FROM TROLLEY STOPS TO WALKABLE BLOCKS:

MACHINE LEARNING INSIGHTS INTO SAN DIEGO'S URBAN FABRIC

GPEC 447 — DATA SCIENCE APPROACHES TO SPATIAL ANALYSIS
SCHOOL OF GLOBAL POLICY AND STRATEGY, UC SAN DIEGO

DANIEL PRYKE
DREW SUTHIPONGCHAI
PUTRA AZHAR

WHY DON'T WE WALK MORE?

No place to rest

Boring scenery

Too far to walk

Too hot!

Unsafe

Poor sidewalk

Pollution

Macro Level Features

- Poor Public Transportation Network
- Multimodal Transfers
- Safety Concerns
- Traffic Congestion and Speed

Micro Level Features

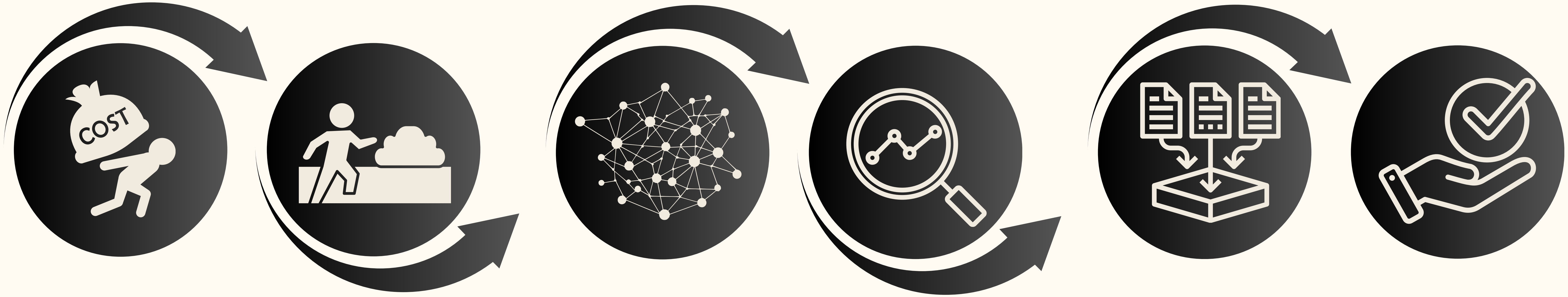
- Tree Canopy
- Quality Pavements
- Street Furniture
- Lighting
- Crosswalks and Pedestrian Signals
- Curb Cuts and Ramp Access
- Visual Interest and Aesthetics

PROBLEM STATEMENT

San Diego has the **potential to enhance its walkability**, especially in areas surrounding public transportation hubs. Currently, only 6.6% of households have no vehicle.

The San Diego Association of Governments could benefit from implementing a tool designed to assess **the micro level features** in the city-design -> which would could then aid in improving walkability of the city.

ANALYSIS PROCESS



Identify Area of Analysis

Identify area of analysis using **network analysis** indicating walking time to the nearest trolley stop. Network analysis gives us polygons to pinpoint areas to draw images from for further analysis

Walkability Criteria

Define a comprehensive set of walkability criteria based on literature and best practices in urban planning

Deep Learning Model

Utilized DeepLab’s model on trained Cityscapes images according to their predefined annotated attributes. Apply the trained model to a broader dataset from additional roads surrounding the trolley stations.

Analysis of Result

Compile and analyze the results from the deep learning model to pinpoint specific issues contributing to poor walkability.

Integration of Walkability Data

We integrate the result into cost distance analysis, prioritizing area closer to the station that needs improvement

Policy Recommendations

Provide detailed, specific recommendations for each prioritized area around trolley stations. As well as how the model can be adapted to work with other areas.

1) IDENTIFY AREA OF INTEREST: NETWORK ANALYSIS

We conduct a walking distance network analysis to **understand the scope of the <u>trolley station's catchment areas.</u>**

The resulting polygons display walking distances across 3-time-based categories:

- 5-min walking
- 10-min walking
- 15-min walking

Trolley Service Areas

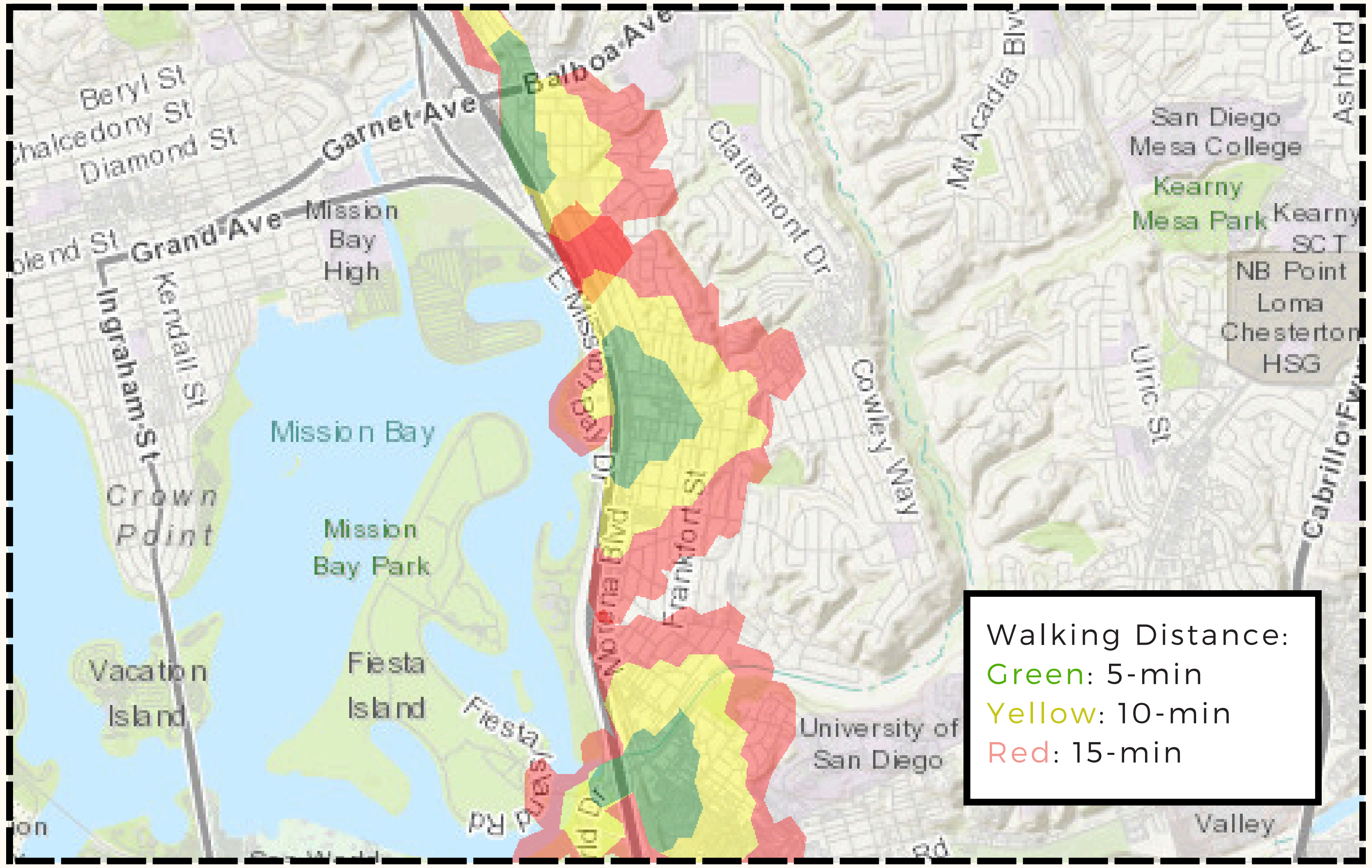


Figure 1: Network Analysis (in walking distance)

2) SCRAPE GOOGLE STREET VIEW IMAGERIES

Using the polygons produced by the network analysis, we buffer and select the areas to download the street-level images through Google API.

Using the Google Static Street View API, we gathered over ***1,500 street images**

