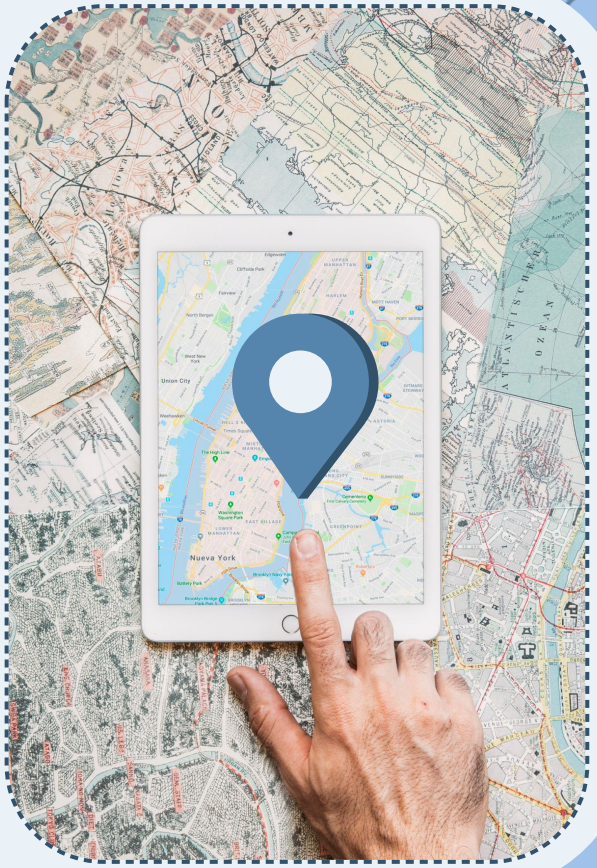
Project E:
Extracting Place Entrances from Mapillary Imagery

Evan Rantala & Julien Howard

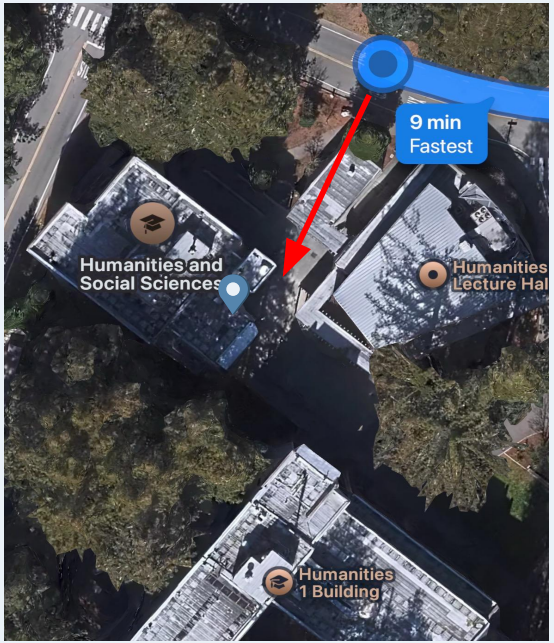
Project Statement

“Automatically extract precise place entrance locations from Mapillary street-level imagery to enhance navigation data.”



Project Description

- Existing datasets lack consistent entrance coordinates
- People want to be routed to a **specific location**
- Entrance points improve **navigation, accessibility, delivery logistics**, etc.
- **Automated** improvement of large-scale open geospatial data



Sponsor Collaboration

- Met with **Said Turksever** (engineer at Meta working on Mapillary) early in the quarter
- Priorities **shifted** - "detect doors" to "correctly associate the detection with the right building wall."
- Early feedback shaped our initial OKRs
- Follow-up communication was conducted via email/linkedin
- Learned the real-world challenges of coordinating with industry partners

Objectives and Key Results

Objective #1

Develop a robust computer vision model for building entrance detection.

Key Results

1. Fine-tune a YOLOv8 model for entrance detection using at least **750** manually labeled street-level images from Mapillary.
2. Achieve **>60%** precision and **>60%** recall on a validation set of urban street images.

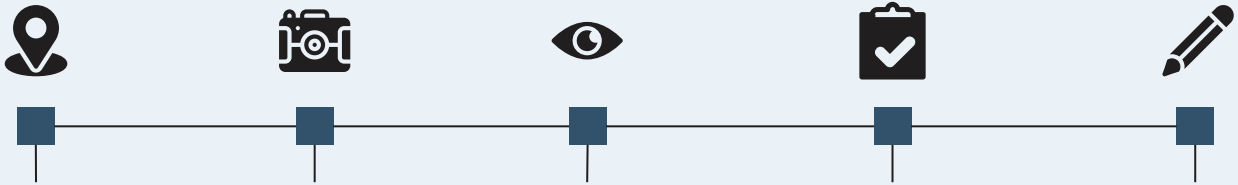
Objective #2

Deliver a pipeline that successfully provides coordinates of building entrances in a pilot area.

