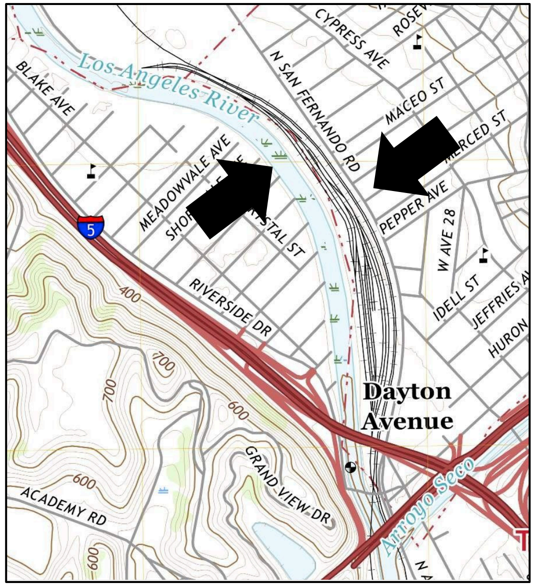
A Label Correction Algorithm Using Prior Information for Automatic and Accurate Geospatial Object Recognition

Presenter: Weiwei Duan

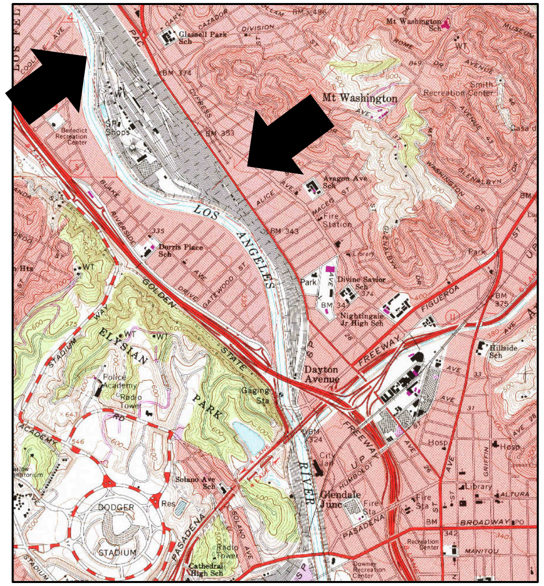
University of Southern California

Motivation

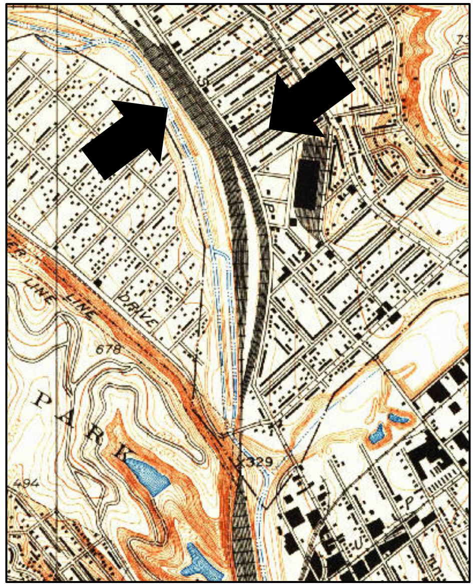
- Historical maps store valuable information
    - The railroad network evolvement in Los Angeles



2018



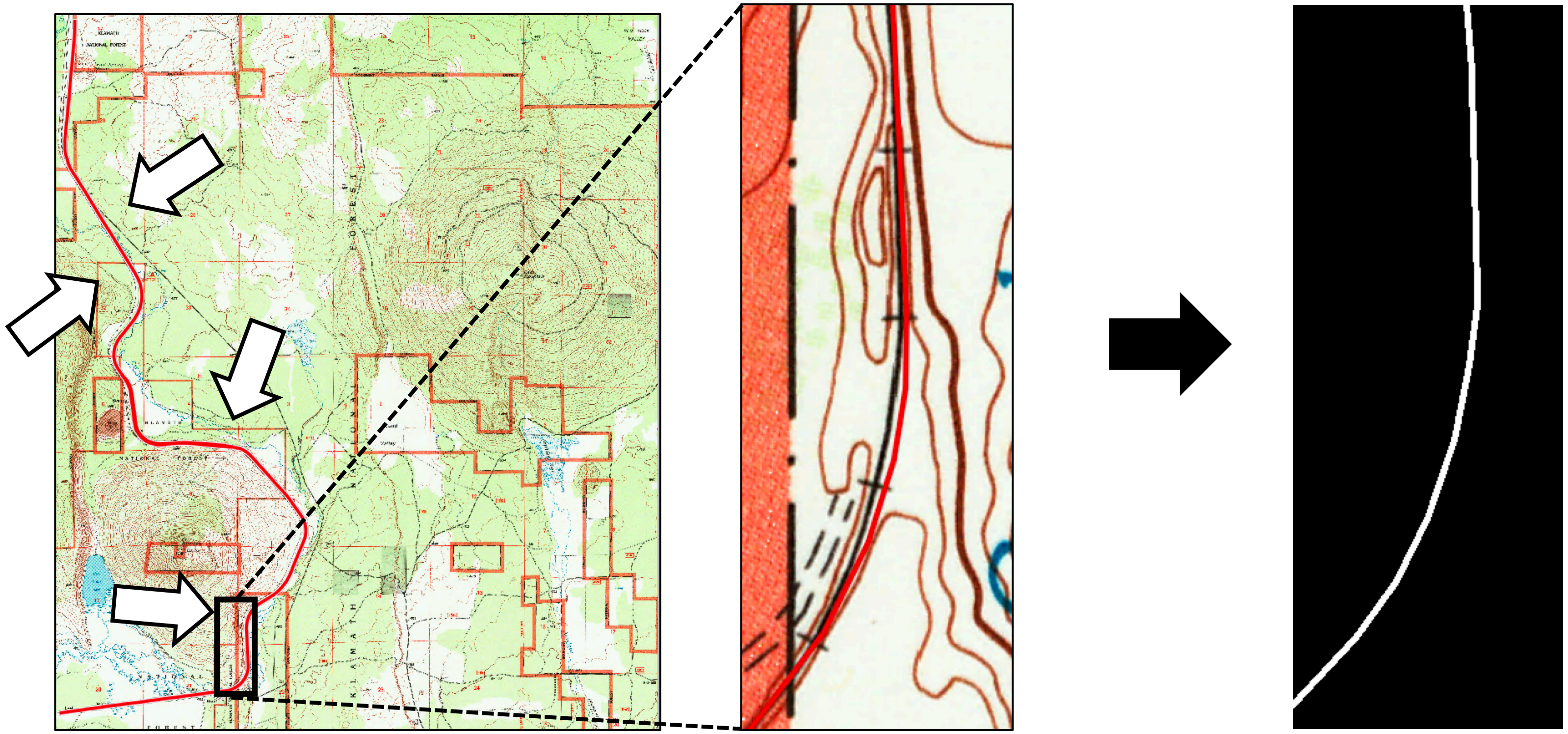
1966



1928

Problem

- Automatically detecting linear target objects’ locations in the maps
- To reduce the manual work, we use the external vector data to automatically annotate the target objects’ locations in the maps



The overlaying vector and maps

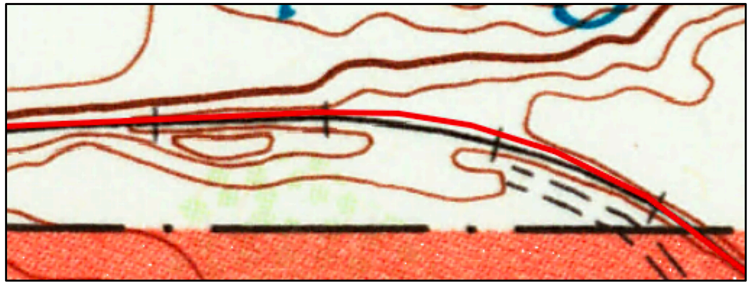
Annotation

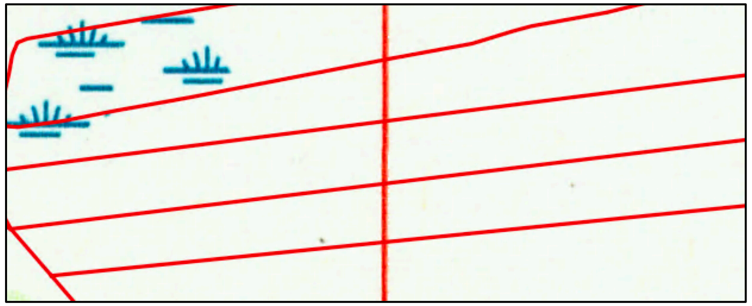
Challenges

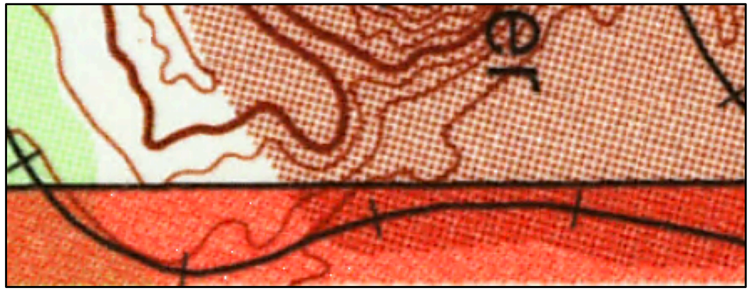
- Misaligned annotations

  Because of different scales and coordinate projection systems between the map and vector data
- False annotations
- Missing annotations

