Extraction of Building Footprints from Satellite Imagery

Elliott Chartock, Whitney LaRow, Vijay Singh

Motivation

Geospatial mapping is a hundred billion dollar industry that, to date, relies heavily on manual techniques. The application of Computer Vision (CV) and deep learning techniques to automate mapping will lead to higher quality and more resilient mapping, contribute to CV feature extraction algorithms, and has proven benefits in humanitarian work and disaster response efforts. Recent attempts to extract building footprints with Convolutional Neural Networks have yielded promising results, and we hope to improve on previous approaches.

Problem Statement

- Our goal is to accurately extract building footprint polygons from high-resolution satellite imagery
- We will pass image pixels through a Fully Convolutional Neural Network (FCNN) to predict building footprint boundaries
- We evaluate our predictions with the F1 score of proposed polygons, where true positives are predictions which have greater than 0.5 Intersection over Union (IoU) with labelled polygons

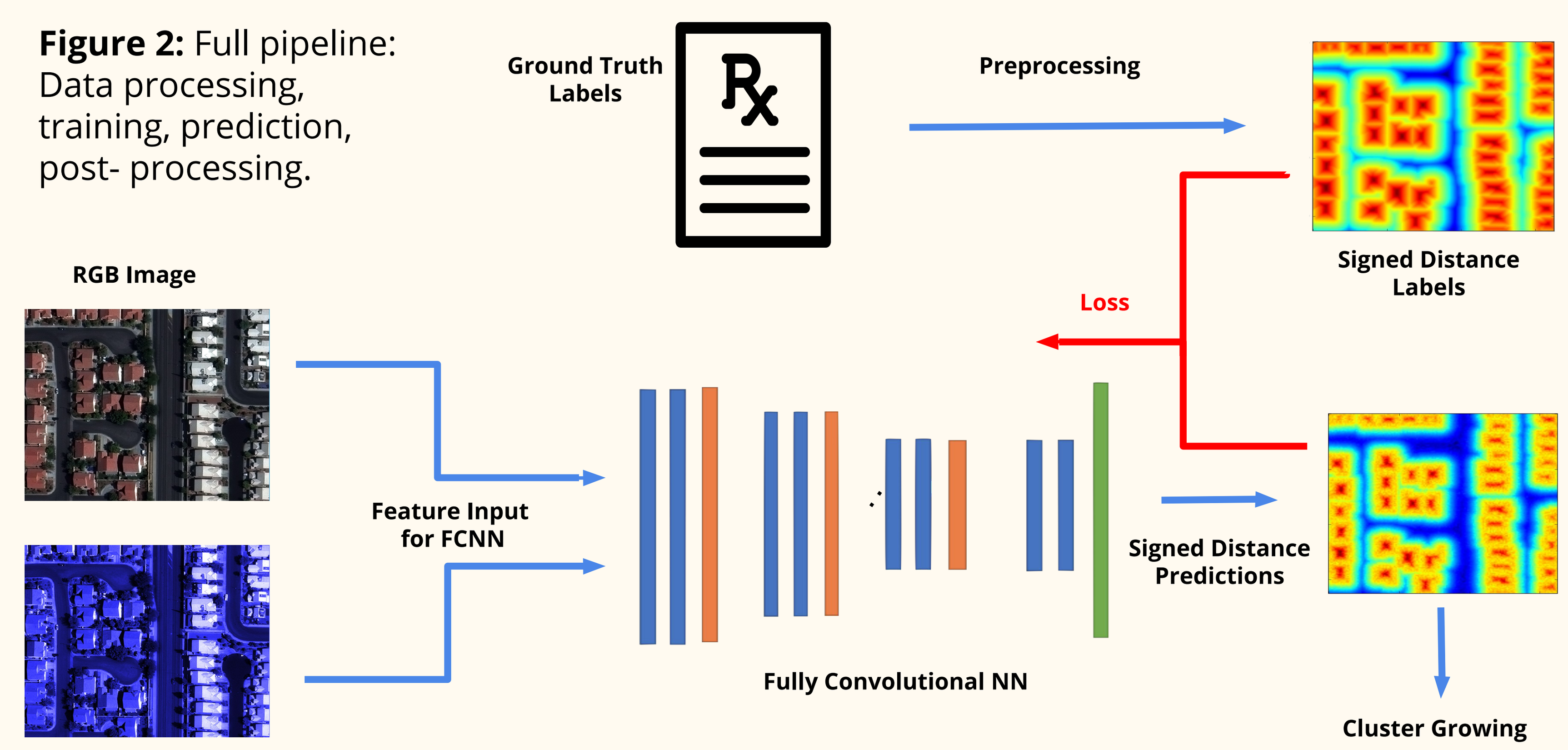
Figure 1: Concatenate RGB and 8-band multi-channel images to produce model input features with dimension (650, 650, 11)

