Mapillary An open platform for street-level imagery

How street level imagery helps us understand our world

Meta Marcie Hogan

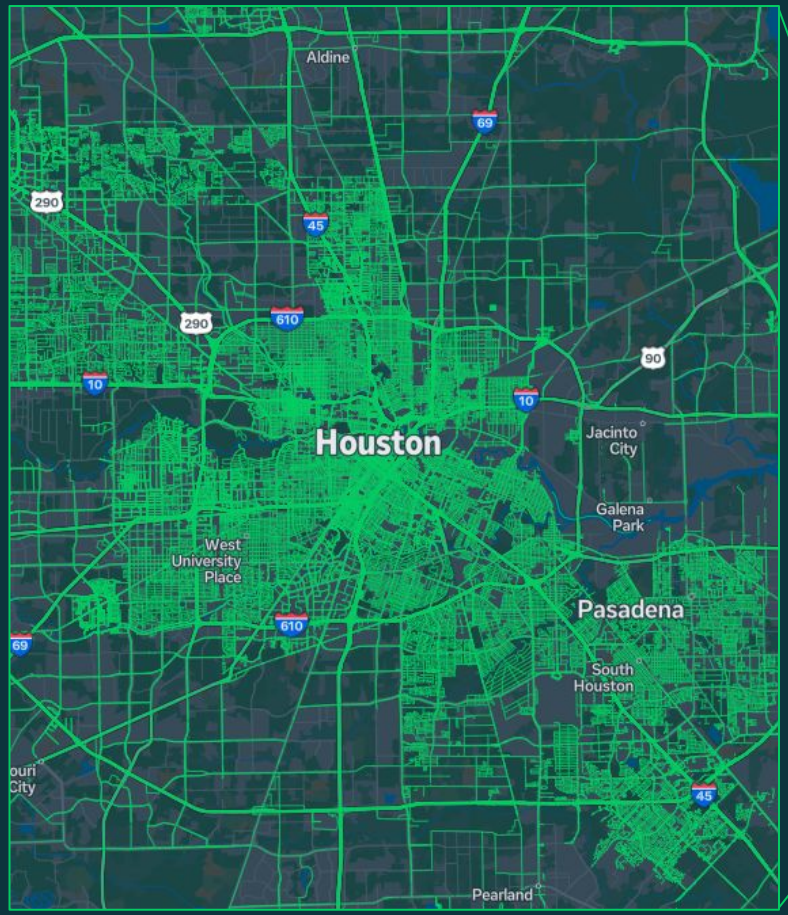
Houston GDW 2/5/25

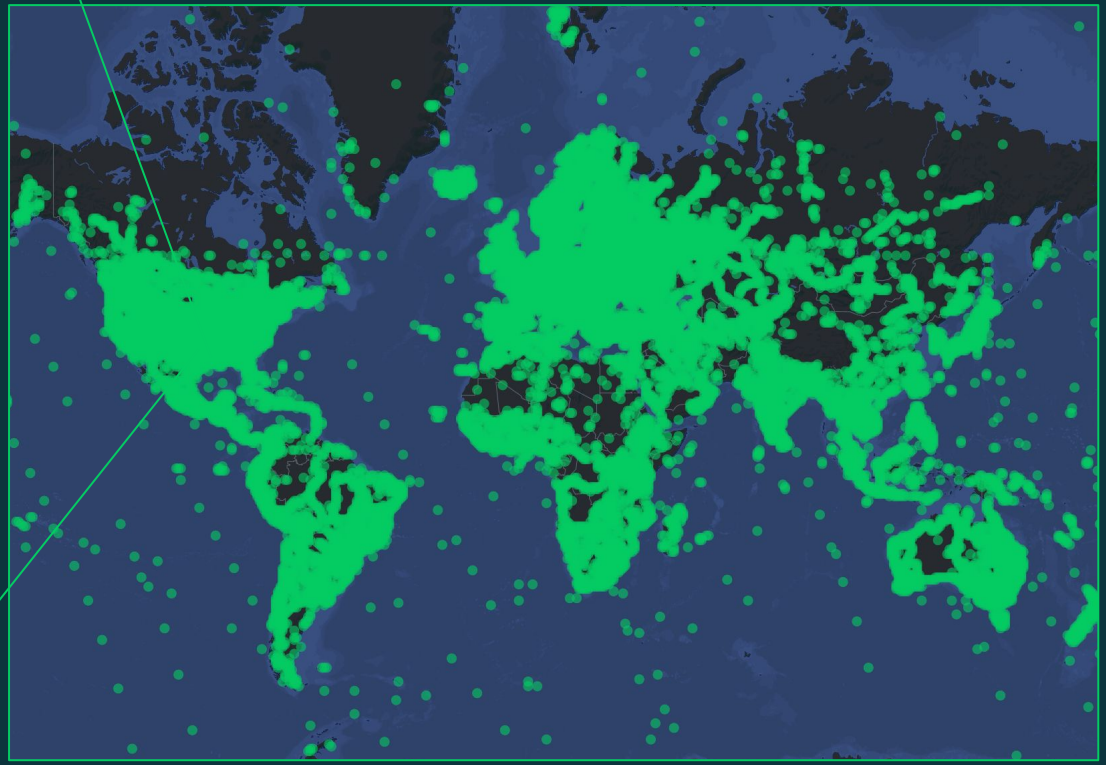
Mapillary overview

A platform for street-level imagery, powered by *collaboration* and *computer vision*

Some facts...