  Because of different publishing years between the map and vector data







Related Work

- Semantic segmentation models dealing with the noisy annotations
  - Before the training process
    - An alignment correction network (ACN) requires manual annotations
  - During the training process
    - Design new loss function based the similarity measurements
    - The normalized cut loss (Ncut) measures pixels' similarity in the color and spatial space
- Alignment algorithms for vector data and raster images
  - Vector-to-raster alignment algorithm

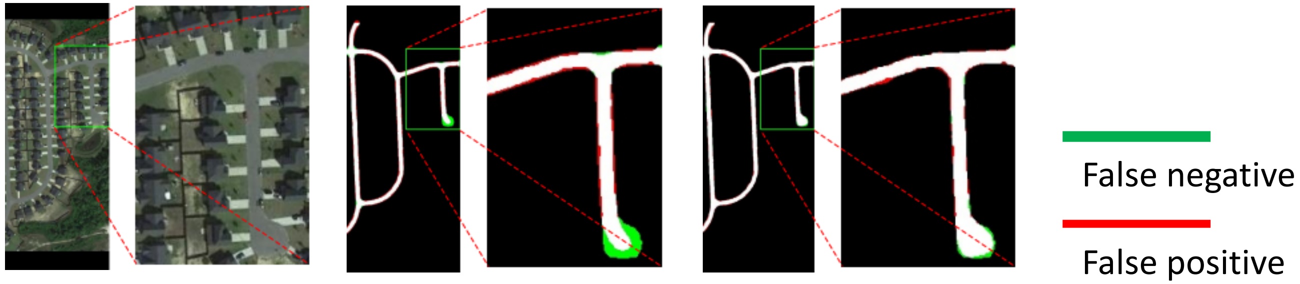
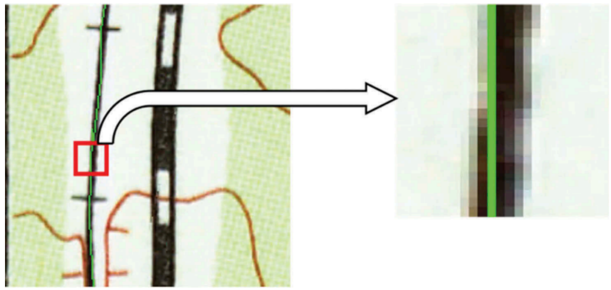
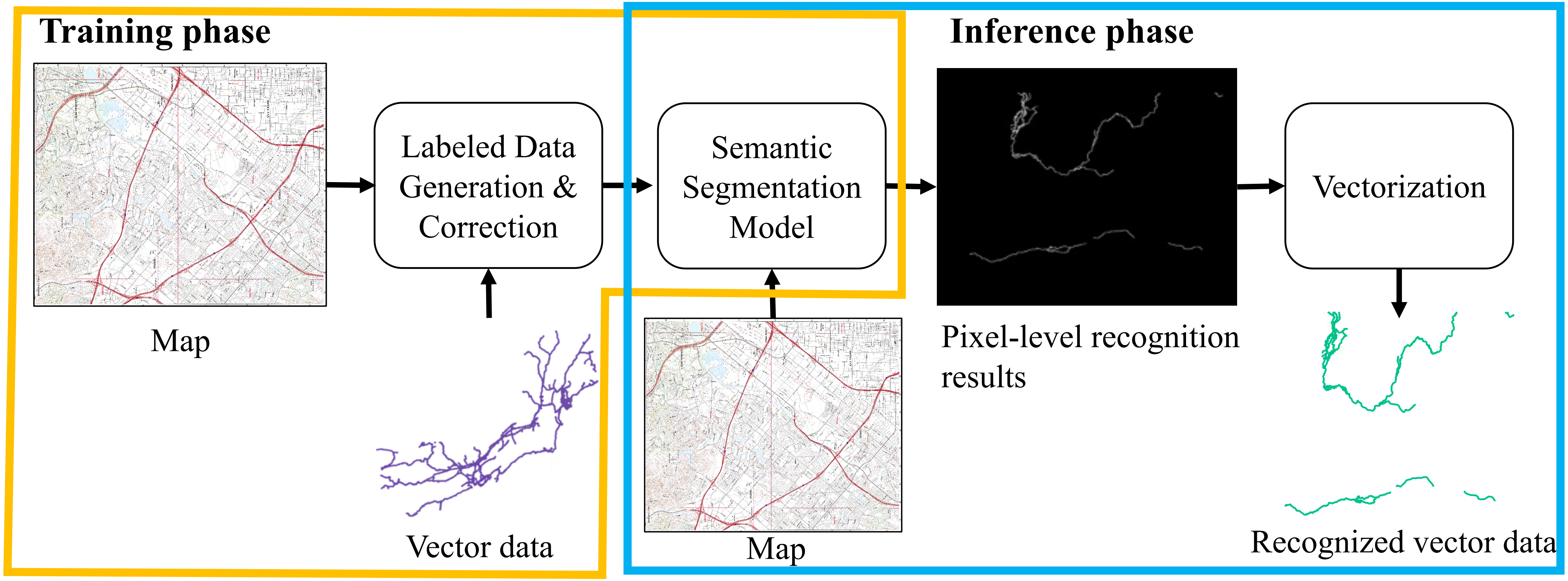