- 3 Stations
- Over 400 points sampled
- 4-cardinal directions per point

GOOGLE API IMAGERIES

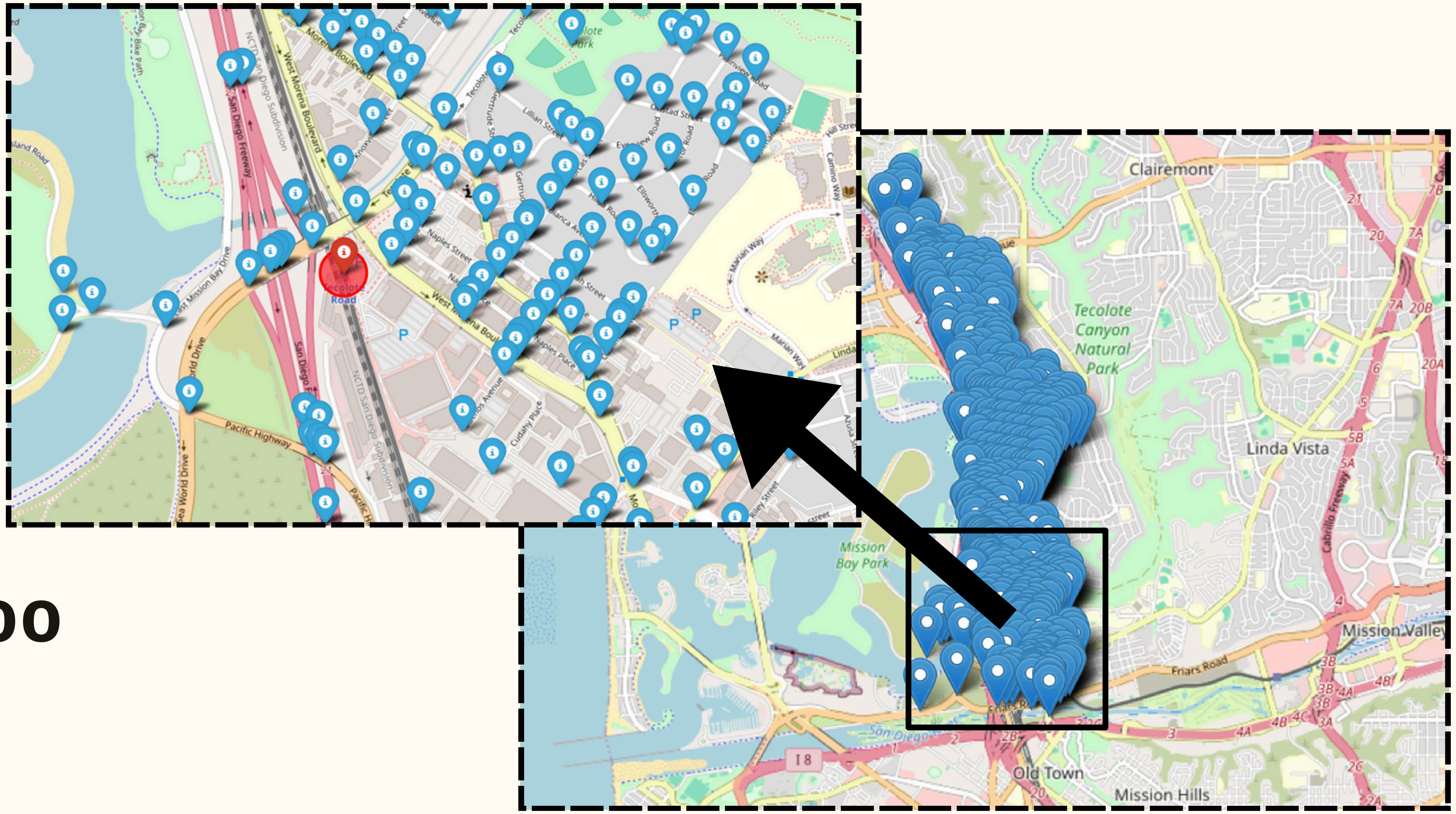
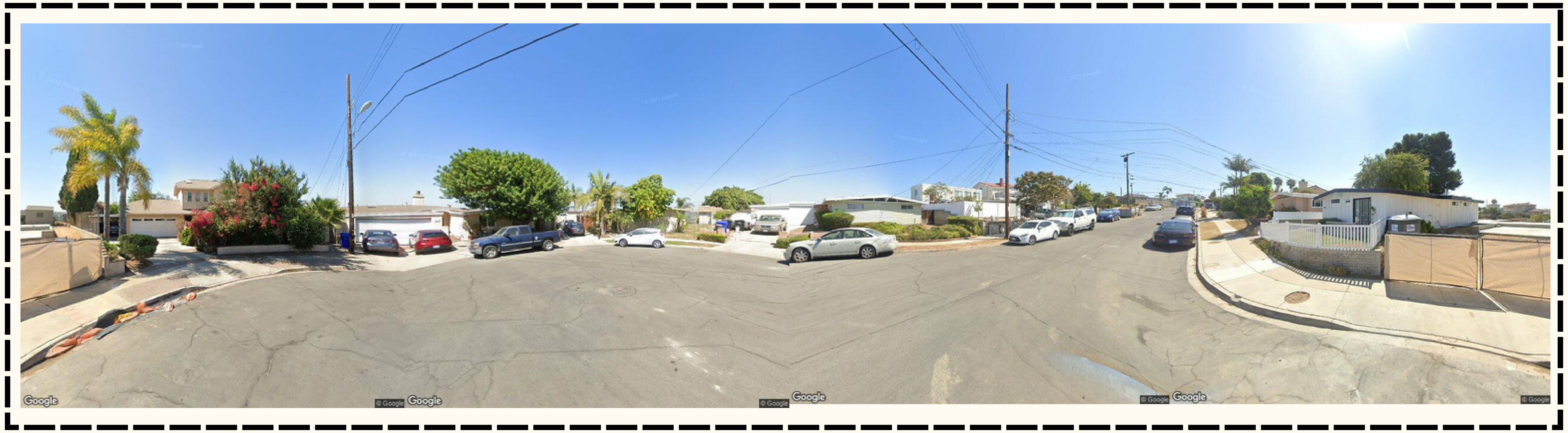
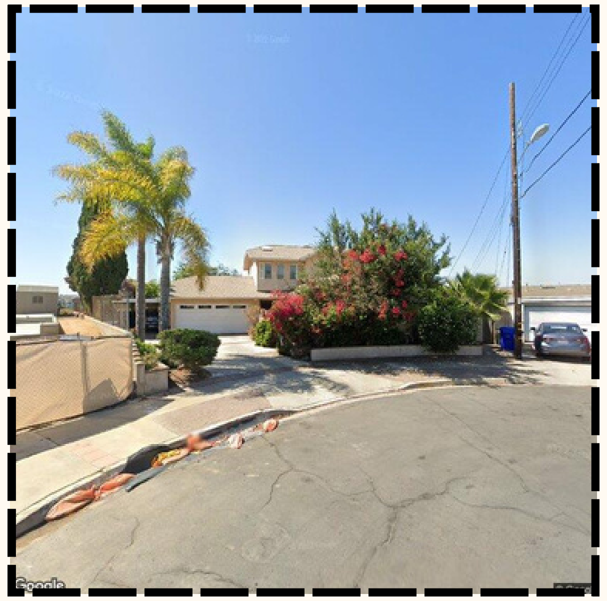