Over 2.4 billion images globally

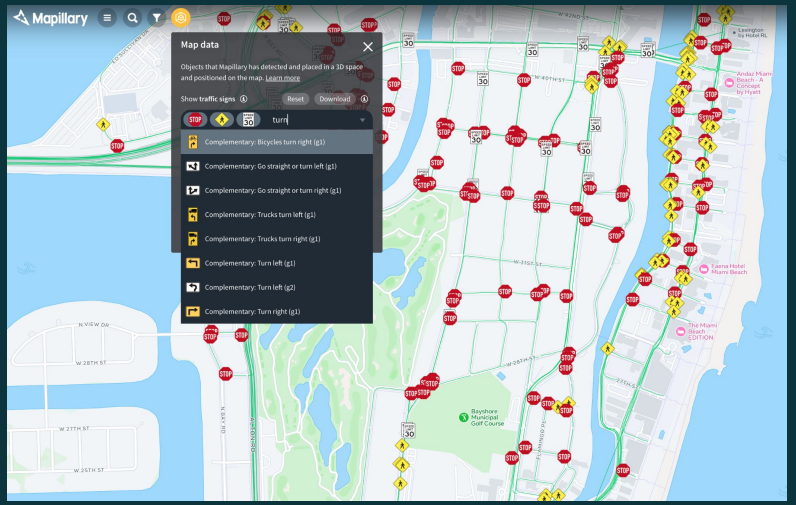




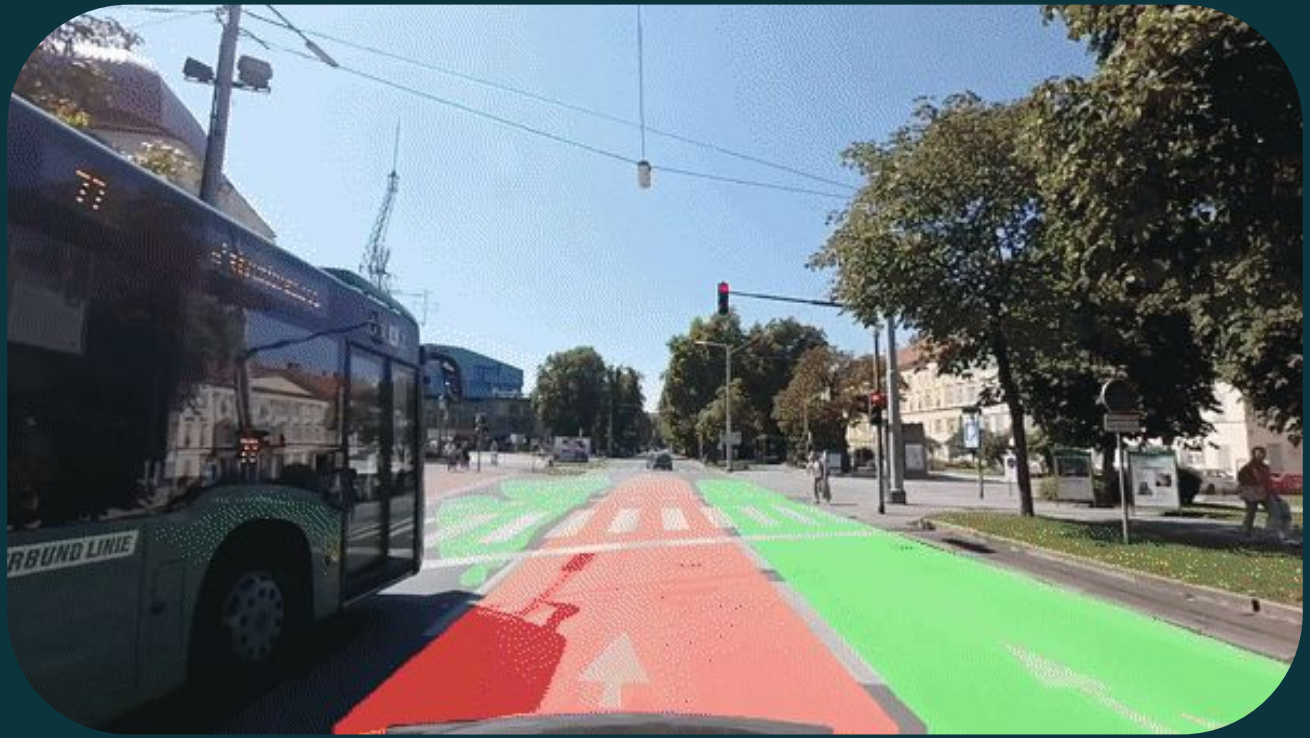
Some more facts!

- 9M+ miles of coverage
- 42 object classes for points
- 1,500 traffic sign classes

- Mapillary joined Meta in 2020

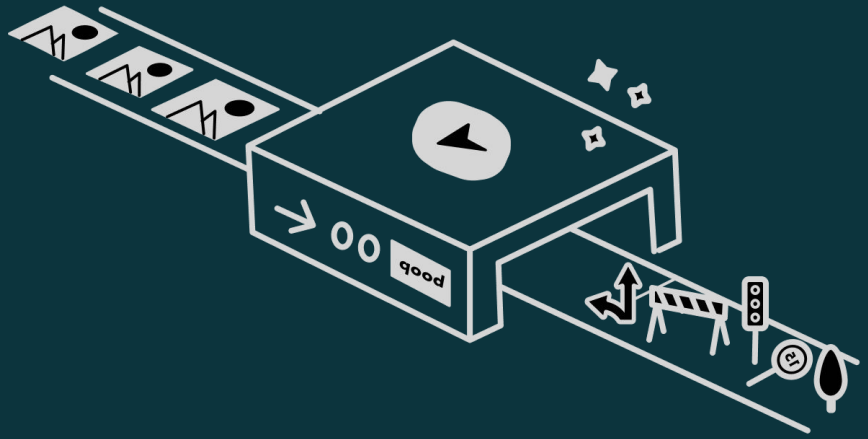


We use computer vision...

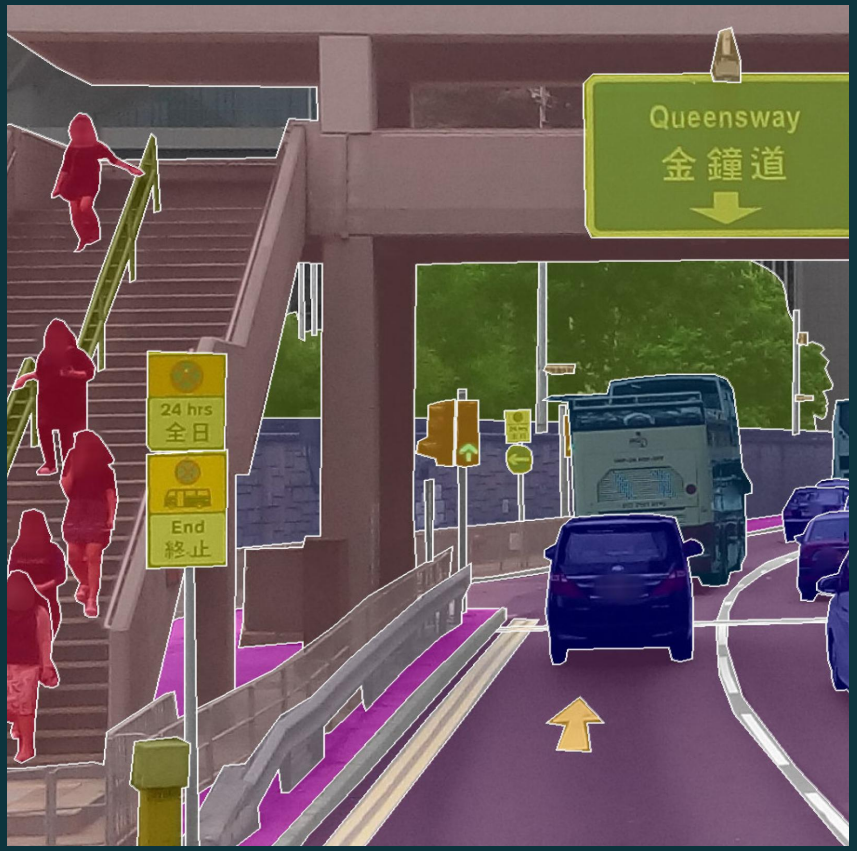


To create open data!