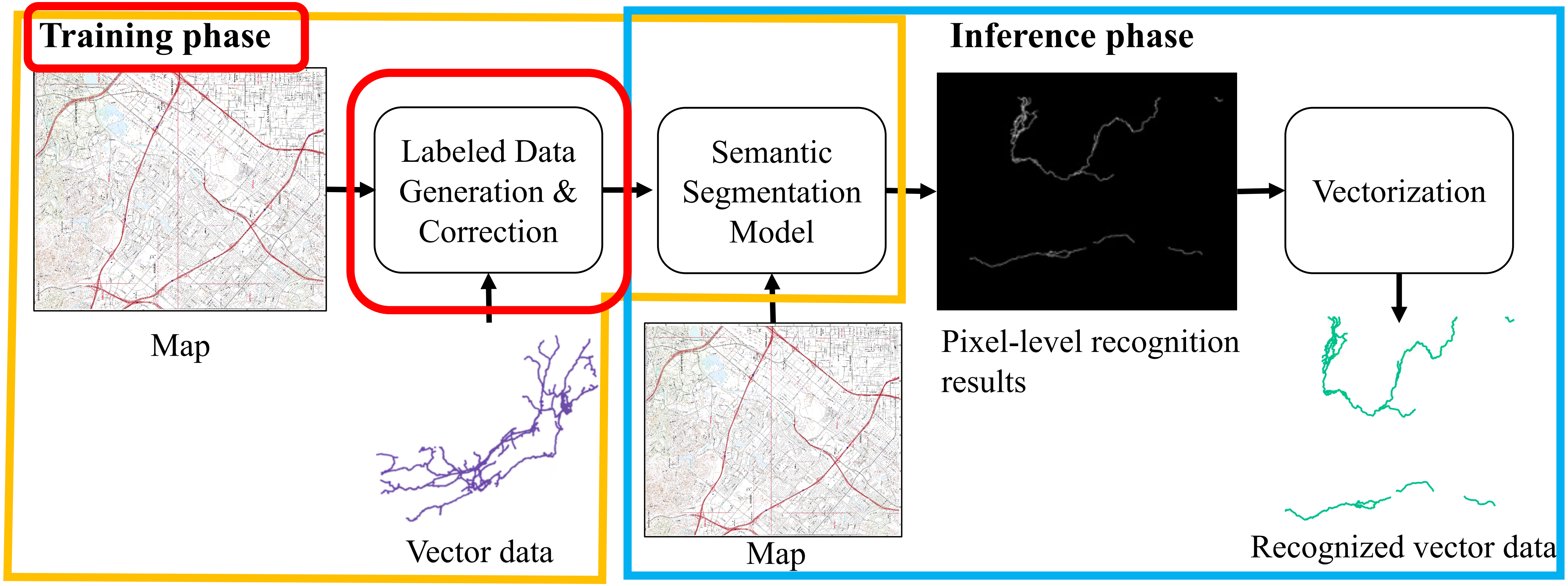
Image Annotations Inference



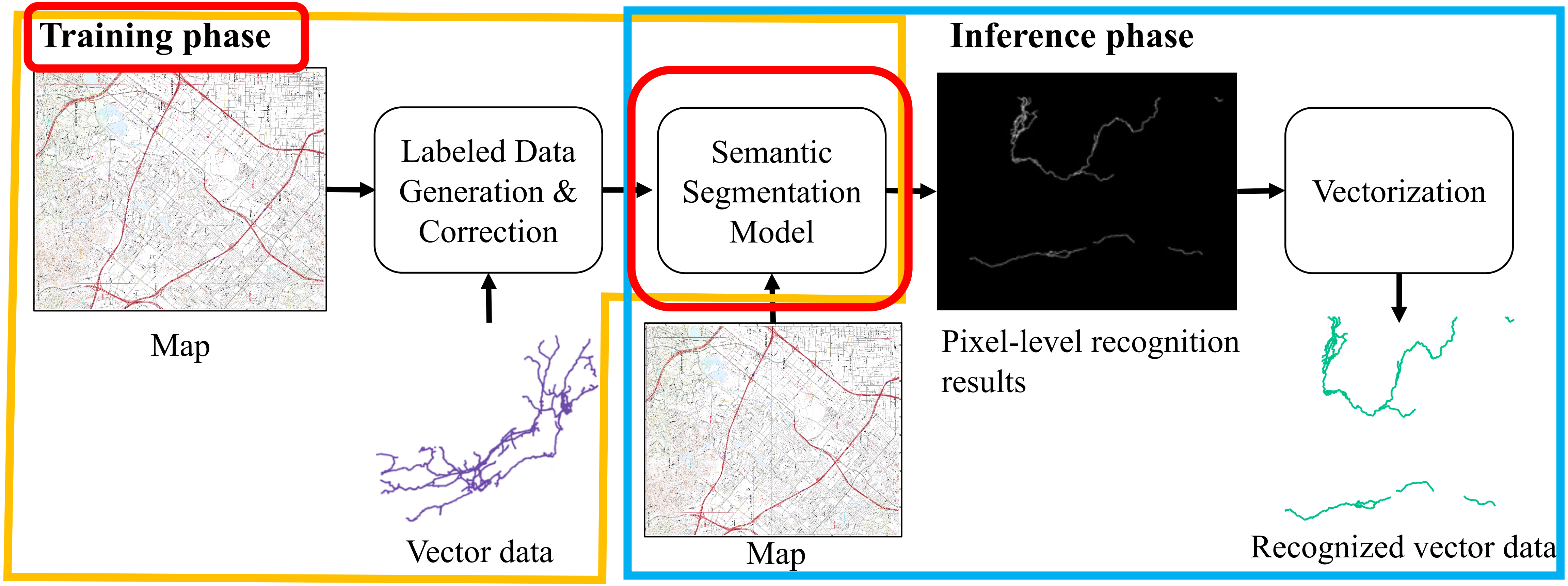
Our Geospatial Object Recognition System



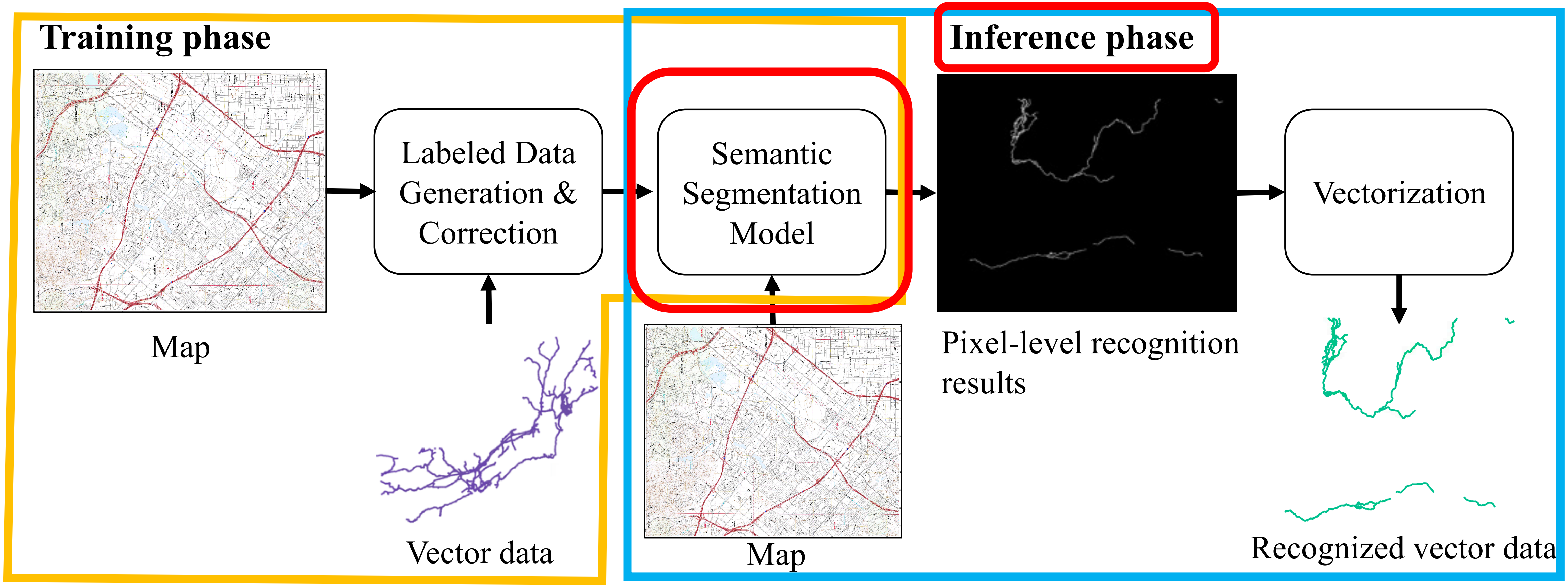
Our Geospatial Object Recognition System



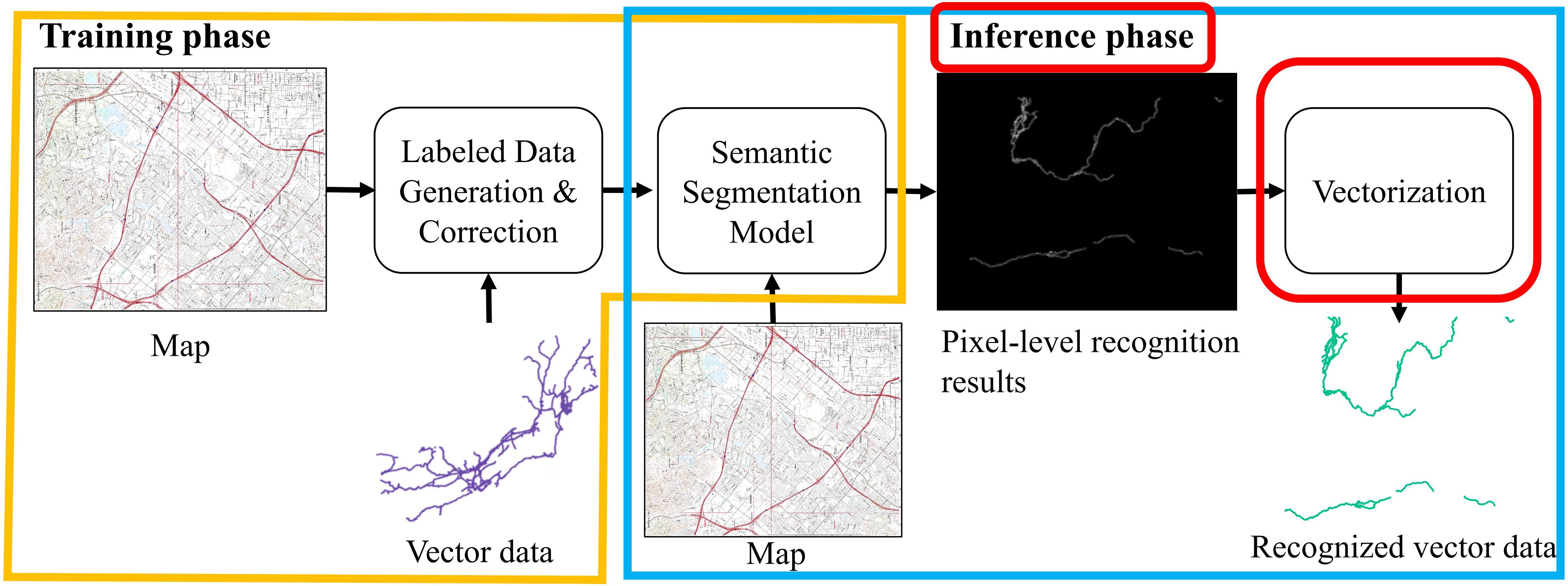
Our Geospatial Object Recognition System