Figure 2: Google Street View API Imageries
Red Point: Tecolote Road Trolley Station

*Approximately

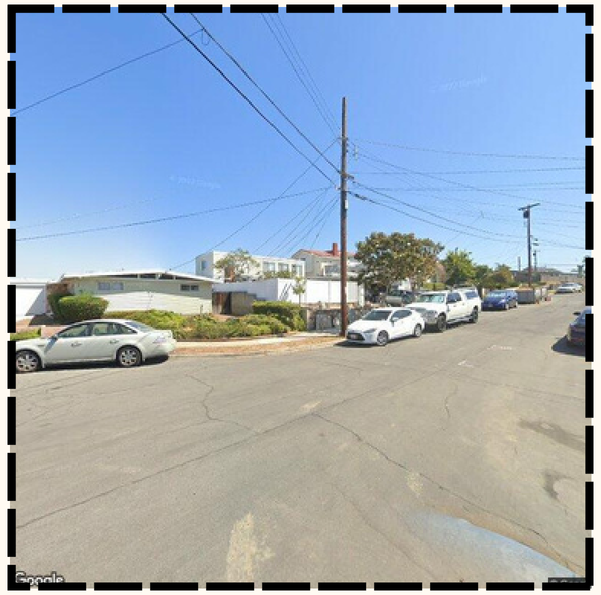
2.1) GOOGLE API INPUT IMAGERIES



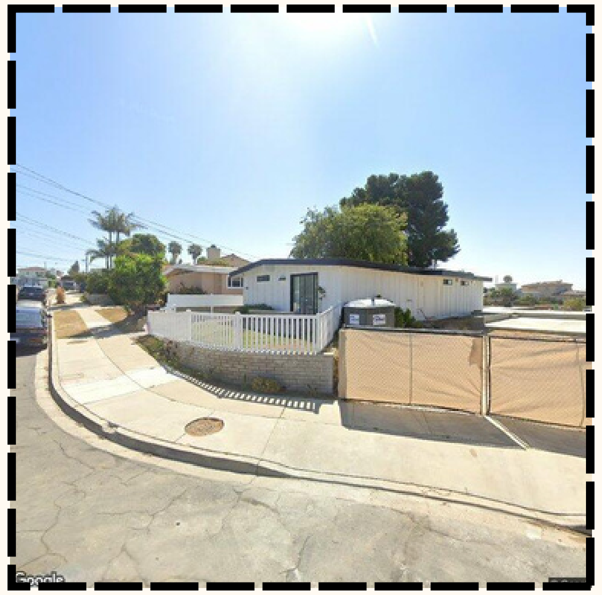


South

West



North



East

3) DEEP LEARNING MODEL 1/4

We leveraged the framework that was developed by MIT. In which they utilized:

Deep Learning Algorithm: DeepLabv3

Training Dataset:

- Microsoft COCO: Common Object in Context.
- Cityscapes Urban Scenes

MIT Deep Learning Framework (GitHub)

4) DEEP LEARNING MODEL 2/4

The model allows us to classify various components within our images, such as trees, pavement, and other urban elements. Using this framework, we can identify and analyze the elements that impact walkability in our area of focus.