Street Imagery



Map Data



We believe in collaboration...

…to make better maps!

CC BY-SA

maptiler

amazon

Meta

HOT Humanitarian OpenStreetMap Team

MapLibre

THE WORLD BANK

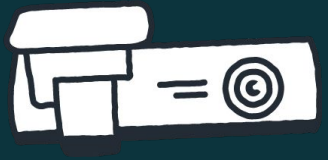
American Red Cross

How does Mapillary work?

Any camera, anywhere

Smartphone

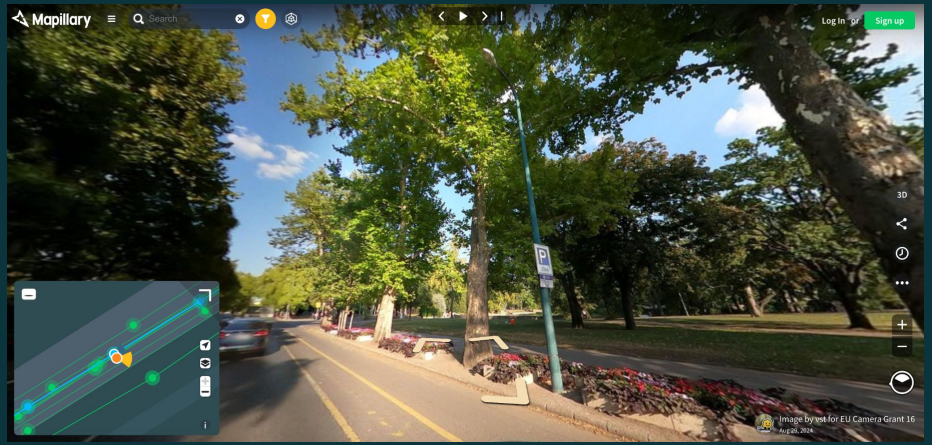
Action cam



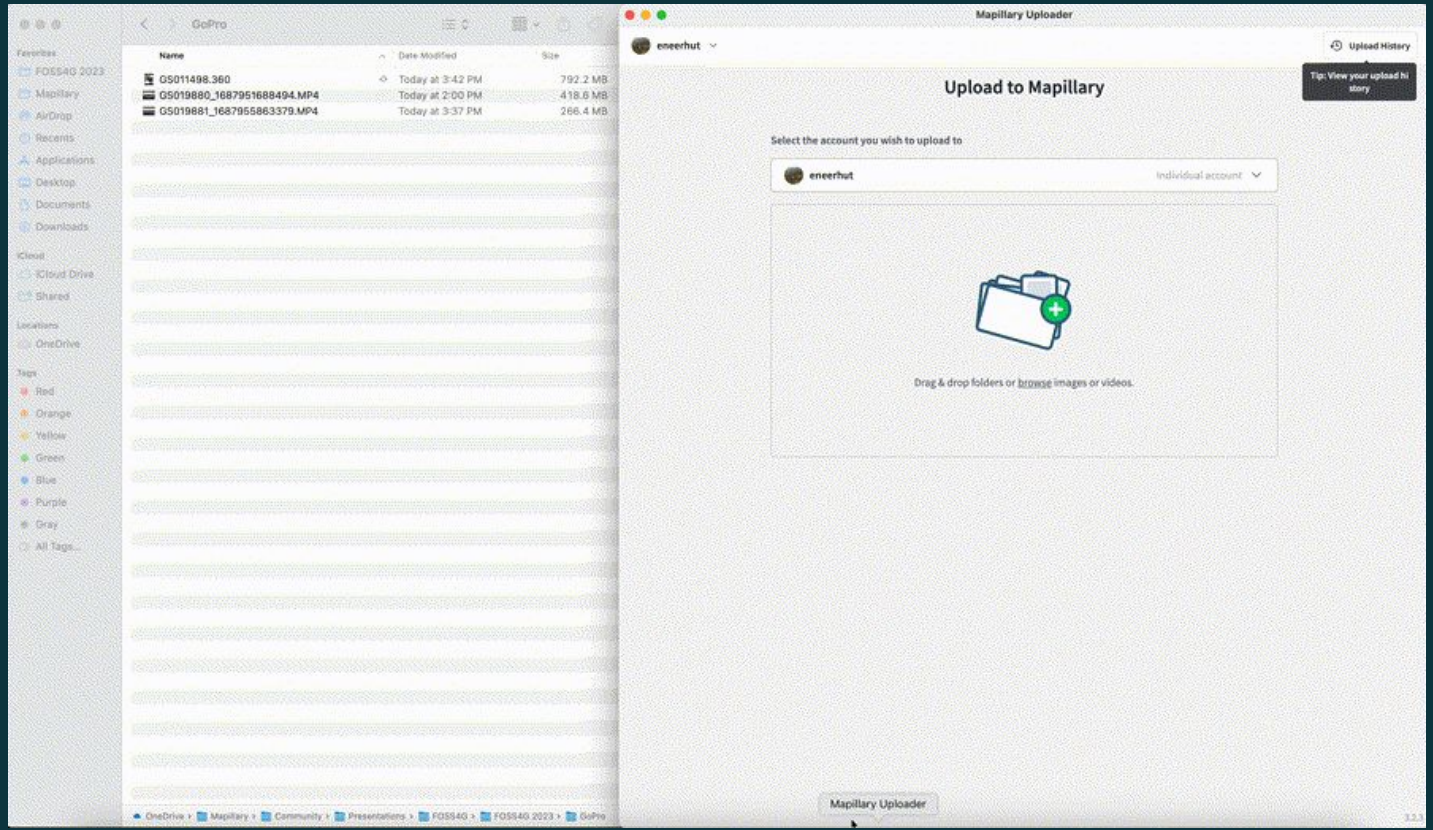
Dashcam

360 cam

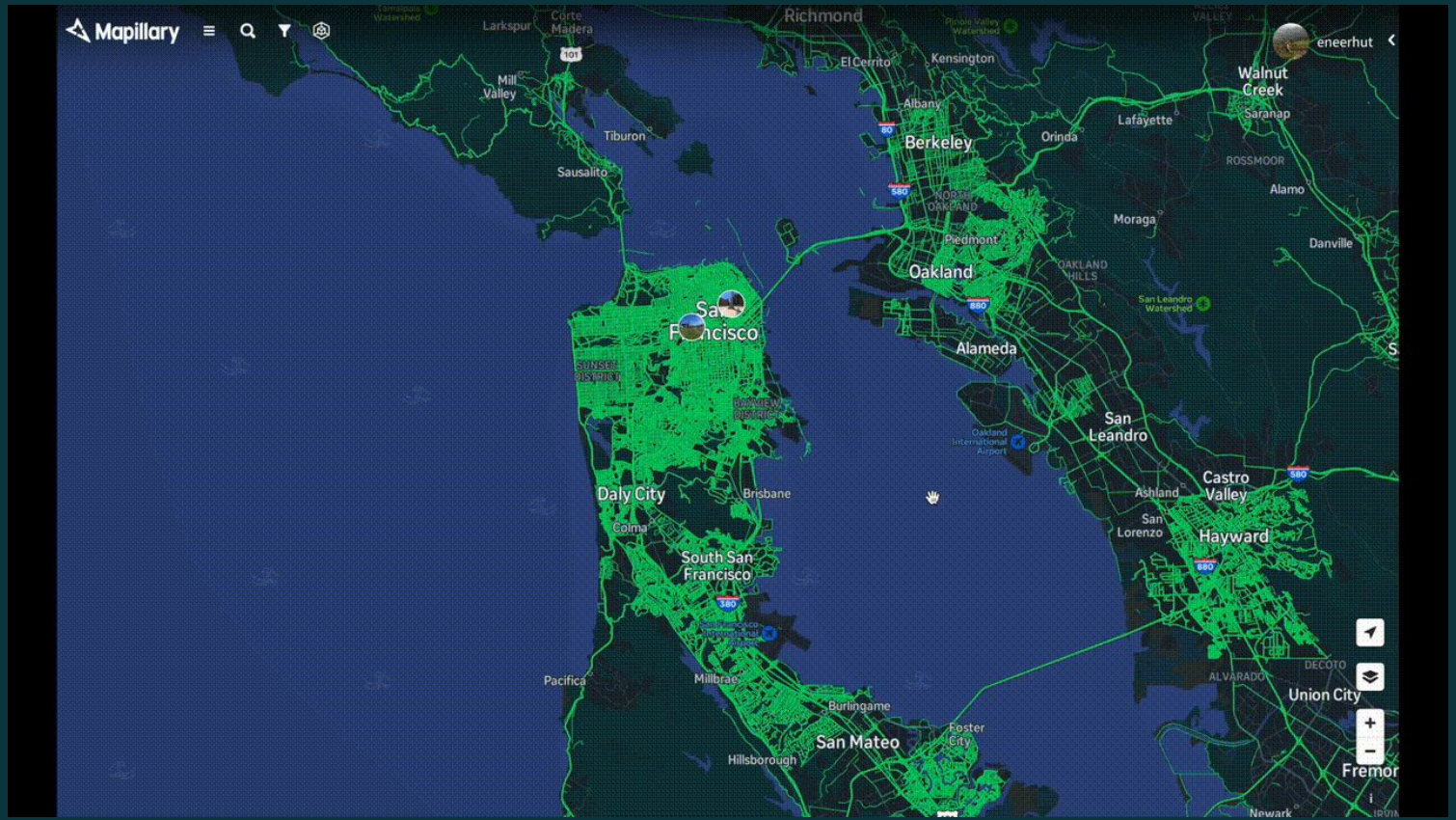
GoPro MAX: Our favorite camera



Desktop Uploader



View your organization's imagery



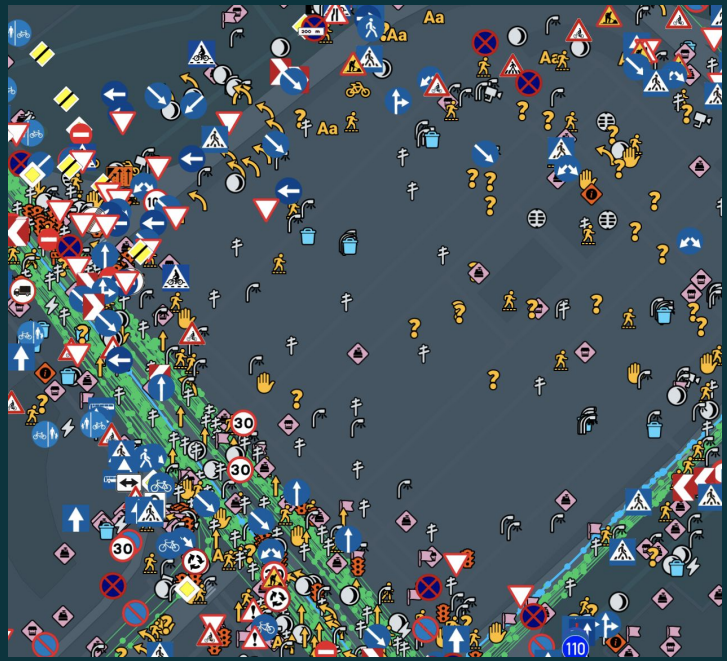
Implementing Mapillary data

Available data types

- Object detections
  - A grouping of pixels in a photo believed to be a part of the same object
  - Created via semantic segmentation
  - 150+ object detections available to query, download, etc. with the Mapillary API