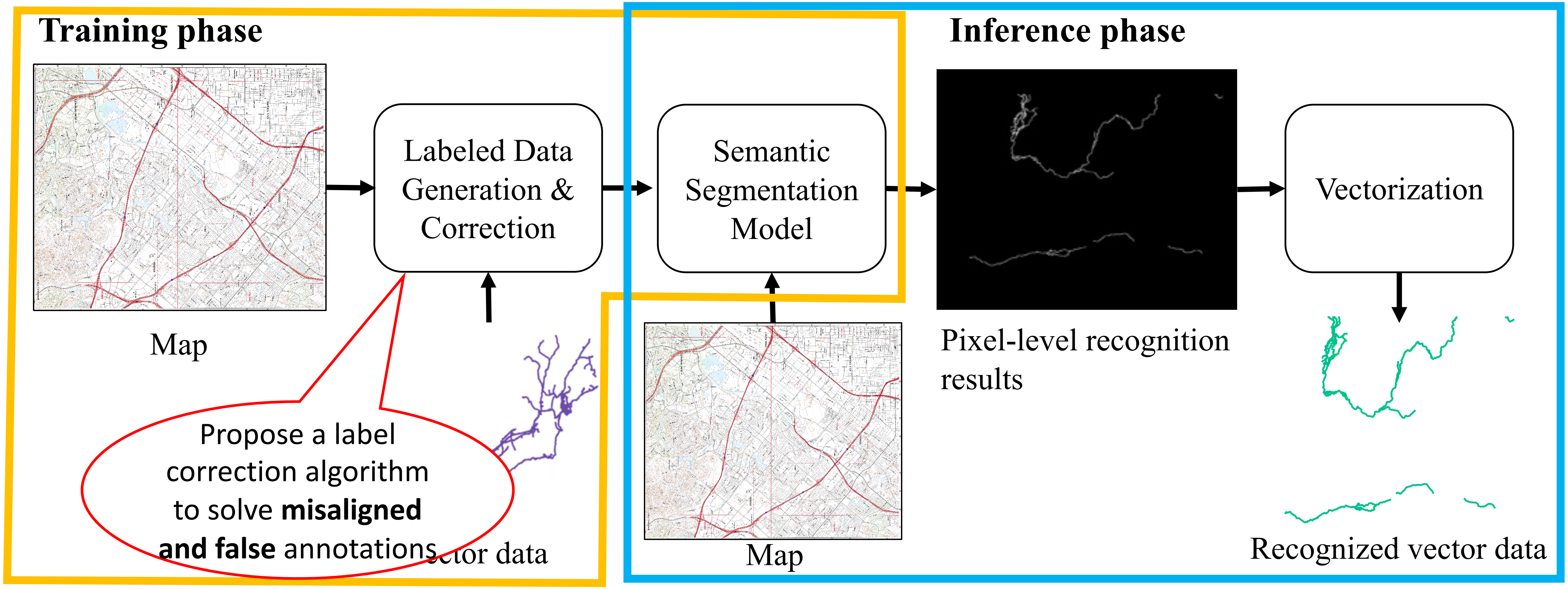
Our Geospatial Object Recognition System



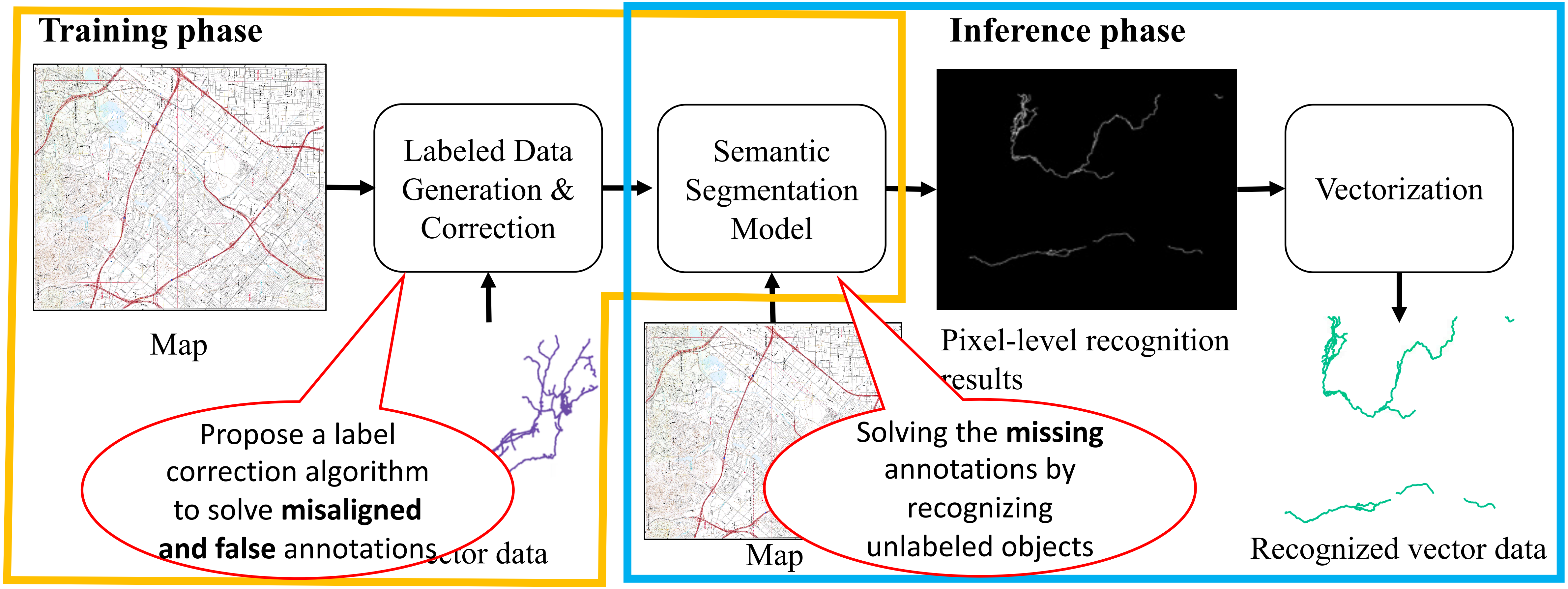
Our Geospatial Object Recognition System