Key Results

1. Join datasets to create sets of images with visible entrances within **40 degrees** of image compass angle.
2. Generate building entrance coordinates for a 10–15 block pilot area with **<5m** error

Our Approach



Download Overture Building and Places data within desired area

Fetch Mapillary images taken within area

Infer entrances with detection model on images

Match detected entrances to existing building data

Export validated entrance points to GeoJSON

Git Repository

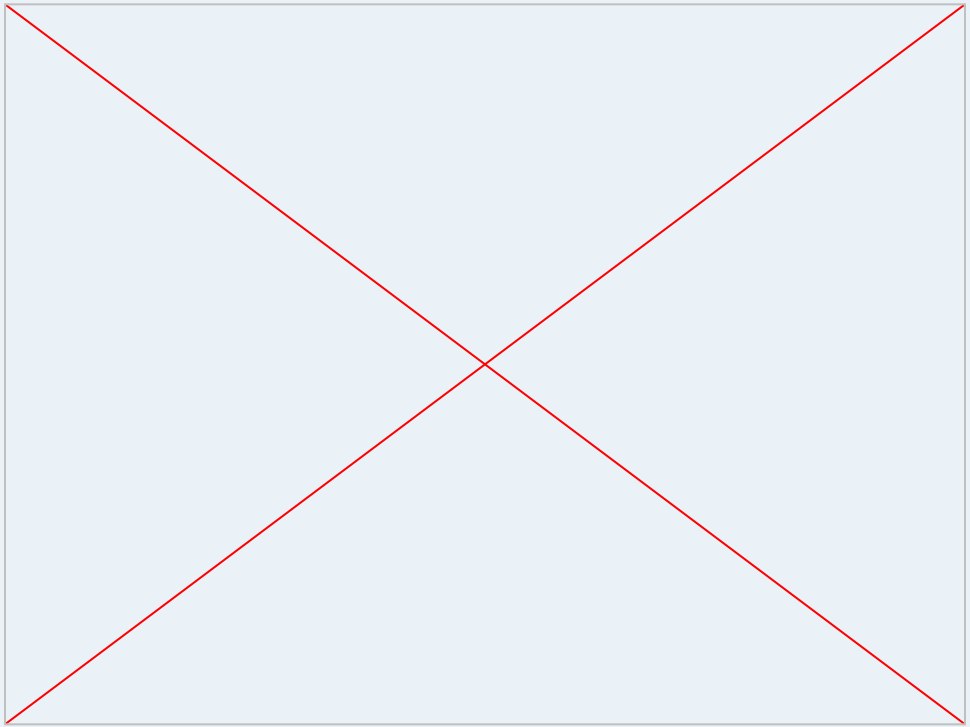
- The pipeline and supporting files are found in **/src**
- All five steps are automated inside of **pipeline.py**
- **download.py** - Step 1 (**Download**)
- **imagery.py** and **api.py** - step 2 (**Fetch**)
- **inference.py** - Steps 3 and 4 (**Infer** and **Match**)
- Step 5 (**Export**) is completed at the end of **pipeline.py**
- **/src/utils** - helper files/functions (geometry, data, geospatial, etc)

Example Usage:

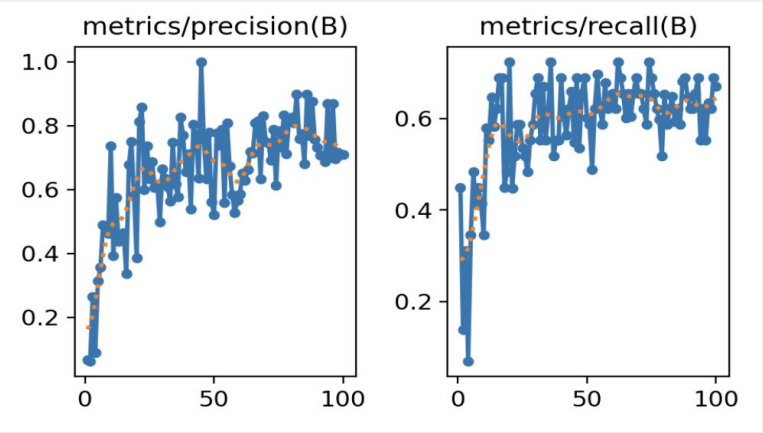
```
PYTHONUNBUFFERED=1 PYTHONPATH=. python3 -m src.pipeline \
--input_point="47.610,-122.341" \
--search_radius=100 \
--place_radius=100 \
--max_images=50 \
--prefer_360 \
--model="yolo_weights_750_image_set.pt" \
--device="cpu" \
--conf=0.60 \
--iou=0.50 \
--save="outputs"
```