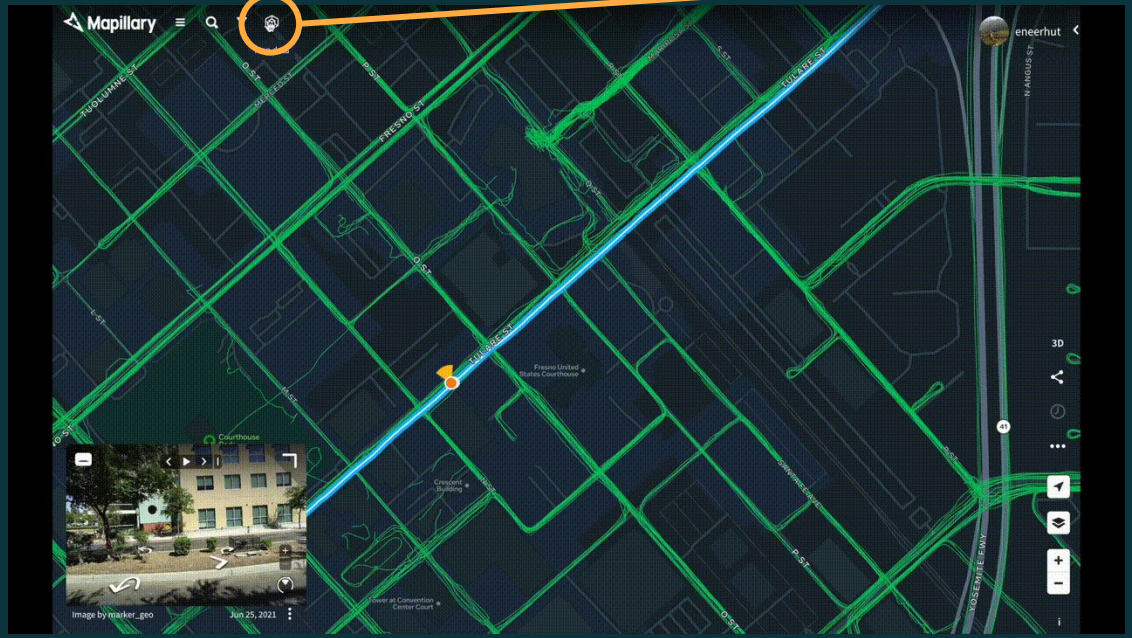
Available data types

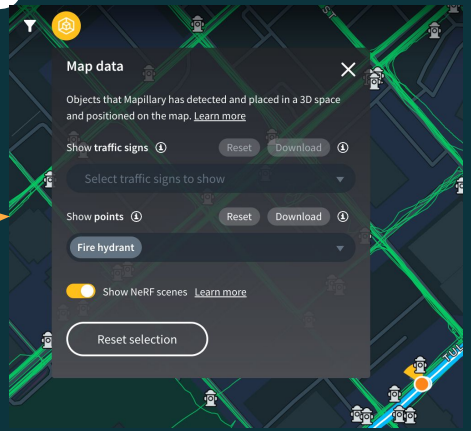
- Map features
  - 40+ object types
  - 1,500+ street sign classes
  - Generated via matching object detections across 3+ images
  - Coordinates are assigned via triangulation
  - Great for asset management