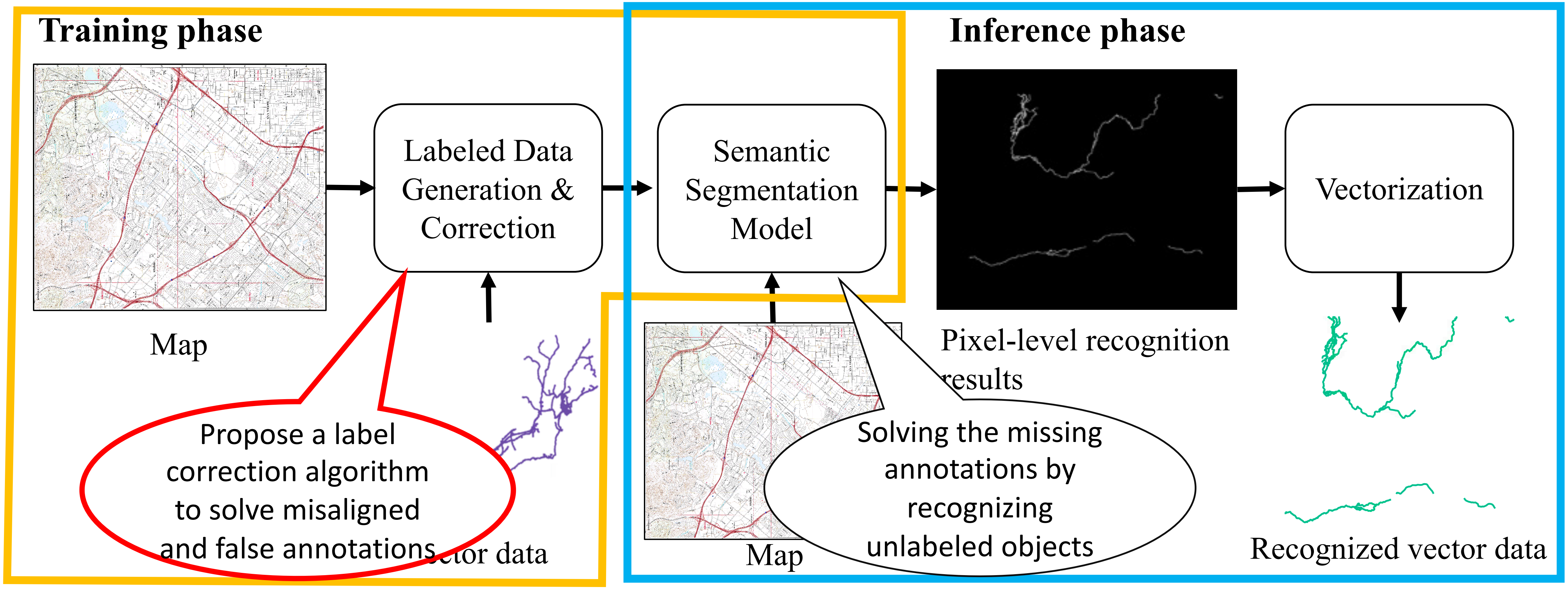
Our Geospatial Object Recognition System



Our Geospatial Object Recognition System

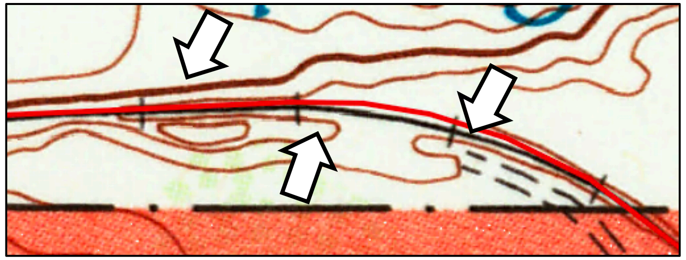


Our Geospatial Object Recognition System



What does Label Correction Algorithm want to do?

- Finding the desired group of foreground pixels in an unsupervised way
- The desired foreground pixels are the annotations for the target object
- The desired foreground pixels are close to the vector data
- But how to find the desired desired foreground (target object) pixels?
    - Shape information from the vector data

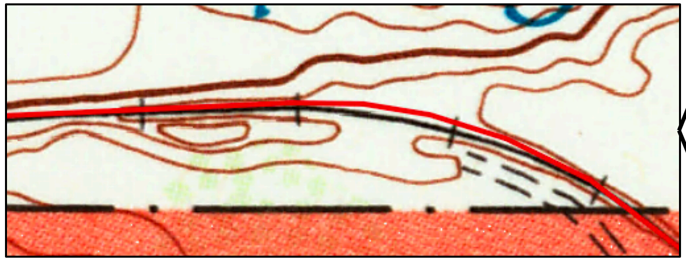


Image, the red line is the vector data

White pixels are
the desired foreground pixels

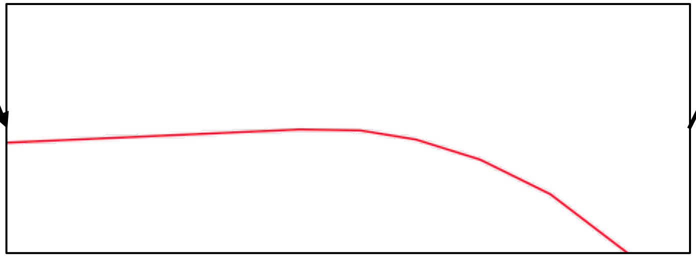
Prior Shape Information

- The external vector data provides the shape information
- But the shapes of vector and target object are not the exactly same

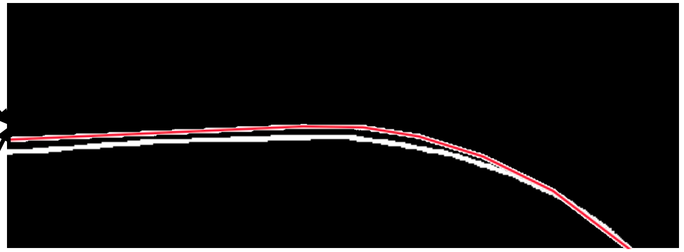


Railroads in the map and vector data (red line)

Railroad’s shape



Vector data shape



Overlapping railroads and vector data shape

Affine transformation (contribution)

- Not exactly same shape results from
  - Different projection systems or scales
- The vector data after an affine transformation and target object’s shape can be aligned in a subregion of a map

Goal of Label Correction Algorithm (LCA)

- The desired group of foreground pixels has
  - Homogeneous colors
  - The shape aligned to the vector data after an affine transformation
- Inputs: image, shape prior, and pixels-of-interest
- Level-set function represents shape prior, pixel-of-interest, and desired foreground pixels

image | shape prior | pixels-of-interest | desired foregrounds

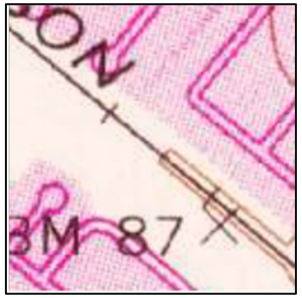
- Shape prior: buffer and rasterize the vector data
- pixels-of-interest: buffer and rasterize the vector data using large buffer

Objective Function of LCA

$$E(c_1, c_2, \Phi, L, \Psi) = \underbrace{\boxed{\int_\Omega (u - c_1)^2 H(\Phi) + (u - c_2)^2 \big(1 - H(\Phi)\big) dxdy}}_{\text{Color loss}} + \lambda \underbrace{\boxed{\int_\Omega \big(H(\Phi)H(L) - H(\Psi)\big)^2 dxdy}}_{\text{Shape loss}}$$

Color homogeneous area

- c1, c2: average color of foreground and background area
- To minimize the color loss, assign pixels the area which has smaller color difference between the pixel’s color and the average color than the other one does

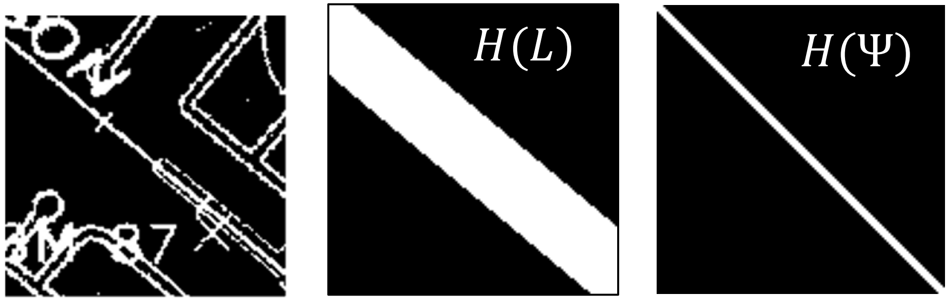


U, u(x,y): pixel color

$\mathrm{H}(\phi)$

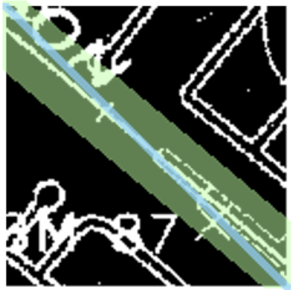
Objective function of LCA

$$E(c_1, c_2, \Phi, L, \Psi) = \int_\Omega (u - c_1)^2 H(\Phi) + (u - c_2)^2\big(1 - H(\Phi)\big)dxdy + \underbrace{\lambda \int_\Omega \Big(\overbrace{H(\Phi)H(L)}^{\text{Desired foreground pixels}} - H(\Psi)\Big)^2 dxdy}_{\text{Shape loss}}$$