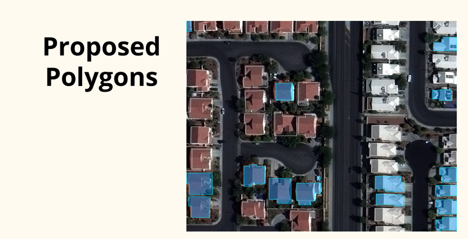
Data

- 10,000 16-bit GeoTiff images collected by the DigitalGlobe Worldview-3 satellite. 70% Train, 15% Validation, 15% Test
  - **Locations**: Las Vegas, Paris, Shanghai and Khartoum
  - **Formats**: grayscale, RGB, 8-band multi-channel, and higher-resolution 8-band multi-channel
  - **Scale**: 200 meter x 200 meter ground area
- Ground truth labels in geoJSON format; requires preprocessing
- Poster results are on a smaller subset of the data; full test set will not be tested until final FCNN model is obtained

Methodology

Figure 2: Full pipeline: Data processing, training, prediction, post- processing.





Preprocessing

Signed Distance Transform

- Rasterize labelled footprints so that each pixel has the value of the distance to the nearest boundary of a building
- Exterior pixels have negative distance, interior pixels have positive distance, footprint border pixels have distance 0
- Scale distance values to be between -1 and 1

Model

Fully Convolutional Neural Network

- **Loss**: MSE **Optimizer**: Adam

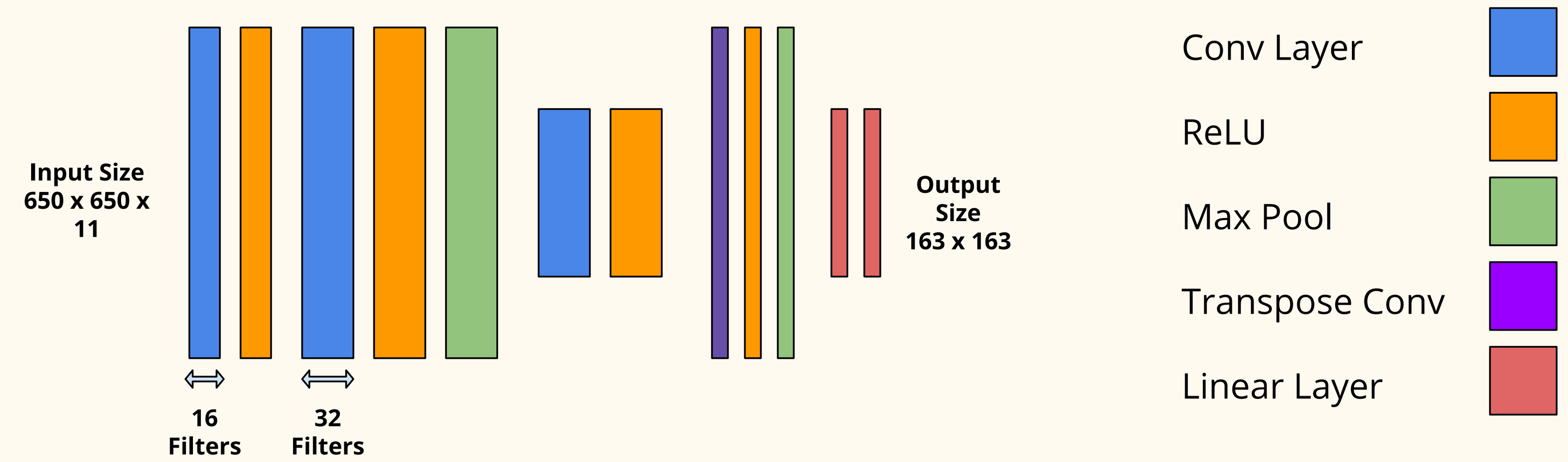


Figure 3: Best FCNN model. Color index:

Postprocessing

Cluster Growing Greedy Algorithm

- Consider only non-negative pixels in the signed distance prediction output from the FCNN
- Greedily create clusters of contiguous regions of the image with strictly decreasing values (to distinguish buildings)
- Convert the rasterized cluster data into vectorized GeoJSON polygons to be fed into our F1 score calculator

Results

Model	Baseline (4-Layer CNN)	FCNN (Train)	FCNN (Test)
Precision	0.004	0.252	0.182
Recall	0.042	0.689	0.415
F1	0.007	0.370	0.253

Figure 4: Results table of initial testing on small sample dataset. Preliminary results show that best FCNN model significantly outperforms baseline CNN.