Data download options

1. Web interface (mapillary.com/app/)



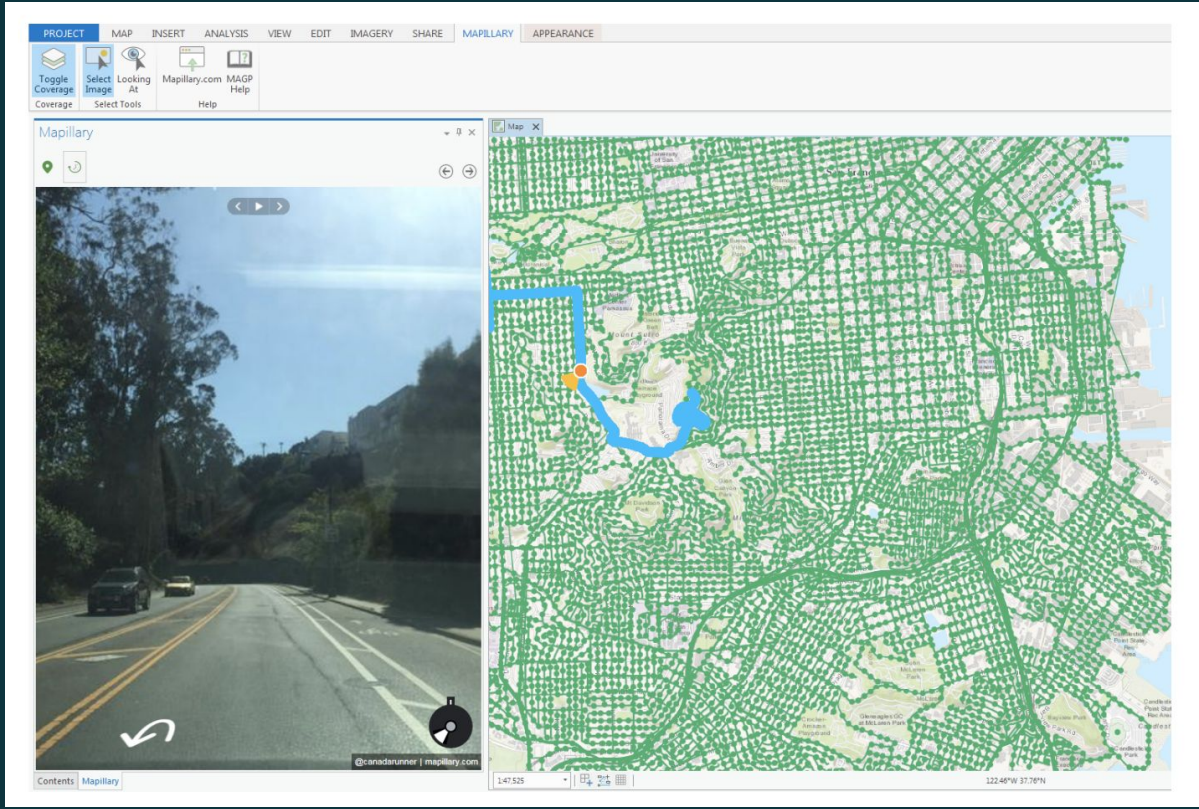


Data download options

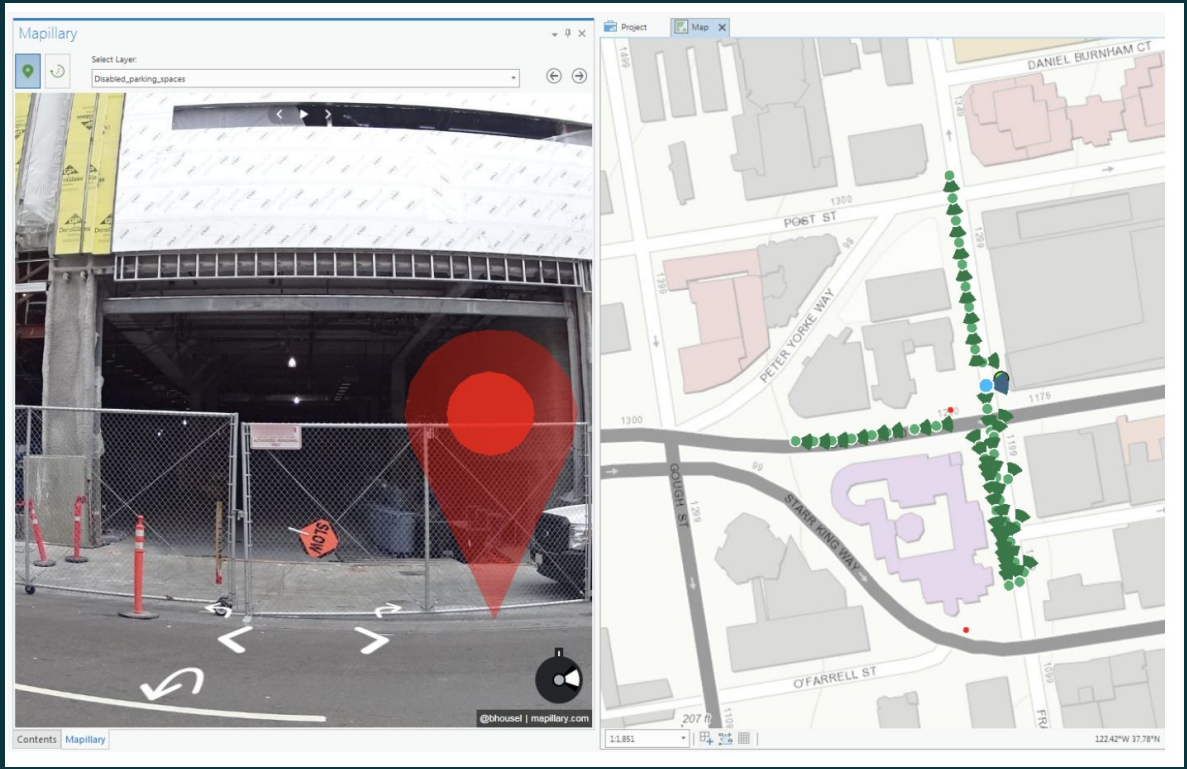
2. API + Python SDK (mapillary.com/developer/api-documentation)

```python
# Make sure to install Mapillary's Python SDK (https://mapillary.github.io/mapillary-python-sdk/) using: pip install mapillary. You may need to install with version Python 3.8 to ensure the dependencies work correctly.
import mapillary.interface as mly
import requests
import json
import geojson

# Your token here!
# To get one, visit https://www.mapillary.com/dashboard/developer, go to 'developers',
# Then 'register application', register a new application (read access atleast),
# then copy & paste the 'Client Token' here
MLY_ACCESS_TOKEN = 'YOUR_ACCESS_TOKEN'

mly.set_access_token(MLY_ACCESS_TOKEN)

# Define your area of interest.
bbox = { 'west': -61.166464989667475, 'south': 13.67576664580504, 'east': -60.813735924979966, 'north': 14.144804377935927 }

# Define the traffic signs or point features you're interested in.
    # Traffic sign list: https://www.mapillary.com/developer/api-documentation/traffic-signs
    # Point feature list: https://www.mapillary.com/developer/api-documentation/points.