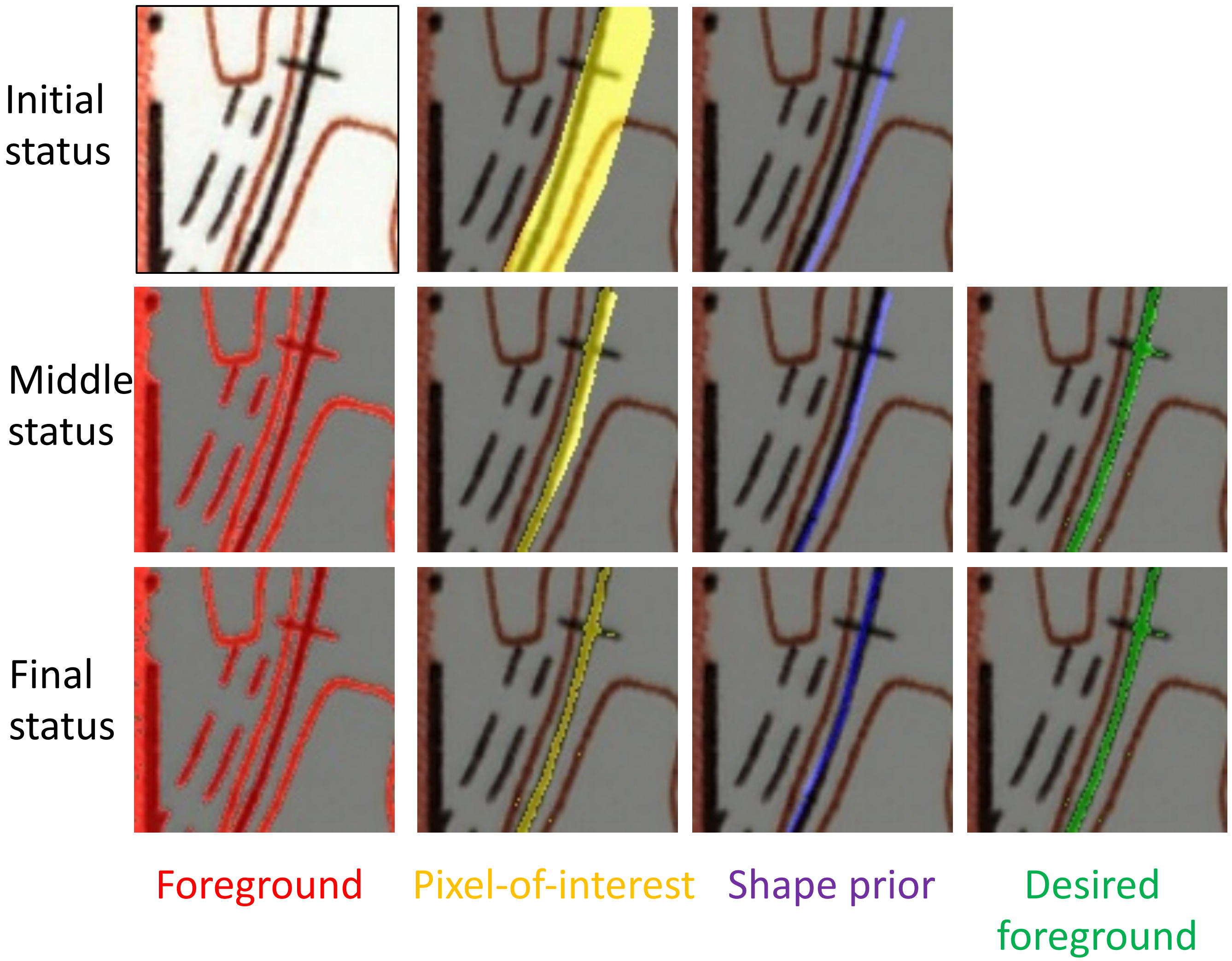


Shape similarity

- $H(L)$: pixel-of-interest
- $H(\phi)H(L)$: the desired foreground area
- $H(\Psi)$: rasterized vector data, $\Psi = affine(\Psi_0)$
- Affine transformation moves vector data to $H(\phi)H(L)$
- Non-overlapping area = Desired foreground area ($H(\phi)H(L)$) − target object's shape ($H(\Psi)$)
- Non-overlapping area ↓ Shape similarity ↑



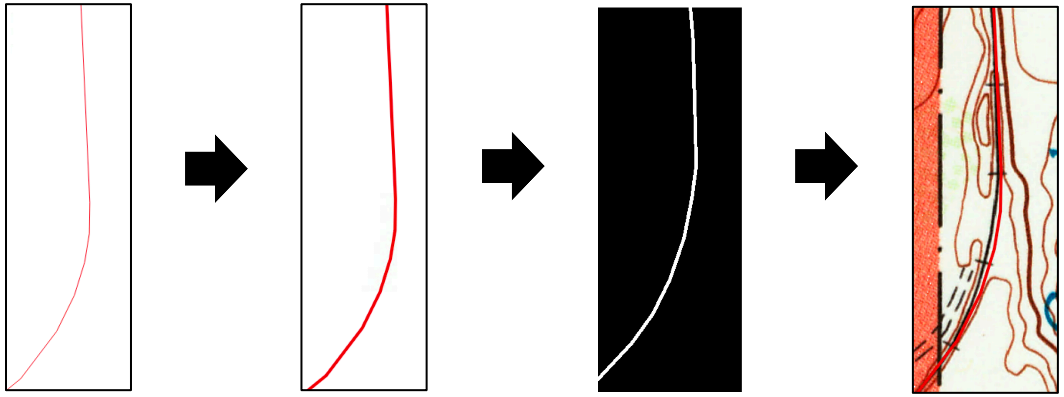
Optimization process



- Update the average colors of foreground and background pixels
- Update the pixel assignments
- Reduce the pixels-of-interest
- Update the affine transformation
- Move the vector data towards the desired foreground area

Experiment Data and Settings

- Experiment data
  - Two USGS topographic scanned maps: Bray 2001, Louisville 1965
  - Two target objects:
    - Railroads and waterlines in Bray
    - Railroads in Louisville
- Three groups of annotations
  - Original vector data: buffer → rasterize → annotate



Experiment Data and Settings

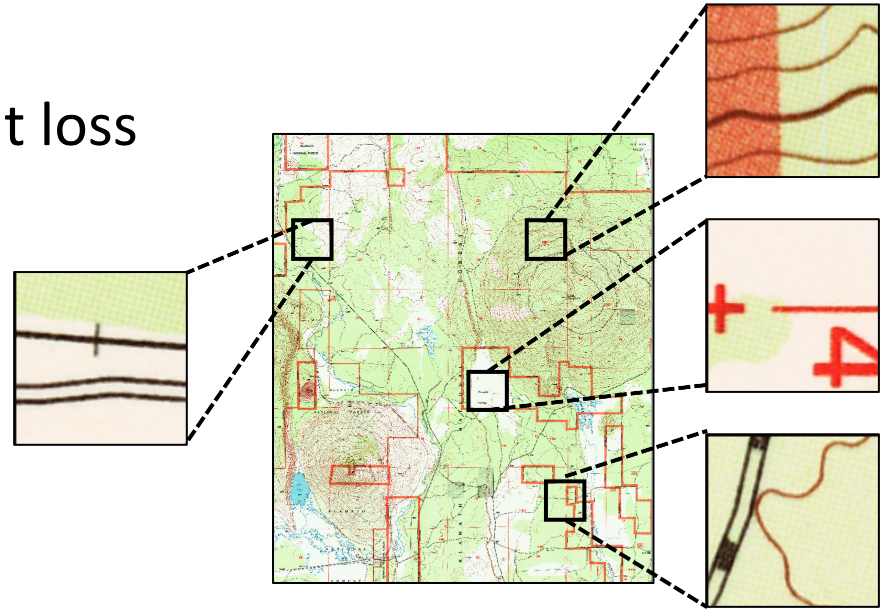
- Experiment data
  - Two maps: Bray 2001, Louisville 1965
  - Two target objects:
    - Railroads and waterlines in Bray
    - Railroads in Louisville
- Three groups of annotations
  - Original vector data: buffer → rasterize → annotate
  - Vector-to-raster: align → buffer → rasterize → annotate

Experiment Data and Settings

- Experiment data
  - Two maps: Bray 2001, Louisville 1965
  - Two target objects:
    - Railroads and waterlines in Bray
    - Railroads in Louisville
- Three groups of annotations
  - Original vector data: buffer → rasterize → annotate
  - Vector-to-raster: align → buffer → rasterize → annotate
  - LCA: the desired foreground pixels from the proposed LCA

