2020-12-07

DeepMapper : Automatic Updating Crowdsourced Maps

Lasith Niroshan Hewa Manage
Technological University Dublin, d19126805@mytudublin.ie

James Carswell
Technological University Dublin, james.carswell@tudublin.ie

Funder: Technological University Dublin College of Arts and Tourism

DeepMapper : Automatic Updating Crowdsourced Maps

Lasith Niroshan, James D. Carswell

Technological University Dublin
{lasithniroshan.kottawahewamanage@mydit.ie , james.carswell@tudublin.ie}

Introduction

To get accurate information returned from location-based services (e.g., LBS info on nearby restaurants, retail outlets, points-of-interest, etc.), the underlying map (spatial data) must be up-to-date. However, the built environment (e.g., roads, buildings, bike paths, etc.) can change quickly over time, either through planned developments or as the result of natural/manmade disasters. The problem is that keeping online crowdsourced maps like Open Street Map (OSM) updated is still very much a manual process. As such, it can take considerable time to sync the online maps used by LBS with up-to-date spatial data in "real-time".

Figure 1. The process of bringing OSM vector geometries up-to-date can be automated with the help of satellite imagery, deep learning, and people.

Our case study considers the Grangegorman area in Dublin city. It is a green/brownfield site that has seen much infrastructure change in the past decade due to its renewal as the new home for Technological University Dublin (TU Dublin). New buildings have been built, car parks have changed size/location, and new roads have been constructed due to the recent expansion. Figure 2 and Figure 3 illustrate the difference/mismatch between the current OSM crowdsourced (vector) map and Google's satellite (raster) view of the Grangegorman area.

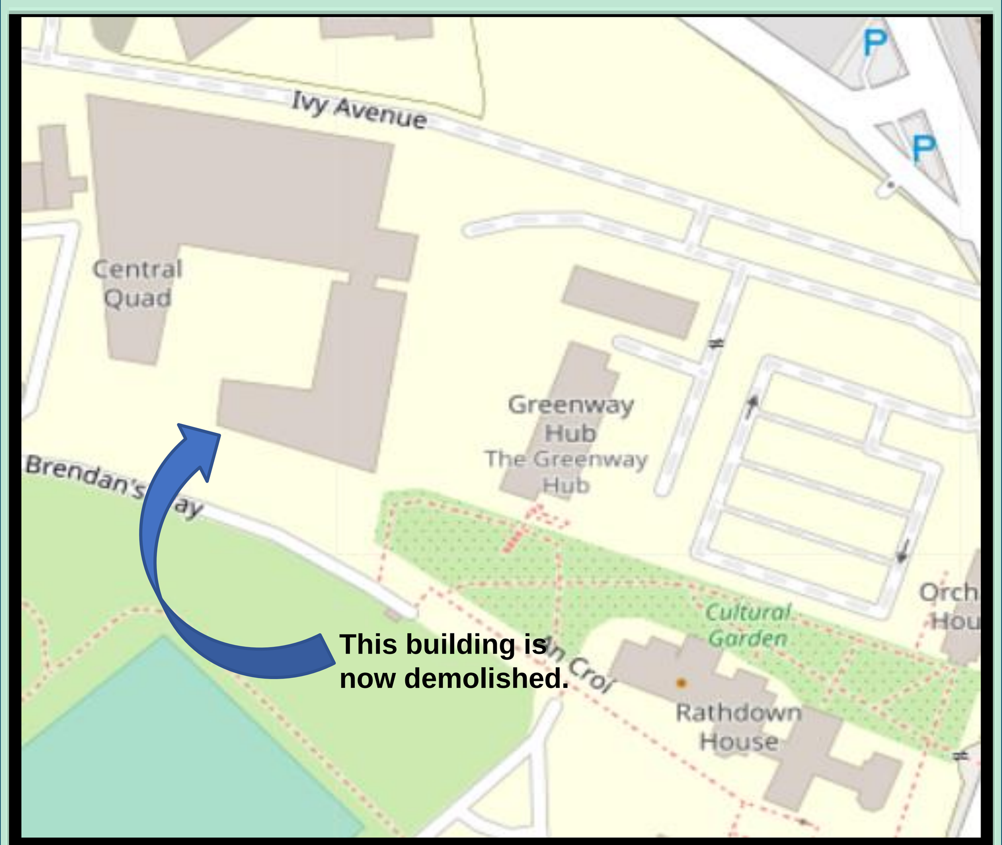

Figure 2. Current OSM vector map of TU Dublin in Grangegorman (Sept. 2020)

Figure 3. Current Google Earth raster image of TU Dublin in Grangegorman (Sept. 2020)

Aims & Objectives

This project aims to develop an end-to-end map processing pipeline to update crowdsourced maps with the help of satellite imagery, deep learning, and people (Figure 1). The significant objectives to complete the pipeline are:

- Raster and vector data crawling
- Extracting relevant information from satellite images
- Raster and vector data comparison
- Rebuild OSM standard spatial data *changeset* file
- Verify changeset with area-specific knowledge
- Upload changeset to OSM

Methodology

Build the Deep Learning model
Our research focuses on developing reconfigurable deep learning models to detect built environment changes in satellite imagery within a particular region (minimum bounding rectangle). These deep learning models need to be sensitive to even relatively small geo-spatial data changes, so several aspects need to be considered.