traffic_signs = ('regulatory--stop--g1','regulatory--stop--g2','regulatory--stop--g3','regulatory--stop--g4','regulatory--stop--g5','regulatory--stop--g6','regulatory--stop--g7',
'regulatory--stop--g8','regulatory--stop--g9','regulatory--stop--g10','regulatory--stop-signals--g1','regulatory--stop-signals--g2')

point_data = mly.traffic_signs_in_bbox(bbox,traffic_signs)

# Parse the input JSON string into a dictionary
data = json.loads(point_data)

# Convert the dictionary to a GeoJSON FeatureCollection object
feature_collection = geojson.loads(point_data)
# Print the resulting GeoJSON object
print(feature_collection)

# Write feature_collection to a GeoJSON file on Google Drive
with open('/Users/eneerhut/Downloads/saint_lucia/traffic_signs.geojson', 'w') as f:
    geojson.dump(feature_collection, f)
print('Your data has been downloaded.')
```

Mapillary ArcGIS Pro add-in



Mapillary ArcGIS Pro add-in



Mapillary for Developers

```
Curl -H "Authorization OAuth: [YOUR_CLIENT_ID]" https://graph.mapillary.
com/1933525276802129?fields=id,captured_at,compass_angle,sequence,geometry
```

Mapillary API

Coverage Tiles (.mvt)

MapillaryJS

(Street-level imagery viewer)

Mapillary Python SDK

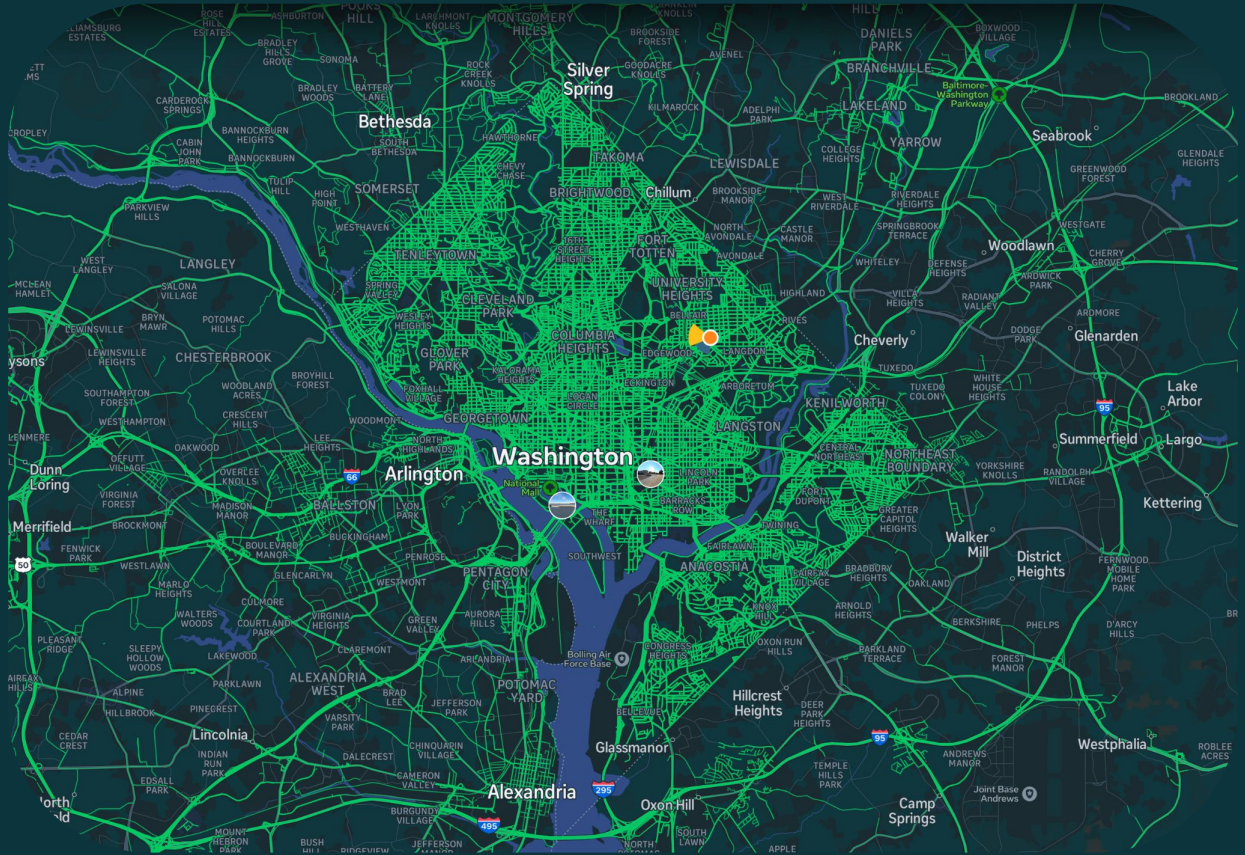
Case studies

Washington D.C. Dept of Public Works
Fresno County, California
City of Detroit, Michigan

