Sprint 134 Recap

INTRODUCTION

The start of this sprint coincided with the start of the low-resolution satellite image change detection model training task. I'm calling the project "Low-Res Construction Prediction Model" for now. As a quick recap, whenever the SOC Team asks for high-resolution satellite imagery, the providers give them a large area of coverage. This is expensive and wasteful as construction doesn't occur throughout the whole extent. Therefore, there is a need to have a filter applied to the area needed, before being sent to the satellite imagery providers, so that the team only receives areas that are likely to experience construction. In other words, it would be great to train a model that could predict the likelihood of construction occurring at each point, and provide sub-areas that the team should bring to the providers.

SPRINT RECAP

There is a distinct order of tasks that need to be completed in order to train a model with the specifications of this project. Note that the details and steps may change as the project develops.

I. Choose testing locations with active construction
II. Collect imagery and other data for the two locations
    A. Key technology: Google Earth Engine
III. Build training datacube
    A. Key libraries: rasterio, xarray
    B. The dataset will include the following as features: time, true color, NDVI, elevation, slope, aspect, proximity to roads, other data sources
IV. Select pre-trained model and further train it on each location
    A. Key libraries: PyTorch
    B. HuggingFace[1] should have models that are geared towards our general use case
V. Test model on each location
    A. Location A will be used to train the model, and Location B will be used to test the model
    B. Then the reverse: Location B will be used to train the model and Location A will be used to test the model

March 31 - April 4

This week involved experimenting with and testing the Google Earth Engine API in Python to download Sentinel-2 imagery. I chose the McKinney, TX area, north of Dallas as an initial test area. This was due to the fact that my previous project involving optimizing the blob matching process used blobs that were located in this region. Google Earth Engine makes it straightforward to download imagery based on a bounding box, time window, and type of data requested. True-color, near infrared, and elevation datasets are available at low resolutions for free.

I think that NDVI will be helpful to identify areas that are not as thick in vegetation, which could indicate places that are closer to city centers and thus likelier to undergo construction. A thick forest (high NDVI, away from roads), or a remote prairie (low NDVI, but also away from roads), may not be good candidates for building new homes en masse.

Elevation (and derived data like slope or aspect) can indicate whether a place is viable for construction: a steep mountainside will be challenging to build on.

The road network is more difficult to register at each time slice as it is difficult to get a historical record of the road network developing over time. Putting the time question aside, I thought of using a kernel density estimation to create a layer derived from the roads that acts as a measure of the density and proximity of the roads at each point. If we think of the density of the roads as a hypothetical surface, the higher the density, the higher the scalar value at that point. We could apply this density value to each grid cell and feed that into the model. See above regarding the NDVI discussion - higher density of roads indicates more economic activity which could indicate a higher chance of construction happening.

Other data sources, like proximity to schools, or soil type, could be included to enrich the model, though more time would be needed to actually incorporate them into the datacube.

April 7 - April 11

The second week of the sprint involved me refining the Google Earth Engine code so that it could receive a specific geohash and output all imagery of that location. I chose two geohashes to focus my initial efforts on: `9vgm6` and `9vgmd`.

I also created a map which can be seen in **Figure 1**[2]. This showcases the types of imagery that I have collected thus far.

I created a draft datacube that incorporates time, location, RGB (true color), NDVI, elevation, slope, and aspect.

LOOKING AHEAD TO SPRINT 135

With the training datacube in place, the next sprint should involve training and testing an initial model. I'm hoping to select a pre-trained model from HuggingFace that matches the project's goals. I will train it further on our specific data and test it. It would be great to incorporate other data sources.

APPENDIX

[1] https://huggingface.co/

[2] **Figure 1**