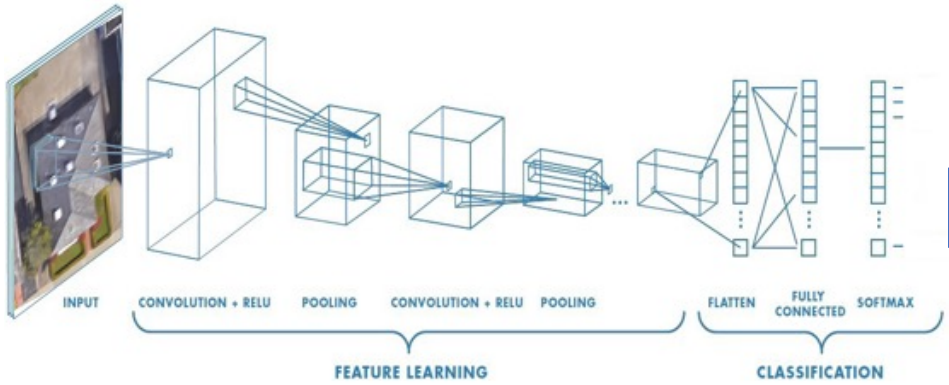




Architecture

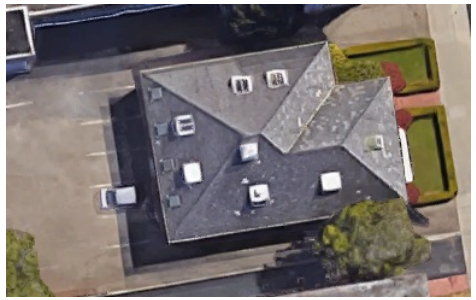


Modules

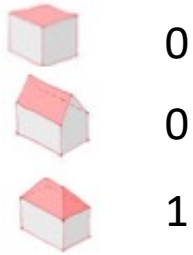


Algorithms

Classification



Classify

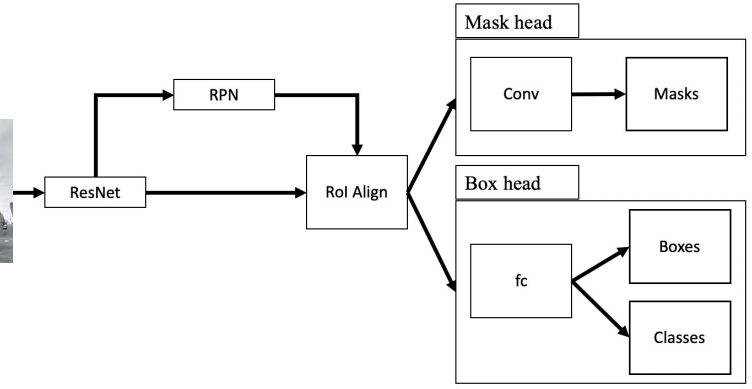


0

0

1

Applications



Segmentation/Detection

Modules

https://nheri-simcenter.github.io/BRAILS-Documentation/common/user_manual/modules/modules.html

Current version has pretrained models to detect the following info from street and satellite images:

Attributes	Accuracy
Roof shape	90% (OpenStreetMap)
Occupancy class	97% (OpenStreetMap)
Soft-story	83% (San Jose + Berkeley)
Number of stories	86% (New Jersey)
Year built	Under study
Foundation elevation	Under study

More validations:
https://nheri-simcenter.github.io/BRAILS-Documentation/common/technical_manual/vnv.html

Modules – Use an existing module

https://nheri-simcenter.github.io/BRAILS-Documentation/common/user_manual/modules/roofClassifier.html

```
# import the module
from brails.modules import RoofClassifier

# initialize a roof classifier
roofModel = RoofClassifier()

# define the paths of images in a list
imgs = ['image_examples/Roof/gabled/76.png',
        'image_examples/Roof/hipped/54.png',
        'image_examples/Roof/flat/94.png']

# use the model to predict
predictions = roofModel.predict(imgs)
```

```
Image :  image_examples/Roof/gabled/76.png     Class : gabled (83.21%)
Image :  image_examples/Roof/hipped/54.png     Class : hipped (100.0%)
Image :  image_examples/Roof/flat/94.png       Class : flat (97.68%)
Results written in file roofType_preds.csv
```