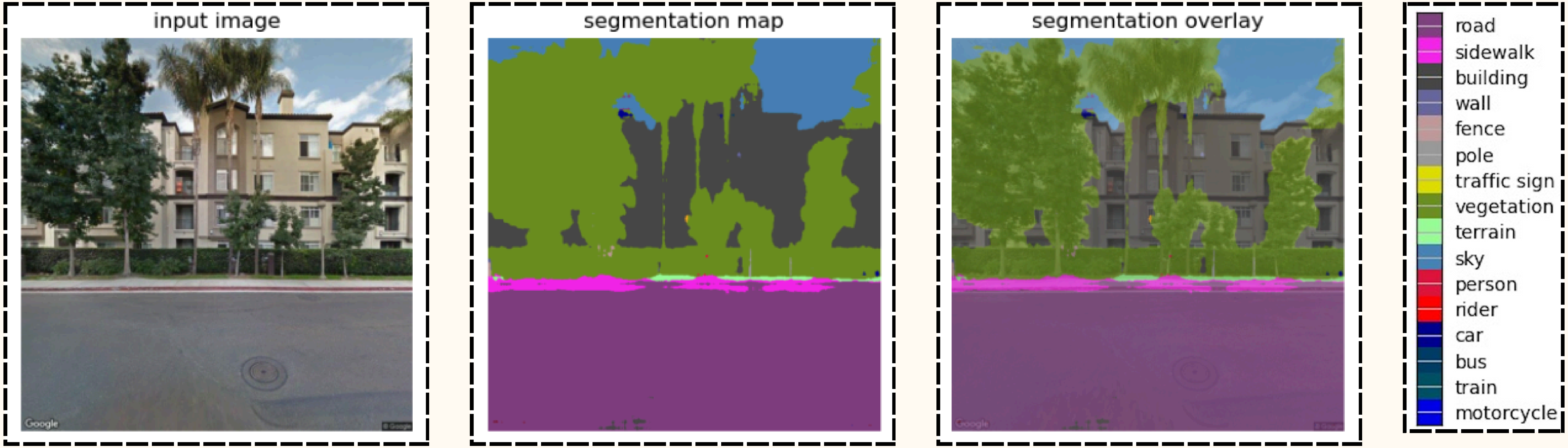


Image Source: Google API

5) DEEP LEARNING MODEL 3/4

- We apply the weighting system outlined in Donghwan et al.'s paper, where the weights were determined using a Random Forest algorithm. This method allows us to create the micro-level index dataset.

A novel walkability index using google street view and deep learning

Donghwan Ki [a], Zhenhua Chen [b], Sugie Lee [c,*], Seungjae Lieu [d]

[a] PhD candidate, Department of City and Regional Planning, The Ohio State University, 275 W Woodruff Ave, Columbus, OH 43210
[b] Associate professor, Department of City and Regional Planning, The Ohio State University, 275 W Woodruff Ave, Columbus, OH 43210
[c] Professor, Department of Urban Planning & Engineering, Hanyang University, Science & Technology Bld. 303-1, Hanyang University, 222, Wangsimni-ro Seongdong-gu, Seoul, 04763, Korea
[d] PhD Student, School of City and Regional Planning, Georgia Institute of Technology, 245 4th St NW, Atlanta, GA 30313

Table 3
Micro-level Features

Variables	Descriptions
Street greenery	% of street greenery (% vegetation + % terrain)
Visual enclosure	100% - % of sky
D/H ratio	% building / (% of sidewalk + % of road)
Obstacles	% of wall + % of fence
Visual complexity	Amount of information in an image
Sidewalk	% of sidewalk
Slope	% of slope

Table 4
Variable Importance and Rescaled Weights

Variable	Importance	Weight (rescaled)	Direction (+ or -)
Walk-score	27.536	1.000	-
Obstacles	32.130	1.167	-
Sidewalk	41.110	1.493	+
Slope	42.862	1.557	-
Visual complexity	42.960	1.560	+
D/H ratio	44.665	1.622	+
Visual enclosure	47.624	1.730	+
Street greenery	51.273	1.862	+

6) DEEP LEARNING MODEL 4/4

The final outcome of this method is a Micro-level index dataset, which provides a detailed assessment of walkability factors.

These micro-level indexes include:

- Street Greenary
- Visual Enclosure
- Percentage of Sidewalk
- Building to Road Ratio
- etc.

Micro-Level Index Dataset

Individual Images Index:

```python
from IPython.display import display

display(gdf)
```

88] ✓ 0.0s Python

lat	lon	adjusted_street_greenery	adjusted_visual_enclosure	adjusted_dh_ratio	adjusted_obstacles	adjusted_visual_complexity	adjusted_sidewalk	geometry
874007	-117.215030	40.860960	172.458108	2.219604	0.0	15.60	15.268379	POINT (-117.21503 32.87401)
.874115	-117.216006	35.620599	172.998592	2.624066	0.0	14.04	29.373147	POINT (-117.21601 32.87411)
.874115	-117.216006	31.800833	172.712371	1.916905	0.0	15.60	23.568939	POINT (-117.21601 32.87411)
874007	-117.215030	27.259783	172.840628	1.181355	0.0	15.60	21.552757	POINT (-117.21503 32.87401)
874007	-117.215030	37.577306	172.960720	2.142245	0.0	17.16	29.298302	POINT (-117.21503 32.87401)
.874115	-117.216006	21.727122	172.683509	1.247879	0.0	15.60	23.664560	POINT (-117.21601 32.87411)
.874115	-117.216006	32.861845	172.969871	1.864852	0.0	15.60	20.272505	POINT (-117.21601 32.87411)
874007	-117.215030	9.409404	172.953399	0.560129	0.0	15.60	15.209087	POINT (-117.21503 32.87401)

Combined Panoma Index:

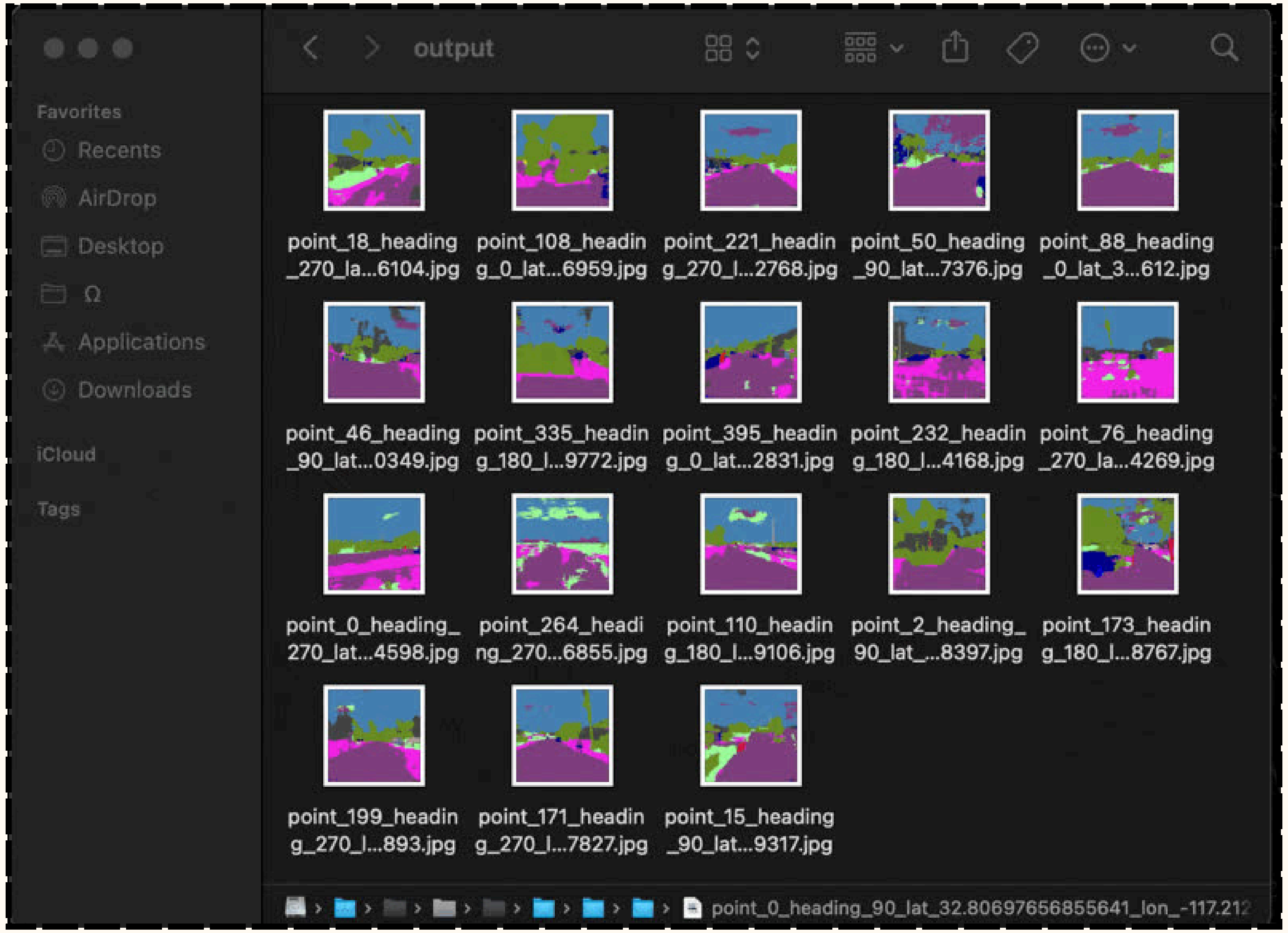
```python
from IPython.display import display

display(combined_gdf)
```

94] ✓ 0.0s Python

	geometry	adjusted_street_greenery	adjusted_visual_enclosure	adjusted_dh_ratio	adjusted_obstacles	adjusted_visual_complexity	adjusted_sidewalk
0	POINT (-117.21601 32.87411)	30.502600	172.841086	1.913426	0.0	15.21	24.219788
1	POINT (-117.21503 32.87401)	28.776863	172.803214	1.525833	0.0	15.99	20.332132