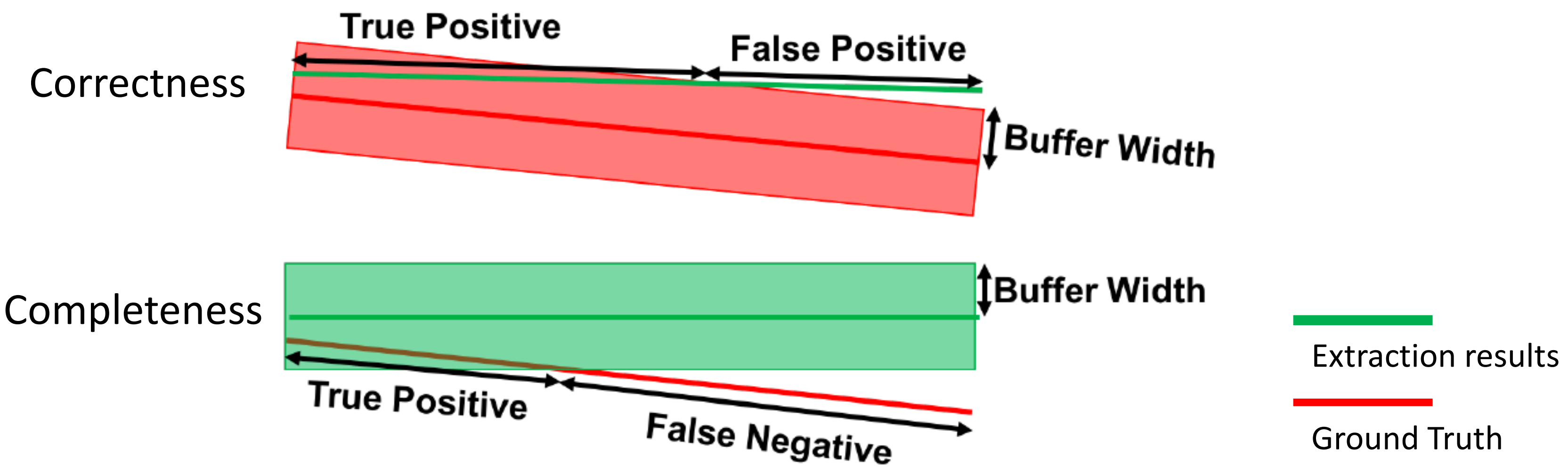
Experiment Data and Settings (cont.)

- Training data
  - 128-*128-pixel images
  - 1,000 positive (target) images, 3,000 negative (non-target) images
- Semantic segmentation model
  - Deeplabv3+
  - With and without the normalized cut loss



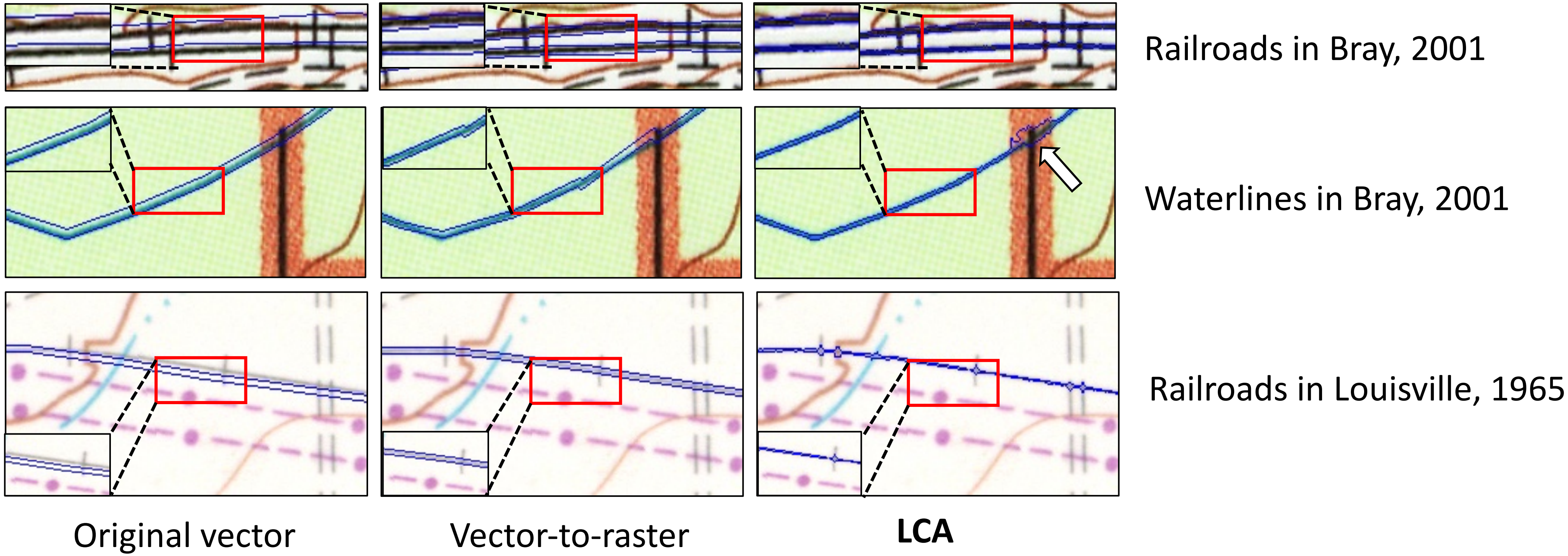
Evaluation Metrics

- Evaluate the quality of annotations
  - Pixel-level precision, recall, F_1 score
- Evaluate the recognition results
  - Correctness: how much length of recognized lines is correct
  - Completeness: how much length of target lines is recognized



Annotations Results and Analysis

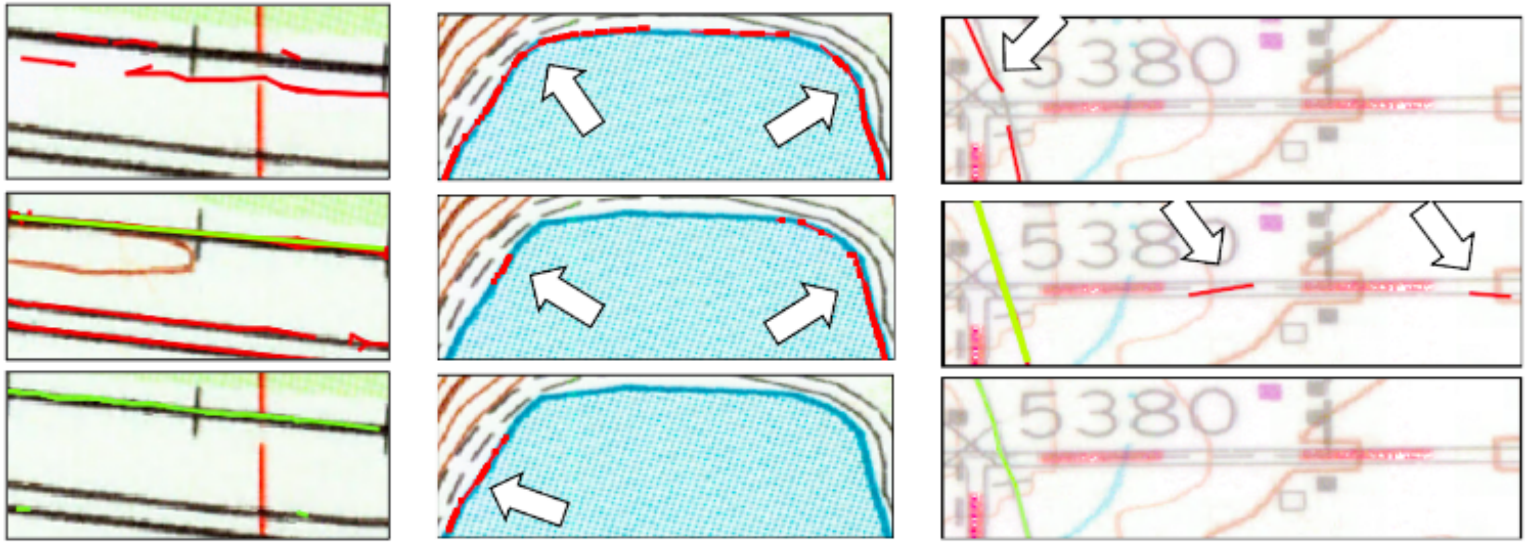
- The average precision of LCA annotations is around 10% higher than the rest two groups of annotations.