Modules – Retrain an existing model with new data

https://nheri-simcenter.github.io/BRAILS-Documentation/common/user_manual/modules/roofClassifier.html

```
# Load images from a folder
roofModel.loadData('my_roof_shapes')

# Re-train it for only 1 epoch for this demo. You can increase it.
roofModel.retrain(initial_epochs=1)

# Test the re-trained model
predictions = roofModel.predict(imgs)

# Save the re-trained model
roofModel.save('myCoolNewRoofModelv0.1')
```

Modules – Build your new model (classification)

https://nheri-simcenter.github.io/BRAILS-Documentation/common/user_manual/modules/genericImageClassifier.html

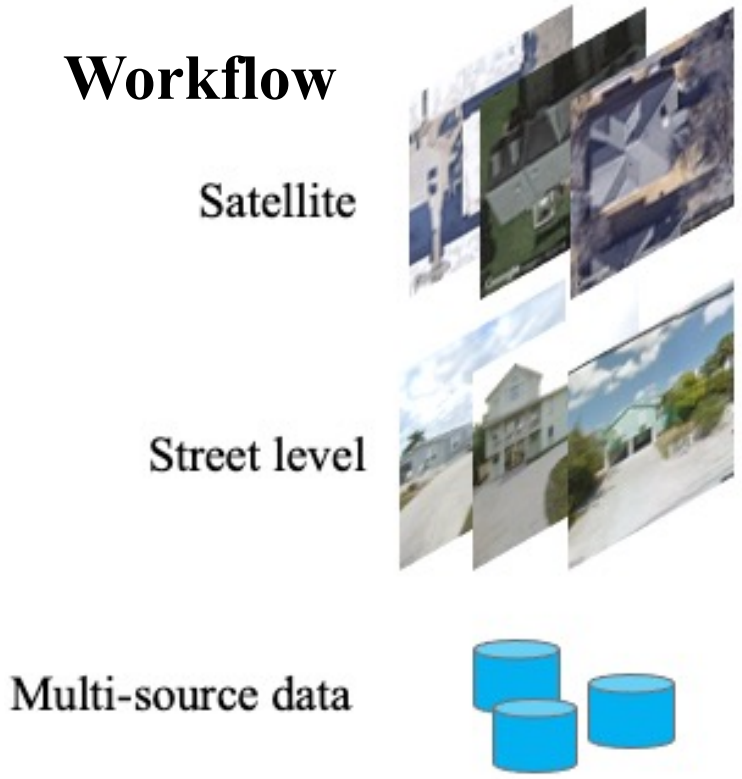
Construct the image classifier

```
# import the module
from brails.modules import ImageClassifier

# initialize the classifier, give it a name
materialClassifier = ImageClassifier(modelName='materialClassifierV0.1')

# load data
materialClassifier.loadData('building_materials')
```

Train the model

```
# train the base model for 50 epochs and then fine tune for 200 epochs
materialClassifier.train(baseModel='InceptionV3', initial_epochs=50,fine_tune_epochs=200)
```

Workflow




```
{
 "id": ""6583",
 "properties": {
 "stories": 3,
 "yearBuilt": 1920,
 "occupancy": "Residential",
 "structureType": "Frame",
 "roofType": "Hipped",
 "elevated": "Yes"
 }
}
```