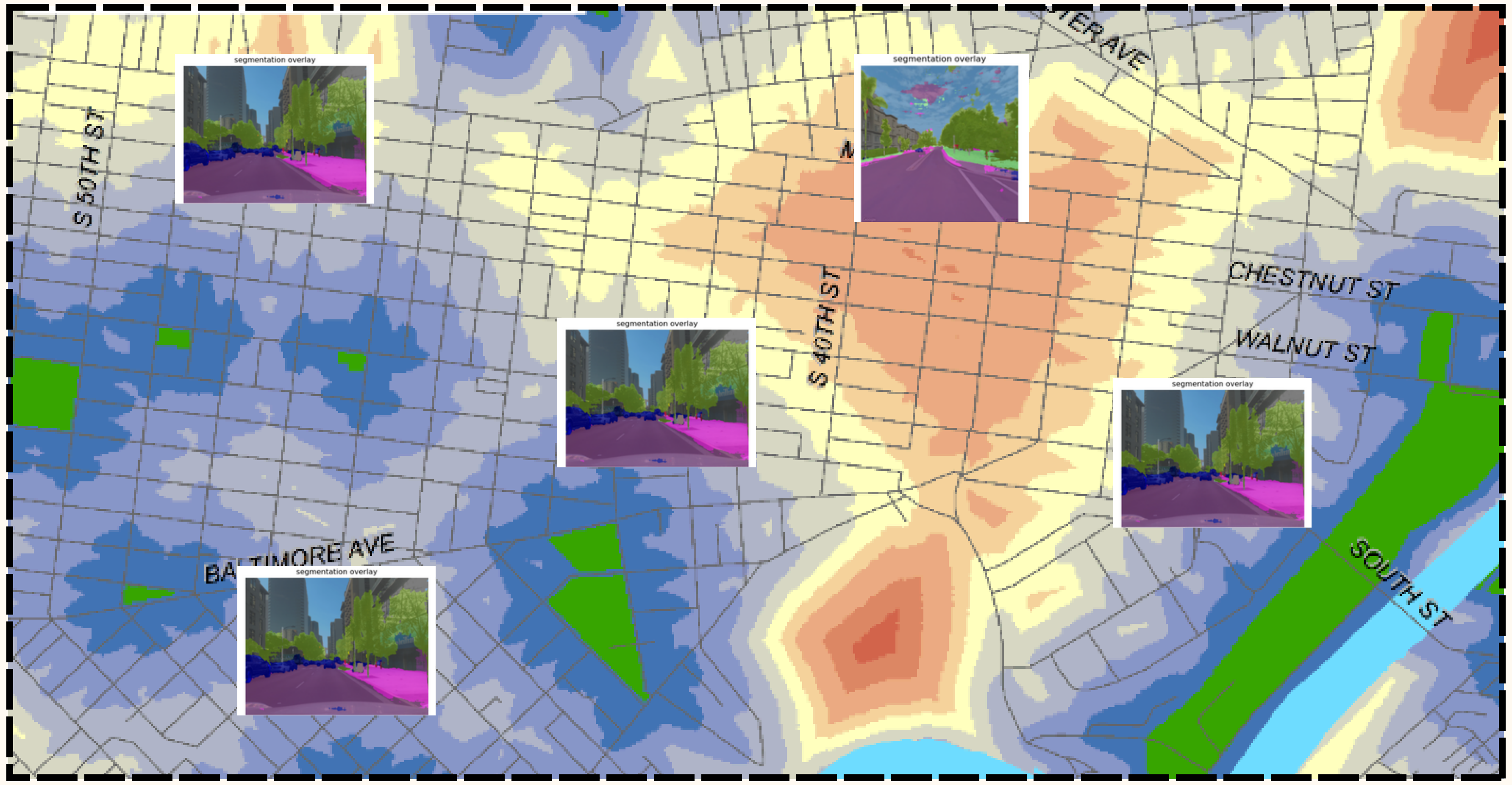
FOR NOW...

The model is still running at this very moment...



RESULT AND DISCUSSION

Upon retrieving the final micro-level indices, we will integrate them into the **cost-distance map.** This integration will **enable us** to **identify areas with low walkability scores, pinpointing those that require further investment.** Additionally, this analysis will help us determine the specific types of walkability issues present in each area, allowing for targeted improvements.

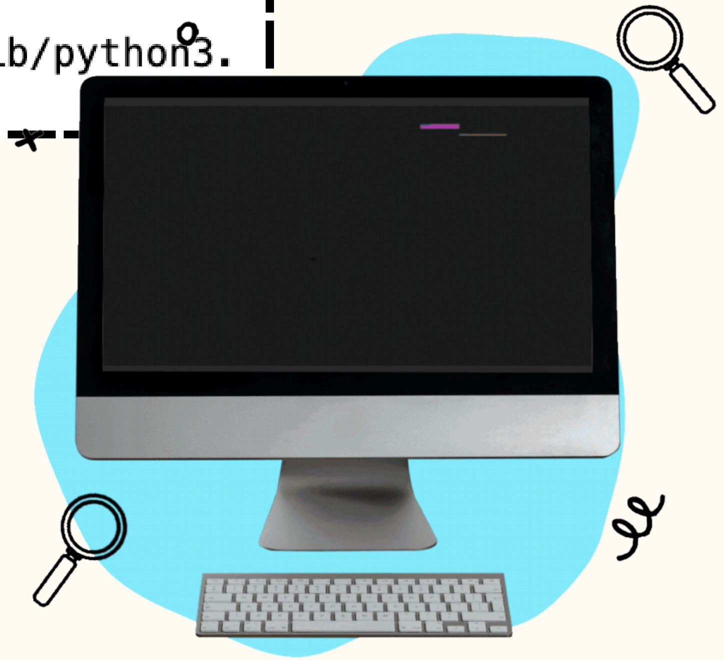


Background Cost - Distance Map Source: https://www.azavea.com/blog/2014/09/04/creating-a-cost-distance-surface-to-measure-park-access/

CAVEATS 1/3

01 **<u>High Computational Power is required</u> to download all available imageries within the area of interest.** Therefore, we restrain our analysis by filtering our imageries to a select sample of images.

