Department of Computer Science and Engineering, The Ohio State University

# P29 ROAD NETWORK RECONSTRUCTION FROM SATELLITE IMAGES WITH MACHINE LEARNING SUPPORTED BY TOPOLOGICAL METHODS

Present by Jiayuan Wang (wang.6195@osu.edu, http://web.cse.ohio-state.edu/~wang.6195/)
Joint work with Tamal K. Dey and Yusu Wang

## INTRODUCTION

- Goal: Satellite images -> Road networks represented by graphs.
- Method: Machine learning and Topological Data Analysis.
- Semi-automatic: Require the user to provide training samples.

**Our contributions:**

**1. Semi-automatic framework:**

- Combine a discrete-Morse based graph reconstruction algorithm with an existing CNN framework to segment input satellite images.
- Better connectivity and less noise.

**2. Fully automatic framework:**

- Leverage the power of the discrete-Morse based graph reconstruction algorithm to train a CNN from a collection of images **without labelled data.**
- Apply the discrete-Morse based graph reconstruction algorithm iteratively to improve the accuracy of the CNN.

Dataset: SpaceNet Challenge Round 3

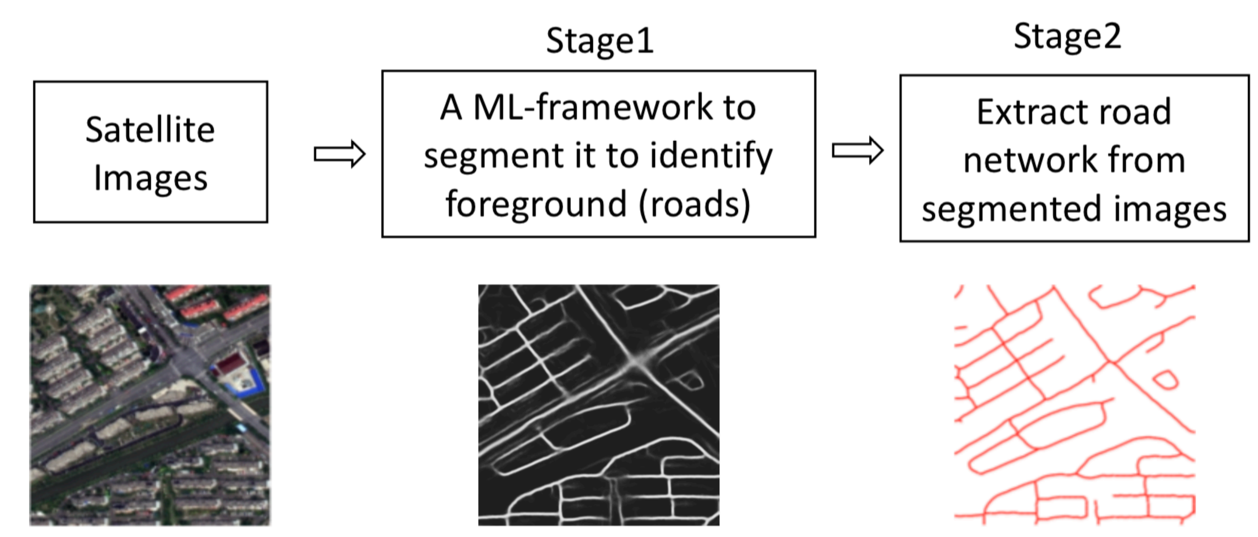



Figure 1. High level road network reconstruction from satellite images framework pipeline.