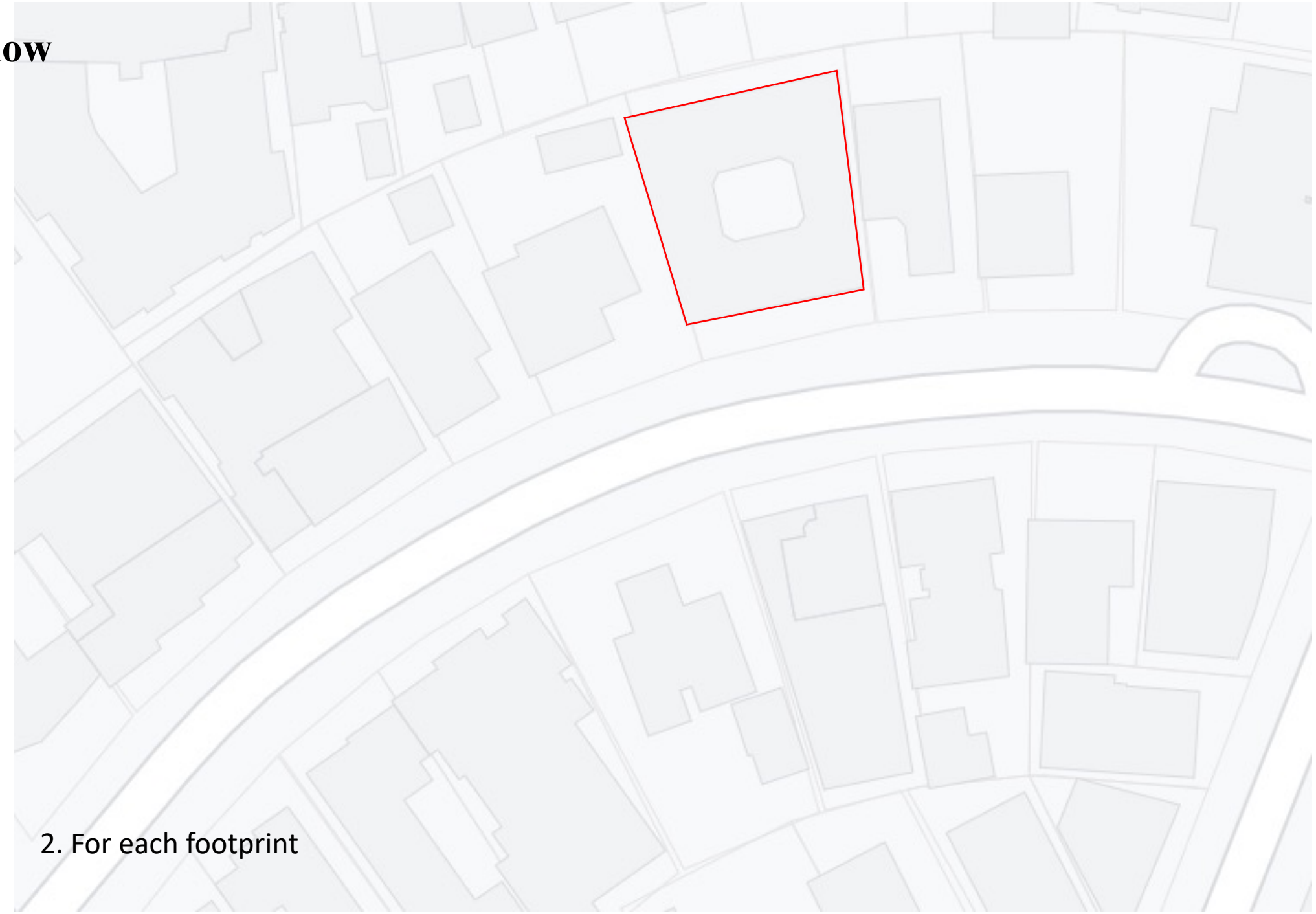

Estimated damage state based on HAZUS
Atlantic City area, New Jersey

```
from brails.CityBuilder import CityBuilder

cityBuilder = CityBuilder(attributes,
                  numBldg,
                  random,
                  bbox,
                  place,
                  footPrints,
                  save,
                  fileName,
                  workDir,
                  GoogleMapAPIKey,
                  overwrite,
                  reDownloadImgs)
```

