

Developing an Interactive Machine- Learning-based Approach for Sidewalk Digitalization

January 2018

A Research Report from the National Center
for Sustainable Transportation

Ji Luo, University of California, Riverside

Guoyuan Wu, University of California, Riverside



National Center
for Sustainable
Transportation



College of Engineering- Center for
Environmental Research & Technology

About the National Center for Sustainable Transportation

The National Center for Sustainable Transportation is a consortium of leading universities committed to advancing an environmentally sustainable transportation system through cutting-edge research, direct policy engagement, and education of our future leaders. Consortium members include: University of California, Davis; University of California, Riverside; University of Southern California; California State University, Long Beach; Georgia Institute of Technology; and University of Vermont. More information can be found at: ncst.ucdavis.edu.

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the United States Department of Transportation's University Transportation Centers program, in the interest of information exchange. The U.S. Government and the State of California assumes no liability for the contents or use thereof. Nor does the content necessarily reflect the official views or policies of the U.S. Government and the State of California. This report does not constitute a standard, specification, or regulation. This report does not constitute an endorsement by the California Department of Transportation (Caltrans) of any product described herein.

Acknowledgments

This study was funded by a grant from the National Center for Sustainable Transportation (NCST), supported by USDOT and Caltrans through the University Transportation Centers program. The authors would like to thank the NCST, USDOT, and Caltrans for their support of university-based research in transportation, and especially for the funding provided in support of this project.

The authors would like to thank Mr. Alexander Vu and Dr. Kanok Boriboonsomsin from the UC Riverside Center for Environmental Research and Technology for their technical support. We greatly appreciate Mr. Nathan Mustafa from the City of Riverside, Mr. Dustin Foster from Caltrans, and Cynthia Garcia from the California Air Resource Board for their valuable inputs and feedback. Last but not least, we would like to thank our project manager Mr. Patrick Tyner for his support through the project.



Developing an Interactive Machine-Learning-based Approach for Sidewalk Digitalization

A National Center for Sustainable Transportation Research Report

January 2018

Ji Luo and Guoyuan Wu

College of Engineering - Center for Environmental Research and Technology, University of California, Riverside



[page left intentionally blank]

TABLE OF CONTENTS

| | |
|--|----|
| 1. Introduction | 1 |
| 2. Methodology Overview | 2 |
| 3. Sidewalk Network Mapping | 4 |
| 3.1 Preprocessing..... | 4 |
| 3.2 Mapping Vnet to Pnet..... | 6 |
| 4. Development of Image Processing Algorithm Based on Interactive Machine Learning | 10 |
| 4.1 Sweeping Images | 10 |
| 4.2 Labeling Images..... | 11 |
| 4.3 Training Machine Learning Parameters..... | 12 |
| 5. Case Study and Performance Evaluation | 13 |
| 5.1 Preprocessing and Initial Mapping | 13 |
| 5.2 Sweeping, Labeling Images, and Training..... | 13 |
| 5.3 Training Results and Next Steps | 14 |
| 5.4 Future Work | 16 |
| References | 17 |

Developing an Interactive Machine-Learning-based Approach for Sidewalk Digitalization

1. Introduction

In urban areas, many socio-economic concerns have been raised regarding fatal collisions, traffic congestion, and deteriorated air quality due to increased travel and logistic demands as well as the existing on-road transportation systems. As one of the promising remedies, active transportation has been advocated, which may not only mitigate congestion on local streets, but also promote physical fitness, foster community livability, and boost local economy (i). To promote the active transportation mode, extensive work has been focused on planning and developing a number of pedestrian and bicyclist related programs which require the infrastructure, e.g., sidewalks, as a premise (ii). A significant amount of these efforts have to go for the setup, maintenance and evaluation of the sidewalk inventory on a relatively large geographic scale (e.g., citywide, statewide), which lays a solid foundation for a variety of active-mobility-focused applications and related research, for example:

- *Improved Location-awareness Service.* As illustrated in Figure 1, the state-of-the-art navigation tools (e.g., Google Maps [<https://www.google.com/maps>]) rely on the roadway network which is designed for vehicles to guide pedestrians. Some navigation instructions can be confusing and even pose safety risks to those vulnerable road users, since a portion of the path may not be walkable or be in conflict with motor vehicles. In such cases, a dedicated sidewalk network becomes necessary.