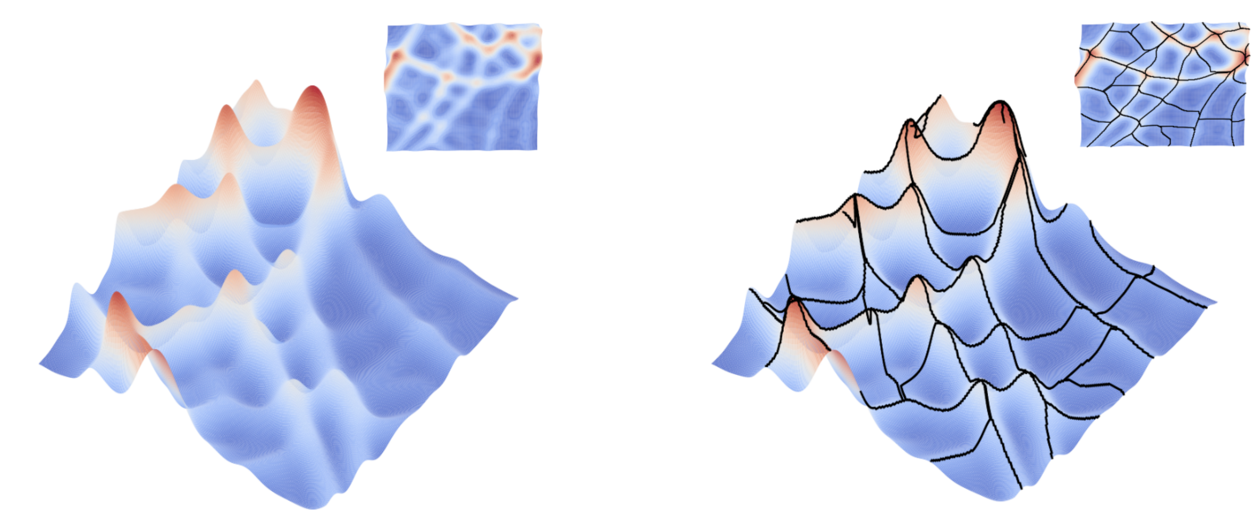


Figure 2. (a) An input density field on the plane (top corner) and its terrain view. (b) Mountain ridges of the terrain (black lines) capture the road network.

## SEMI-AUTOMATIC FRAMEWORK

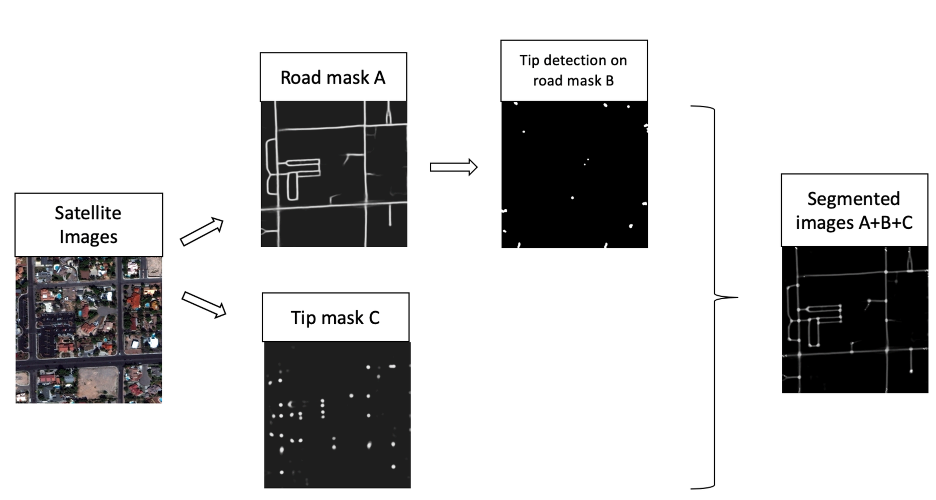



Figure 3. The pipeline for Stage 1 (CNN training) for our semi-automatic framework.

| | APLS | | $S_H$ | |
|---|---|---|---|---|
| | Buslaev[1] | ours | Buslaev[1] | ours |
| AOI_2 | 0.8211 | **0.8278** | 18.3539 | **17.7841** |
| AOI_3 | 0.5848 | **0.6324** | 291.0188 | **289.9532** |
| AOI_4 | 0.6630 | **0.6632** | 69.5775 | **68.9596** |
| AOI_5 | 0.6069 | **0.6477** | 44.4201 | **41.6037** |

Table 1. APLS score and $S_H$ score for the test set.