The areas within the blue outlines are the annotations for the target objects

Recognition Results and Analysis

- LCA obtain the highest correctness compared to the results from the other two annotation groups



Original vector

Vector-to-raster

LCA

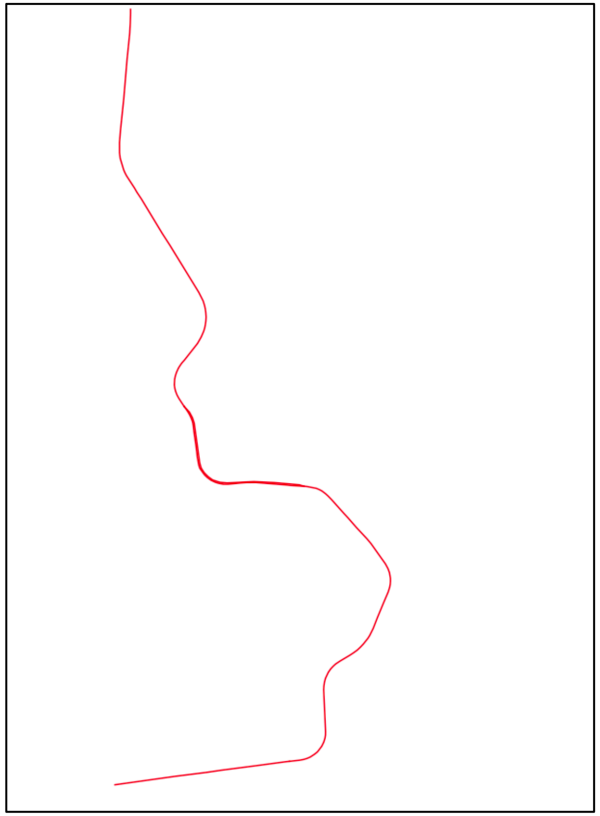
Railroads in Bray, 2001   Waterlines in Bray, 2001   Railroads in Louisville, 1965

True positive

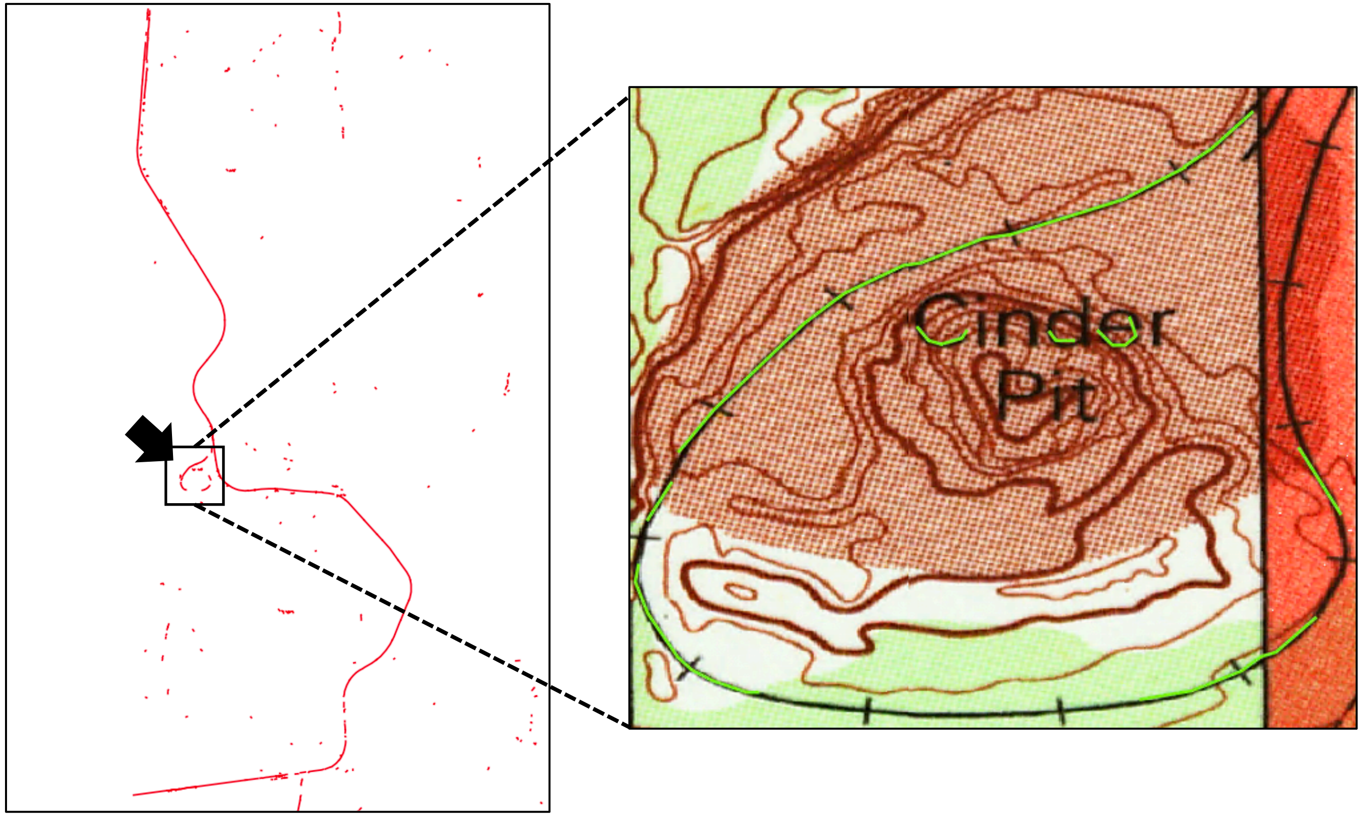
False positive

Recognition Results and Analysis (cont.)

- Semantic segmentation model recognizes unlabeled target objects



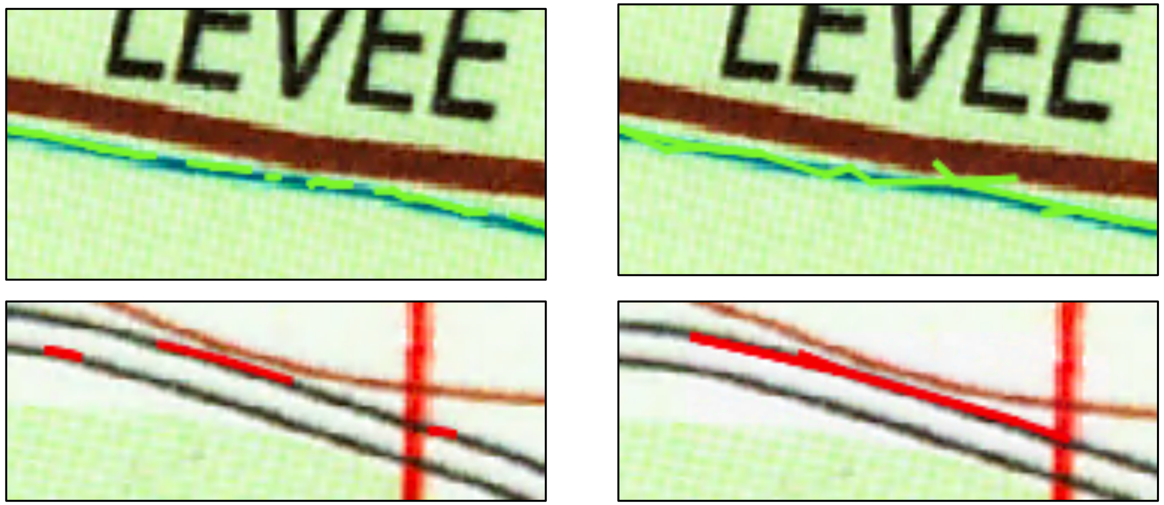
Vector data



Recognition result

Normalized Cut Loss Analysis

- The Normalized cut loss is a similarity measurement
- Assign nearby pixels with similar colors into the same category
- Normalized cut loss can reduce the gaps in the recognition results
  - True positive lines
  - But also false positive lines



Waterlines in Bray, 2001

Railroads in Bray, 2001

True positive

False positive

LCA w/t Ncut

LCA with Ncut

Summary & Future Work

- Proposed an automatic recognition system for the linear geospatial objects in the map images
- Proposed LCA reduces the misaligned and false annotations
- LCA calculates an affine transformation for vector data
- LCA improves the recognition results
- In the future, improve the connectivity of recognized lines