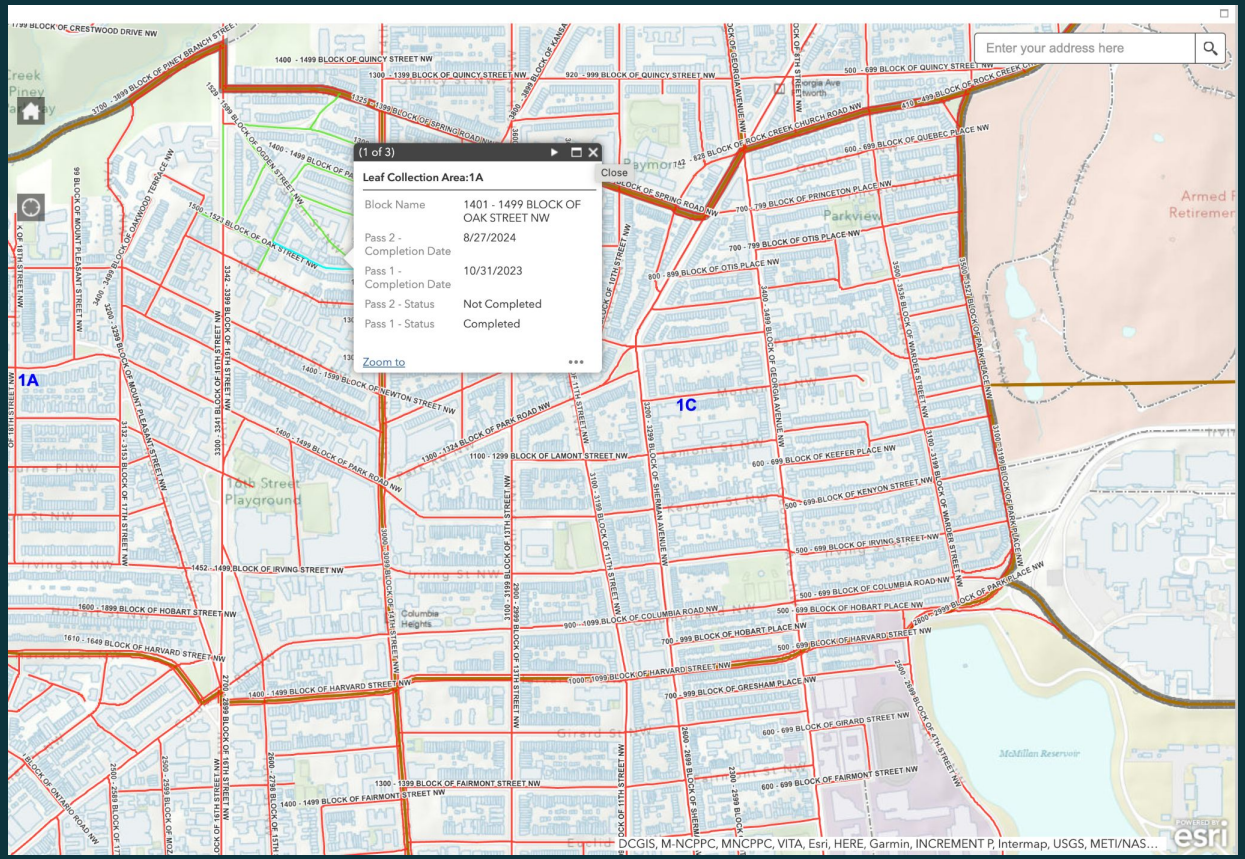
D.C.’s Department of Public Works

Fall leaf collection

D.C.’s Department of Public Works

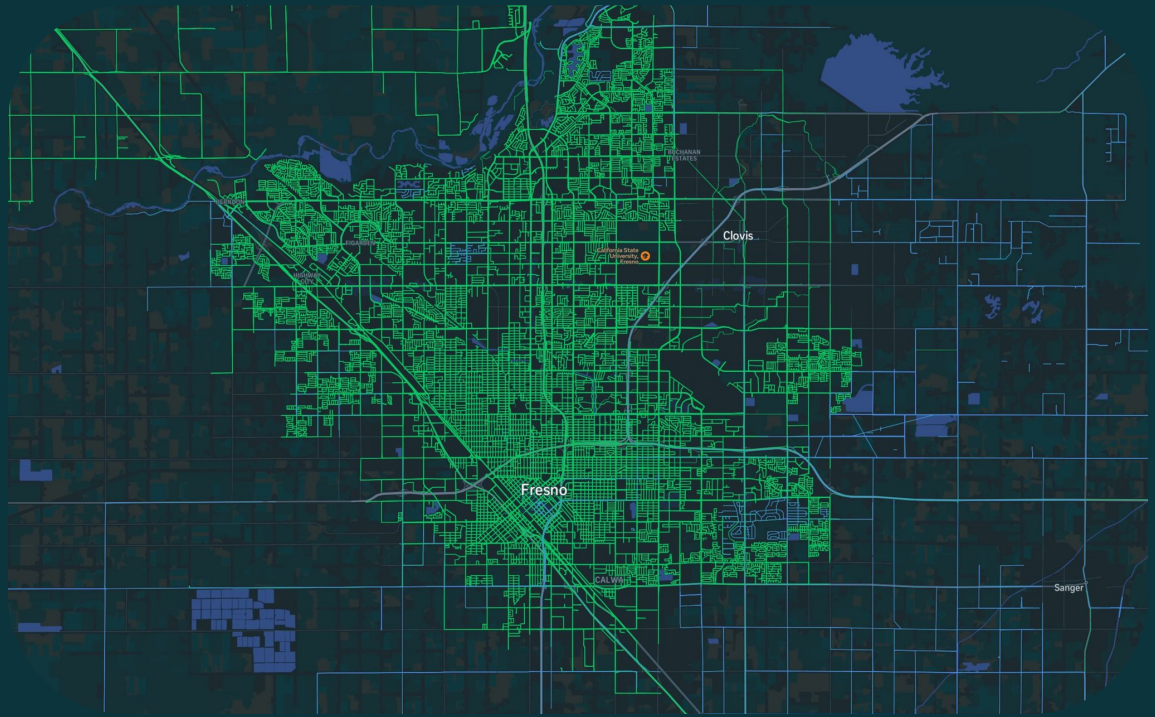


D.C.’s Department of Public Works



Fresno County

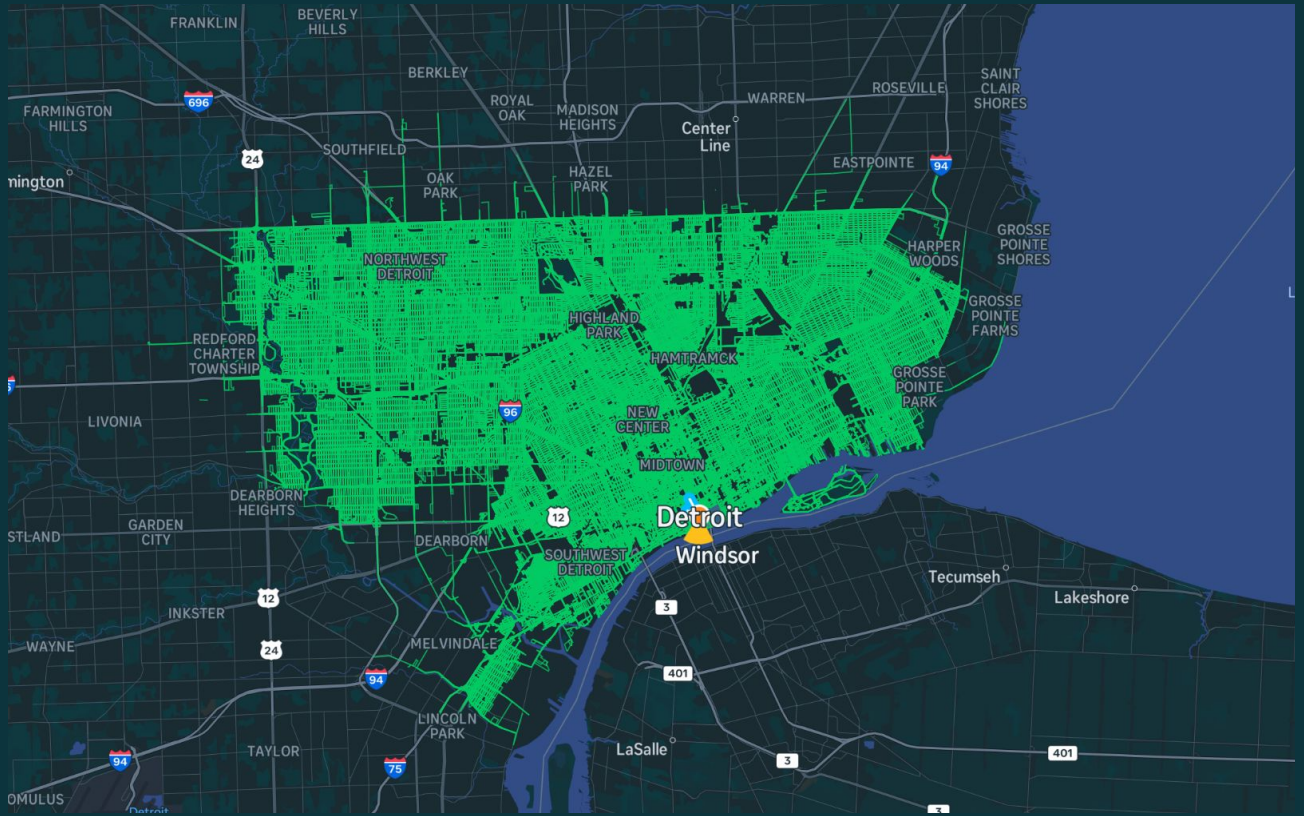
Managing critical infrastructure



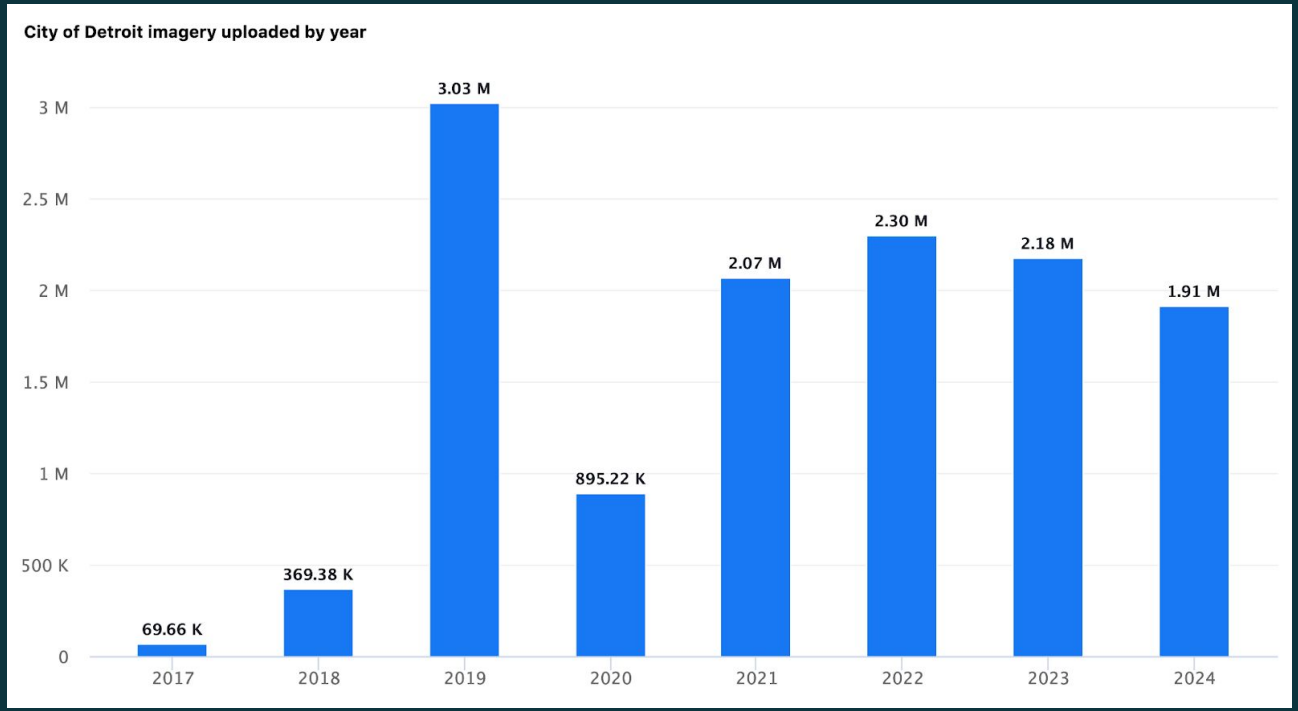
Fresno County

Fresno County

City of Detroit



City of Detroit



Quick facts on Detroit:

- Started contributing to Mapillary routinely in 2018
- **99.8% of all public roads mapped**, most many times over (Taking the 👑 for most mapped city on Mapillary!)
- Awarded a $400,000 grant to upload their capture rig in 2020
- Two full time drivers employed by the city
- Working in-house on their own CV models for Detroit-specific objects
- Used Mapillary to **challenge the 2020 census! (they won)**

City of Detroit

Why should I use Mapillary?

- **100% free to use!**
- Enables you to take SLI capture into your own hands
- An affordable asset management solution
- Contribute to open data & help others make better maps!

Thank you!

Marcie Hogan

mhogan@meta.com