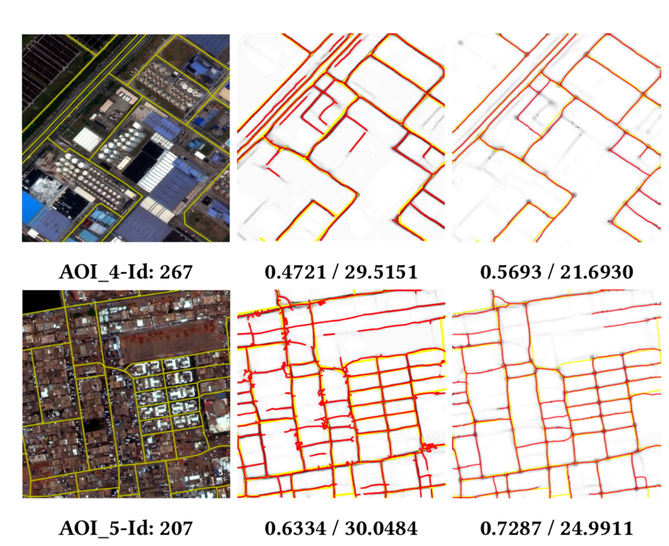



Figure 4. Left: original satellite image overlapped with ground truth (Yellow). Middle: The winner's results (Red). Right: Ours (Red).

| | Train | Test | Road extraction |
|---|---|---|---|
| AOI_2 | $2 \times 242m$ | $8m \times 2$ | $22m \times 9$ |
| AOI_3 | $77m \times 2$ | $2m \times 2$ | $6m \times 9$ |
| AOI_4 | $293m \times 2$ | $10m \times 2$ | $27m \times 9$ |
| AOI_5 | $71m \times 2$ | $2m \times 2$ | $6m \times 9$ |

Table 2: The running time: m stands for minutes; 2× means it will be run twice (for both road and tip detections). 9× comes from the tuning of the parameters.

Definition 4.1. *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two input graphs. For $a, b \in V_1$ where $path(a, b)$ exists in $G_1$, let $a'$ (resp. $b'$) denote the closet node to $a$ (resp. to $b$) in $G_2$. $L(\cdot,\cdot)$ denote the length of the shortest path. First we define the cost of $path(a, b)$:*

$$c(a,b) = \begin{cases} min\left\{1, \frac{|L(a,b)-L(a',b')|)}{L(a,b)}\right\}, & if\ path(a',b')\ exists \\ 1, & otherwise \end{cases}$$

*We next define*

$$C(G_1, G_2) = 1 - \frac{1}{N}\sum c(a,b)$$

*Where N = # unique paths in $G_1$, and we take the sum over all unique paths. Finally, the* APLS *score of $G_1$* and *$G_2$ is defined to be the harmonic mean of $C(G_1, G_2)$ and $C(G_2, G_1)$:*

$$APLS(G_1, G_2) = \frac{2}{\frac{1}{C(G_1,G_2)} + \frac{1}{C(G_2,G_1)}}$$

Definition 4.2. *Suppose $G_1$ and $G_2$ are two graphs; $P_1$ is the point set sampled from $G_1$; $P_2$ is the point set sampled from $G_2$, and $d$ denotes the Euclidean distance. Then, the one-directional Hausdorff distance is:*

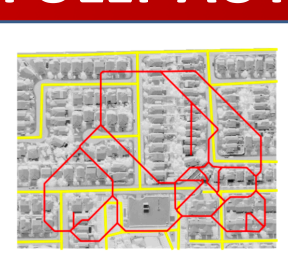
$$S_{HA}(G_1, G_2) := \begin{cases} \frac{1}{|P_1|}\sum_{p \in P_1} d(p, P_2), & G_1 \neq \emptyset\ and\ G_2 \neq \emptyset \\ MAX, & Only\ one\ graph\ is\ empty \\ 0, & G_1 = G_2 = \emptyset \end{cases}$$

*Here, MAX is a specific maximum value. We set $S_H(G_1, G_2) = S_{HA}(G_1, G_2) + S_{HA}(G_2, G_1)$ as final* average Hausdorff distance *between $G_1$ and $G_2$.*

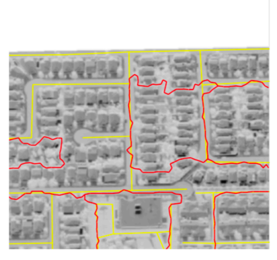
**Buslaev's[1] method** The same CNN architecture. In Stage 2, Buslaev's method first extracts the skeleton from the thresholded segmented images. Then, it transforms the skeleton to a multi-graph using library "sknw". Finally, it translates the multi-graph to a graph with straight edges.