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
/tmp/ipykernel_113/4287785095.py in <module>
      4
      5 from arcgis.gis import GIS
----> 6 from arcgis.raster.functions import raster_function_templates
      7 from arcgis.raster.analytics import convert_feature_to_raster, raster_calculator
      8 from arcgis.raster import ImageryLayer

ImportError: cannot import name 'raster_function_templates' from 'arcgis.raster.functions' (/opt/conda/lib/python3.
9/site-packages/arcgis/raster/functions/__init__.py)
```

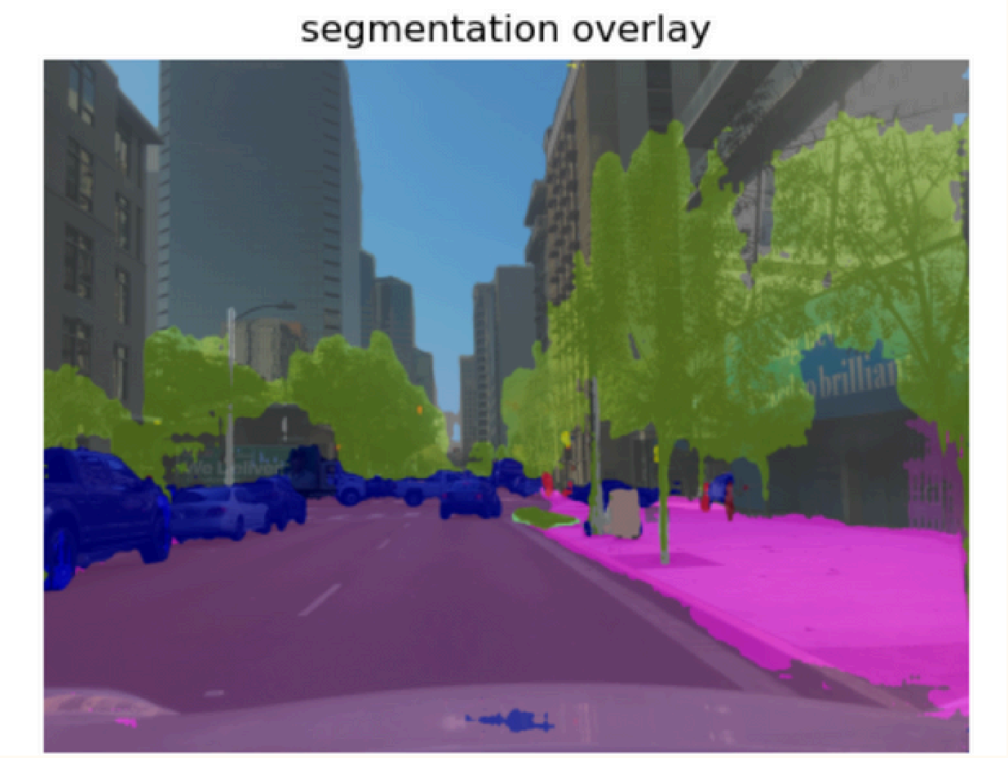


CAVEATS 2/3

02 **Model accuracy is dependent on the quality of image input.**

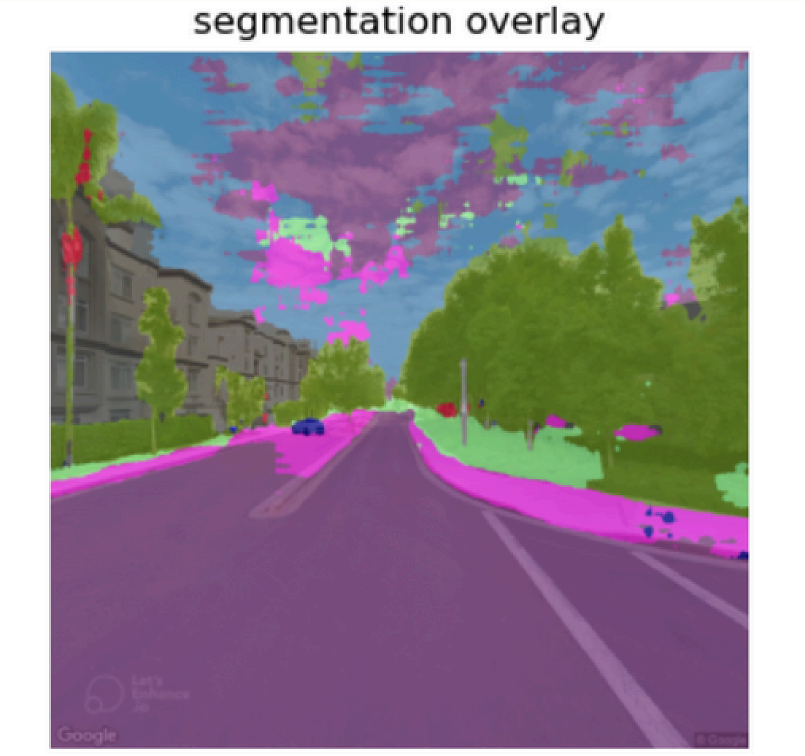
It has proven to work well with good-quality images but not so well with low-quality images. We have tried images from Google API and Mapillary, and the higher quality from Mapillary impacts the accuracy of the results a lot.

Higher Resolution (Mapilary):



2048×1536

Lower Resolution Google API:



640×640

Enhanced 640×640

CAVEATS 3/3

03

<u>We have not established a robust metric to assess the model's accuracy rate.</u> This is because ground-truth level data that annotate semantic segmented images by hand are difficult to gather and also time-consuming.

GIF: https://github.com/lexfridman/mit-deep-learning/tree/master

COMMENTS OR FEEDBACK?