Workflow

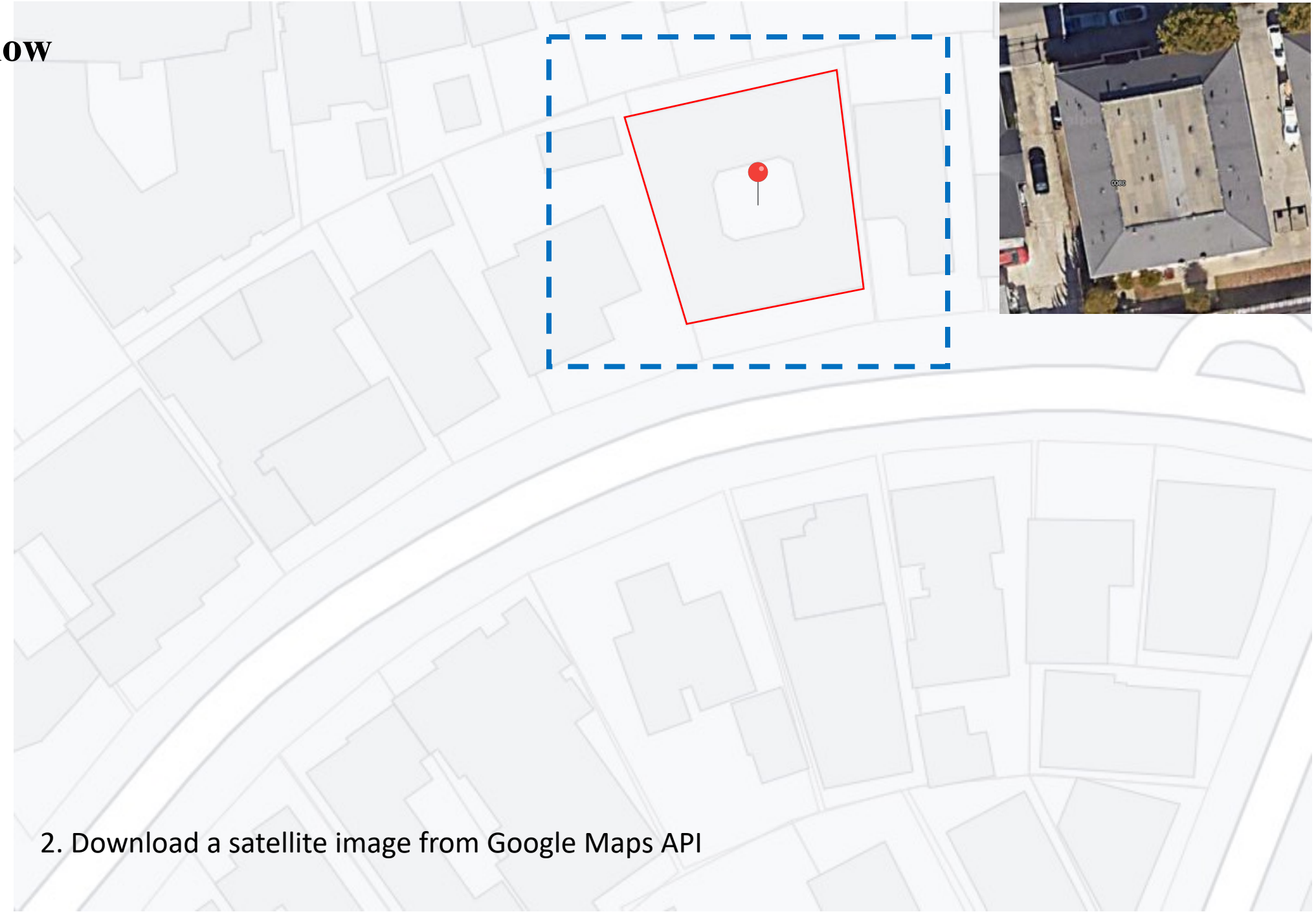
1. Footprints of buildings in region of interest (ROI)
(OpenStreetMap / Microsoft Footprint Dataset)