Freely available, high-resolution (+/-5m pixels) raster satellite image data is used as input for the training process. The Mask-RCNN algorithm [1] is used to build the deep neural network model.

We also benefit from using freely available OSM vector data to populate our training dataset (generate images and masks). Figure 4 shows an instance of this dataset. While the left-side image (A) is the training image, the right-side image (B) indicates the relevant image mask.

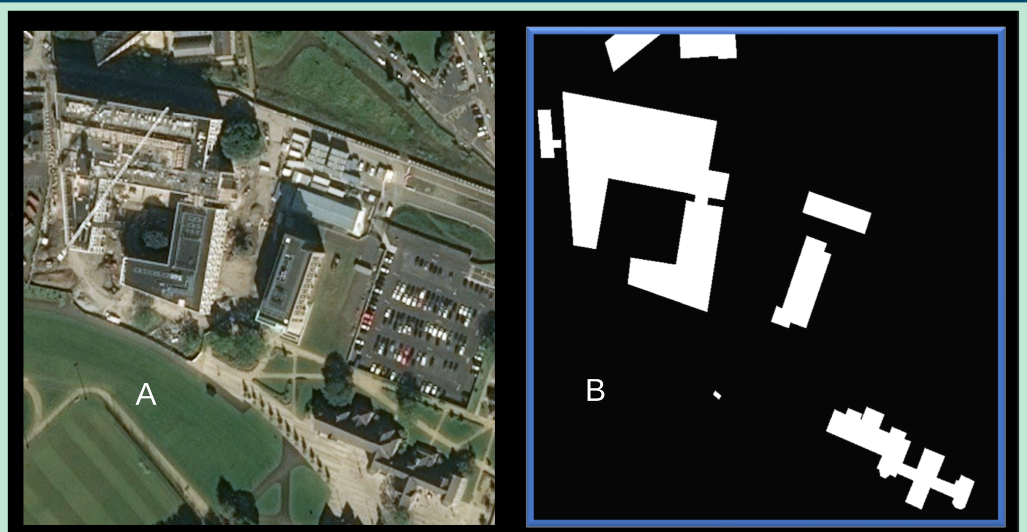

Figure 4. An example of a training image (left) and its mask (right) generated from OpenStreetMap data

OpenStreetMap updating pipeline
We propose a phase-based solution to the problem over the *DeepMapper* pipeline. Figure 5 illustrates our research methodology, including major modules and sub-modules.

- The ***vector crawling module*** downloads the OSM vector data [2] from the OSM server.
- Before initializing the ***raster-vector comparison module***, the deep neural model will inference any pre-defined objects (e.g., buildings, roads) in the satellite images within the specified minimum bounding rectangle.