[1] A. Buslaev. 2018. Spacenet round 3 winner. https://github.com/SpaceNetChallenge/RoadDetector/tree/master/albu-solution. (2018).

## FULLY AUTOMATIC FRAMEWORK



(a) Skeleton by (albu, 2018)



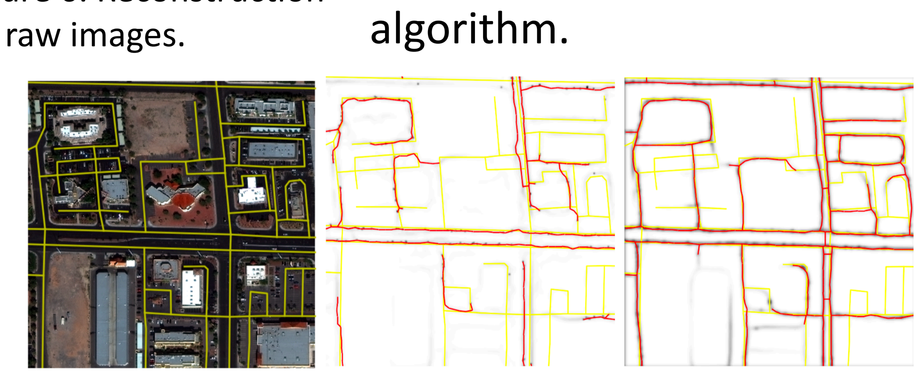
(b) Skeleton by discrete-Morse alg.

Figure 6. Reconstruction on raw images.

**Algorithm 1** MorseLabelTrain($I$)

**Data:** Images $I$
**Result:** Classifier $C$
**main**
  Compute the triangulation $K$ of $I$ and take pixel values as density function $\rho$
  $\widehat{G}$ = MorseGraphRecon($K$,$\rho$, $\delta$)
  Label pixels on $\widehat{G}$ as positive, pixels on the complement of $\widehat{G}$ as negative
  Train a CNN classifier $C$ by above features
  **return** $C$
**end main**

**SkeletonLabelTrain():**
The graph reconstruction algorithm is replaced with the winner's skeleton extraction algorithm.



AOI_2 - Id: 323   $I_1^{te}$: 0.0653 / 0.1213   $I_{11}^{te}$: 0.5564 / 0.1530

Figure 7. Middle / right: the reconstructed graph using CNN after the first and the 11-th iterations.