Flexible input arguments - can be used for any specified input point and search radius

Demo



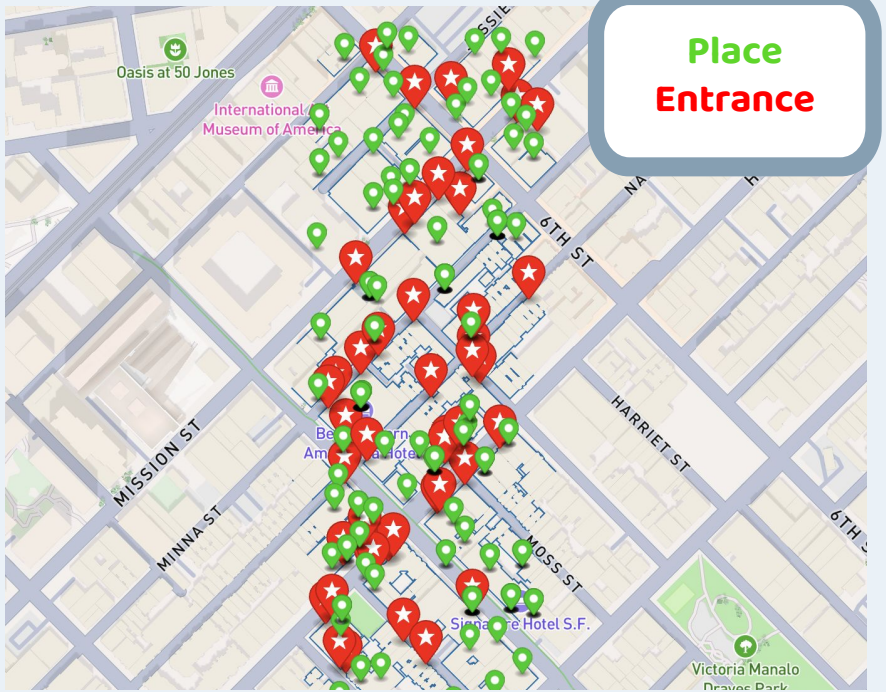
Results



- Accumulated a dataset of **750** images for model fine-tuning which resulted in **83.3%** precision and **62% recall**
- Pipeline correctly uses our model to find entrance points and match them to the correct building
- Verified detections align with real-world locations on Google Earth and geojson.io.

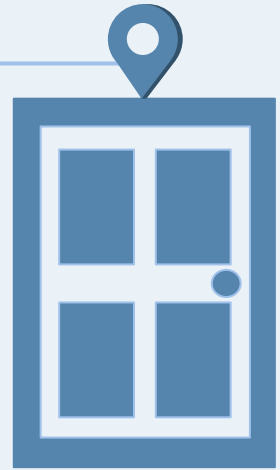
Results

- Test conducted to find entrance points for more than **10 blocks** in San Francisco (near Civic Center)
- Cross validated distances of predicted and true entrances in Mapillary imagery with Google Earth imagery
- Mean error of **4.26 m** (KR was **<5m**)
- **50** total entrance points found



Impact

- We have created an **automated, open-source pipeline** that extracts precise building entrances
- Our system works entirely with **open data** and can be **easily scaled**
- Tool to create a new Overture Maps Foundation **data layer** of entrances
- Efficient utilization of AI - no manual data collection



Takeaways

What We'd Do Differently

- Different open source models
- Attempt other filtering techniques to offload model

Open Questions / Future Work

- Explore ADA accessibility - can we detect additional information about entrance points?
- What should the scope of our model's training data be?

Team + Personal Growth

- Deepened understanding of spatial computing, computer vision integration, dataset curation
- Improved collaboration and communication skills

Thanks!

Any questions?

erantala@ucsc.edu

jphoward@ucsc.edu

Link to Git Repository:

https://github.com/project-terraforma/mapillary-entrances


slidesgo