Workflow



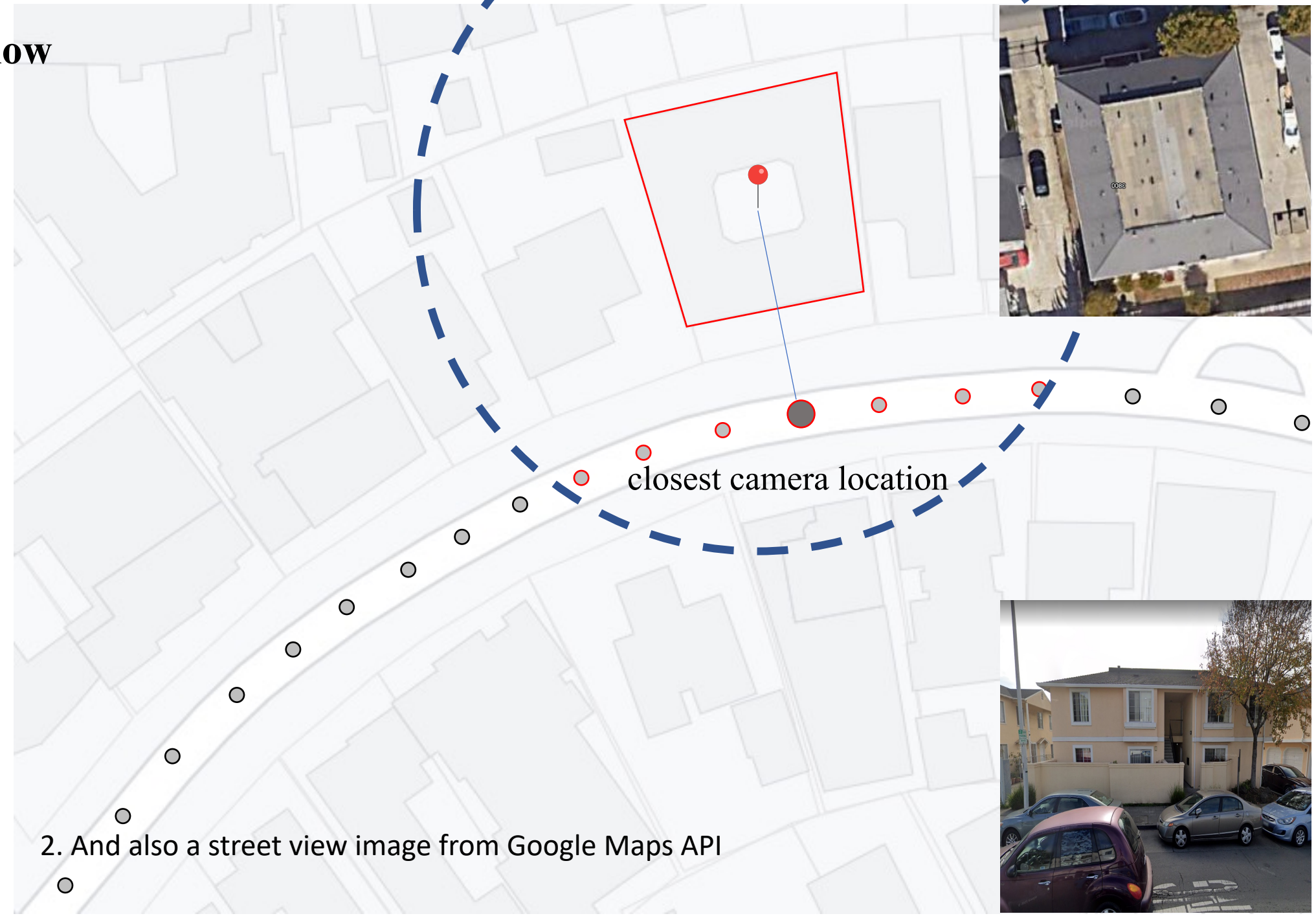
2. For each footprint

Workflow



2. Download a satellite image from Google Maps API

Workflow

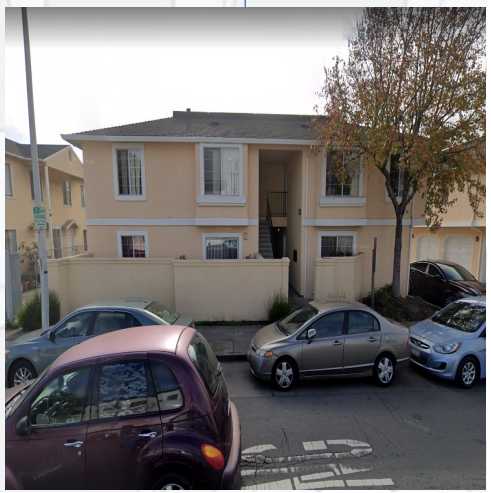


2. And also a street view image from Google Maps API

Workflow

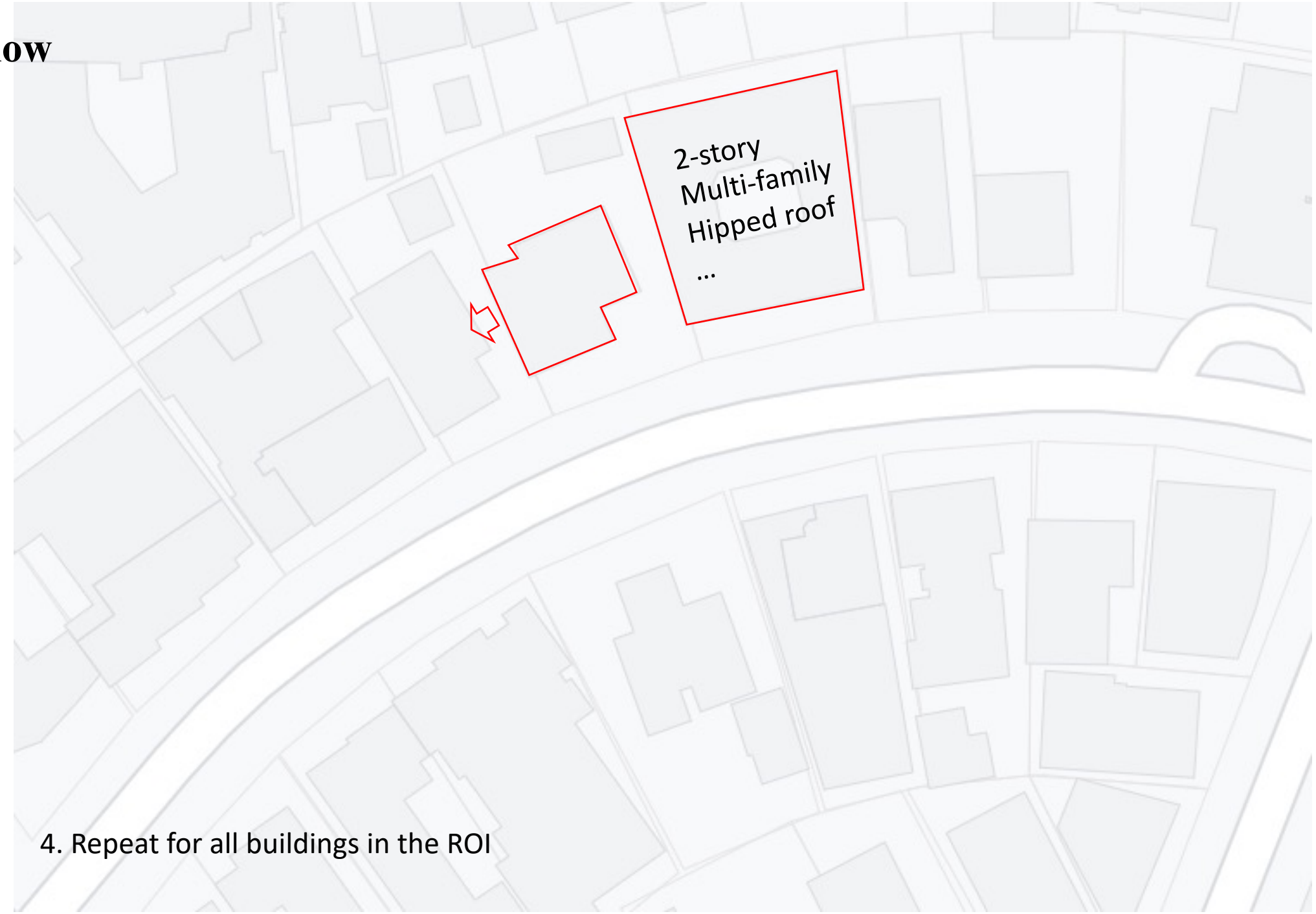
2-story
Multi-family
Hipped roof
…

Modules



3. Infer from images using modules

Workflow



4. Repeat for all buildings in the ROI

Exercises

```
from brails.CityBuilder import CityBuilder

cityBuilder = CityBuilder(attributes=['softstory','occupancy','roofshape'],
                numBldg=100,random=False, bbox=[37.872187, -122.282178,37.870629, -122.279765],
                GoogleMapAPIKey='put-your-API-key-here',
                overwrite=True)

BIM = cityBuilder.build()
```

https://colab.research.google.com/drive/1zspDwK-rGA1gYcHZDnrQr_3Z27JL-ooS?usp=sharing

https://colab.research.google.com/drive/1tG6xVRCmDyi6K8TWgoNd_31vV034VcSO?usp=sharing