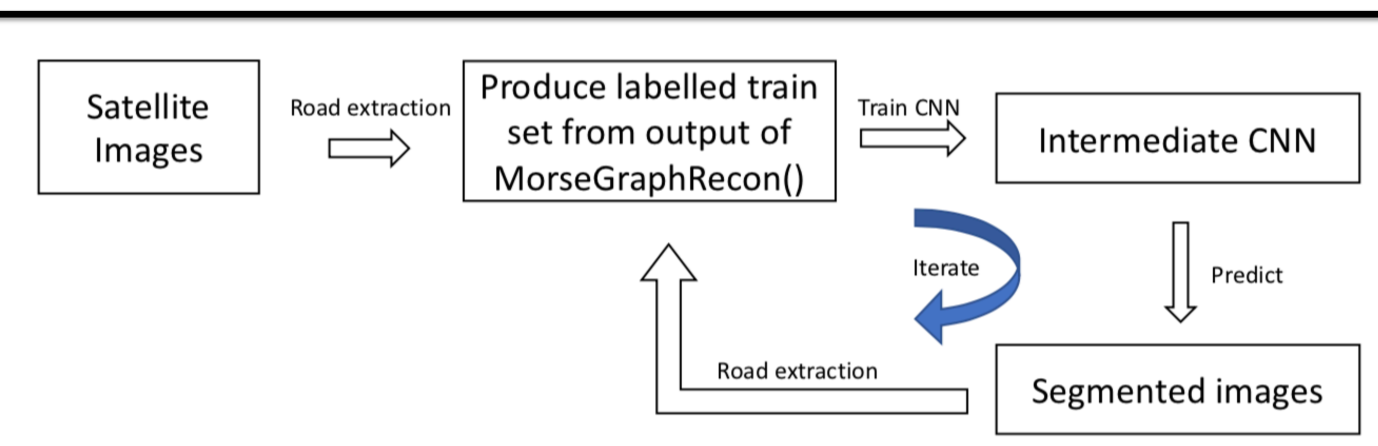



Figure 5. Pipeline for Stage 1 (CNN training) for our fully automatic framework. Note that no input satellite image has labels for roads!

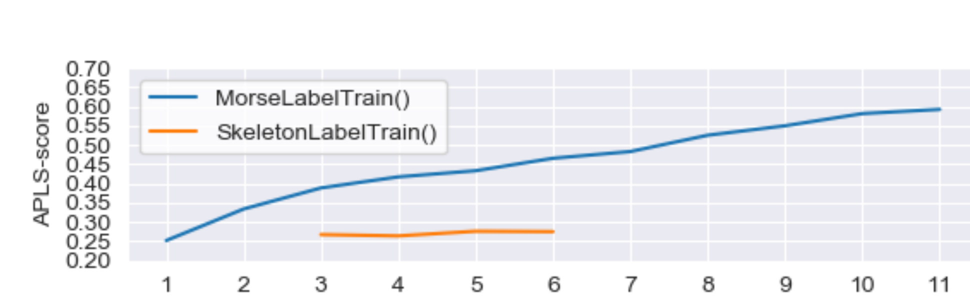
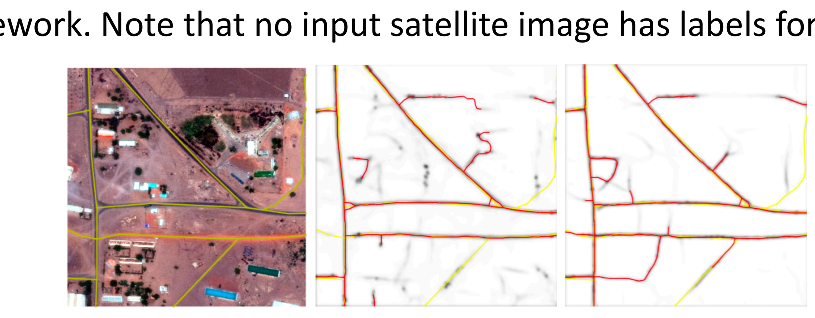



Table 2. APLS-score for the reconstructed road networks for testing images, based on our label-free framework (MorseLabelTrain()), compared to the alternative method SkeletonLabelTrain(). The first two iterations for SkeletonLabelTrain() is done by MorseLabelTrain() and thus are not shown. After 6 iterations, the score does not improve for SkeletonLabelTrain() any more.

| APLS | $I_1^{test}$ | $I_2^{test}$ | $I_3^{test}$ | $I_4^{test}$ |
|---|---|---|---|---|
| AOI_2 ours | 0.6521 | 0.6918 | 0.7305 | 0.7210 |
| AOI_2 Skel. | 0.4860 | 0.5137 | 0.5214 | 0.5252 |
| AOI_5 ours | 0.5351 | 0.5787 | 0.5893 | 0.6077 |
| AOI_5 Skel. | 0.5247 | 0.5091 | 0.4884 | 0.4543 |

Table 3. APLS score for partial-labeled case, where 10% random images have road labels.



AOI_5 - Id: 50   $I_1^{te}$: 0.6449 / 0.0581   $I_4^{te}$: 0.6648 / 0.0788

Figure 8. Using 10% labelled images. Reconstructed graphs by our MorseLabelTrain() after 1 and 4 iterations.

AOI_5 50   $I_1^{te}$: 0.4533 / 0.1365   $I_4^{te}$: 0.1940 / 0.1728

Figure 9. Using 10% labelled images. Reconstruction of the alternative SkeletonLabelTrain() after 1 and 4 iterations.