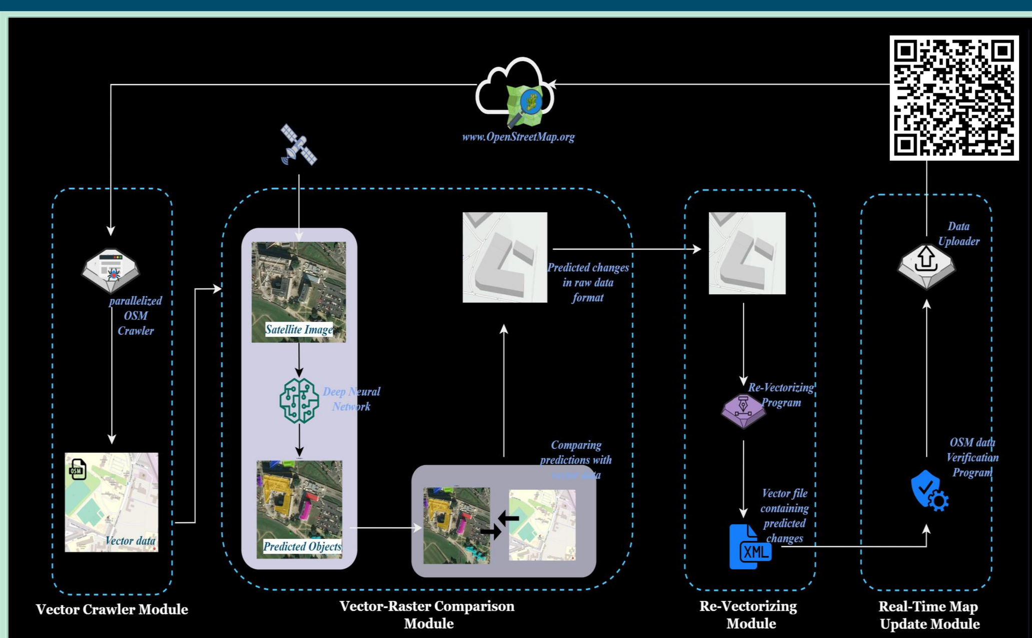

Figure 5. System Architecture Diagram of *DeepMapper*

- Once the inference is completed, the output (raw data) is compared to the previously downloaded OSM data, and possible changes are logged into the next phase.
- The ***re-vectorizing module*** is responsible for converting any detected changes into the vector format adhering to OSM rules.
- The verification process [3] verifies the prepared OSM file. Finally, the XML formatted changes are uploaded to the OSM server via the ***map update module.***

Results

Experimental evaluation of proposed technique
First, a web portal provides a labeled map where users can digitize their area of interest (AoI) as a multisided polygon. Then the application calculates the minimum bounding rectangle (MBR) for the given AoI (Figure 6).

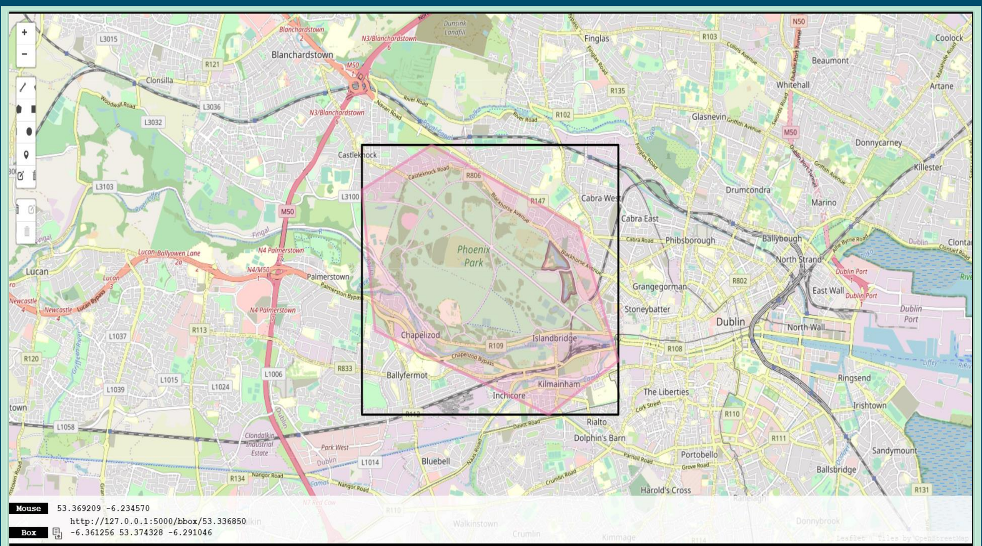

Figure 6. Minimum bounding rectangle finder (web portal)

Once the app obtains MBR coordinates from the web portal, the vector crawler starts to download any contained OSM spatial data in *.osm* format. During the vector crawling process, the Deep Neural Model launches to find possible objects (we consider only buildings in the first stage) in the satellite imagery within the MBR coordinates. Figure 7 shows the preliminary results of building detection using our Mask-RCNN model.

Figure 7. Building detection (instance segmentation) results using Mask-RCNN

Conclusions

Due to the time-consuming manual process and other errors that can arise with contemporary online map production, developing an automated map updating process to maintain up-to-date maps is beneficial to modern society in many ways.

Several related works were studied regarding collecting map data, comparing existing raster and vector data, as well as contemporary approaches for automatic updating of crowdsourced maps (specifically OSM). It was found that there are still gaps in the current state-of-the-art for this technology when applied to performing the entire OSM map update process automatically.

Future research directions will investigate improvements to object detection accuracy through sensing dark and shady edge detection. We will also develop an appropriate polygonization algorithm to construct precise building shapes from the predictions. Finally, the *DeepMapper* pipeline will be fully implemented and tested, with experimental results submitted for publication in leading GIS journals.

References:
1. He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).
2. Mooney, P. and Minghini, M., 2017. A review of OpenStreetMap data.
3. Fan, H., Zipf, A., Fu, Q. and Neis, P., 2014. Quality assessment for building footprints data on OpenStreetMap. International Journal of Geographical Information Science, 28(4), pp.700-719.