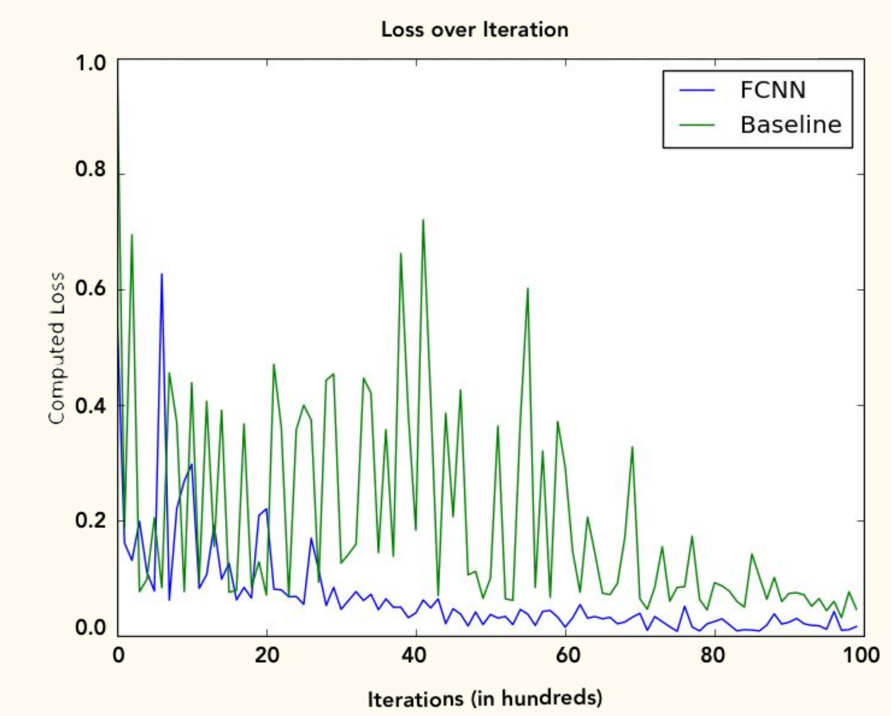


Figure 5: Loss curve shows FCNN model has lower loss than Baseline 4-Layer CNN model. FCNN model has general trend of decreasing loss with more iterations.

Analysis

- After experimenting with multiple networks, we found that using upsampling greatly increased our performance
- Our method of using signed distances as labels has promising performance results and presents a new approach to the problem we are trying to solve
- Our model had low precision because the post-processing performs suboptimally in certain cases (e.g. when there are long, skinny buildings, concave buildings, or no buildings)

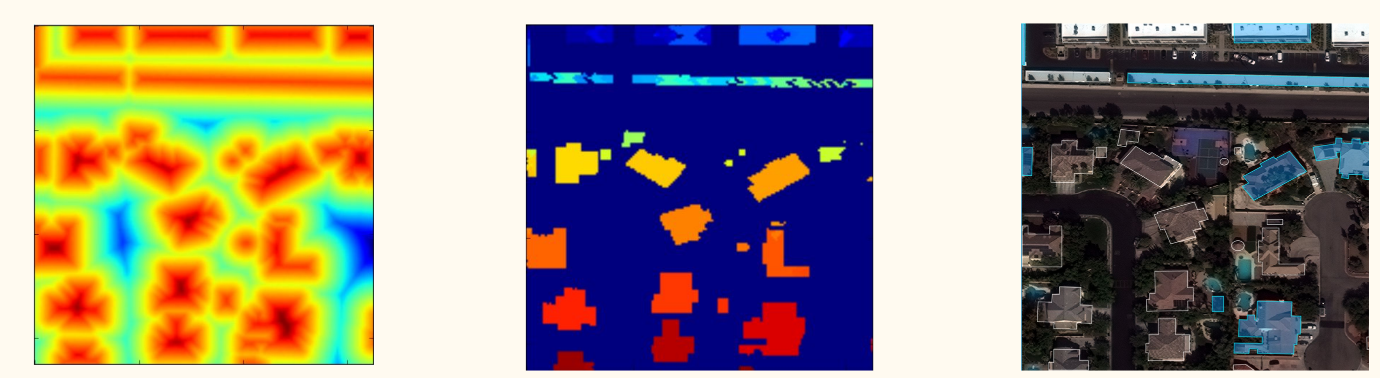


Figure 6: Post-processing errors: for a good heatmap, the clusters for long, skinny buildings can be broken up erroneously, causing us to miss those buildings.

Next Steps

- Try adding dilated convolutions (to better capture spatial information) and batch normalization in future models to see how these layers affect our performance
- Improve post-processing to better handle images without buildings present, potentially using a Fully-Connected Network
- Experiment with Mask R-CNNs