Figure 1. A comparative example of pedestrian navigation using (a) existing roadway network; and (b) potential sidewalk network.

- *Crowdsourcing Based Sidewalk Inventory Maintenance and Update.* As aforementioned, a large-scale sidewalk inventory will facilitate the planning of new sidewalk construction, and maintenance or improvement of existing sidewalks (iii). For example, based on the sidewalk network, active travelers and traffic engineers can identify or report damaged sidewalks and share the locations of potential safety risks in a timely and cost-effective manner.

Conventionally, transportation engineers and researchers have to rely on laborious field measurements to conduct sidewalk survey and assessment (*iv, v*), which is rather resource consuming (in both time and cost). Recently, a few studies attempted to digitize sidewalks as a part of geographic information system (GIS), created the sidewalk inventory under restricted conditions (*vi*), and assessed the quality of sidewalk (*vii, viii*). However, most of the existing methods for sidewalk system digitization are neither comprehensive nor cost-effective.

On the other hand, due to the rapid advances in computational capability and explosion of data availability, machine learning techniques have shown great potential for image recognition and classification (*ix*). One heuristic (brute-force) way to extract the information of sidewalks is to process every piece of satellite/aerial images (with appropriate size and resolution) for the region of interest. However, it may be overwhelming to prepare the image set (for both training and analysis), especially for a large geographic scale. In addition, it would be also very challenging to develop a reliable and effective algorithm to identify the sidewalk from a large mix of images on different facilities.

To address the above issues, we propose a machine-learning-based (*x, xi*) sidewalk digitization method which should be much more reliable and cost-effective than the brute-force one. The basic idea is to take full advantage of roadway networks to reconstruct an initialized (connected) sidewalk network. Then, an image sweeping script is developed to extract a large number of sidewalk images along the initialized sidewalk network. Thirdly, a machine learning technique is applied to the aerial images of focused areas (i.e., surrounding zones along the initial sidewalk network) to identify whether a sidewalk is present or not. It is noted that the category of sidewalk (e.g., landscape/lawn, parking lot/ driveway, crosswalk) may be also recognized if there were sufficiently large training dataset.

The rest of this report is organized as follows: Section 2 will give an overview of the proposed methodology for sidewalk digitization. Section 3 and 4 will introduce the mapping and categorizing algorithms in details. Section 5 will present the application of the method on streets in Riverside City, and evaluate the performance of the method. The last section concludes this report with a discussion on potential future work.

2. Methodology Overview

By a general definition, sidewalks are to accommodate pedestrians at a Level of Service (LOS) equal to that of vehicles using the roadway (*xii*). In an urban or suburban environment, the sidewalk segments usually exist parallel to the vehicle roadways and are largely associated with the roadway network. In addition, it is difficult to predict whether the sidewalk sections are present or not merely based on the surrounding roadway and land use information. Therefore, it is of great interest to classify the initialized sidewalk sections into 'paved' (concrete-surface present) and 'sidewalk missing' (concrete surface do not exist) categories as a first attempt. In

this report, unless otherwise noted, the phrases ‘paved’ and ‘concrete-surfaced’ are interchangeable, and ‘missing’ means that the sidewalk surfaces do not exist.

In this study, we propose to map the features of sidewalks based on the roadway network as the first step. The roadway network data applied in this study should include road link attributes and position coordinates, such as roadway shapefiles (xiii). Secondly, we will write a Python script to sweep each link in the initialized sidewalk network, bound a rectangular area which includes the link and extracts the aerial image within that area. In parallel, we will extract a large number of aerial images of sidewalk network, for example, ESRI ArcMap aerial basemaps (xiv), and set up a machine learning algorithm to learn from the labeled (‘paved’ and ‘not paved’) images. We will train the machine learning classifier to be able to predict a new sidewalk image at a reasonable prediction rate. Then the classifier can be used to predict the surface attributes of the extracted image using the trained machine learning algorithm. The overall method can be illustrated in Figure 2. Here, we name the ‘Vehicle roadway network’ as ‘Vnet’, and ‘Pedestrian sidewalk network’ as ‘Pnet’ for convenient reference.

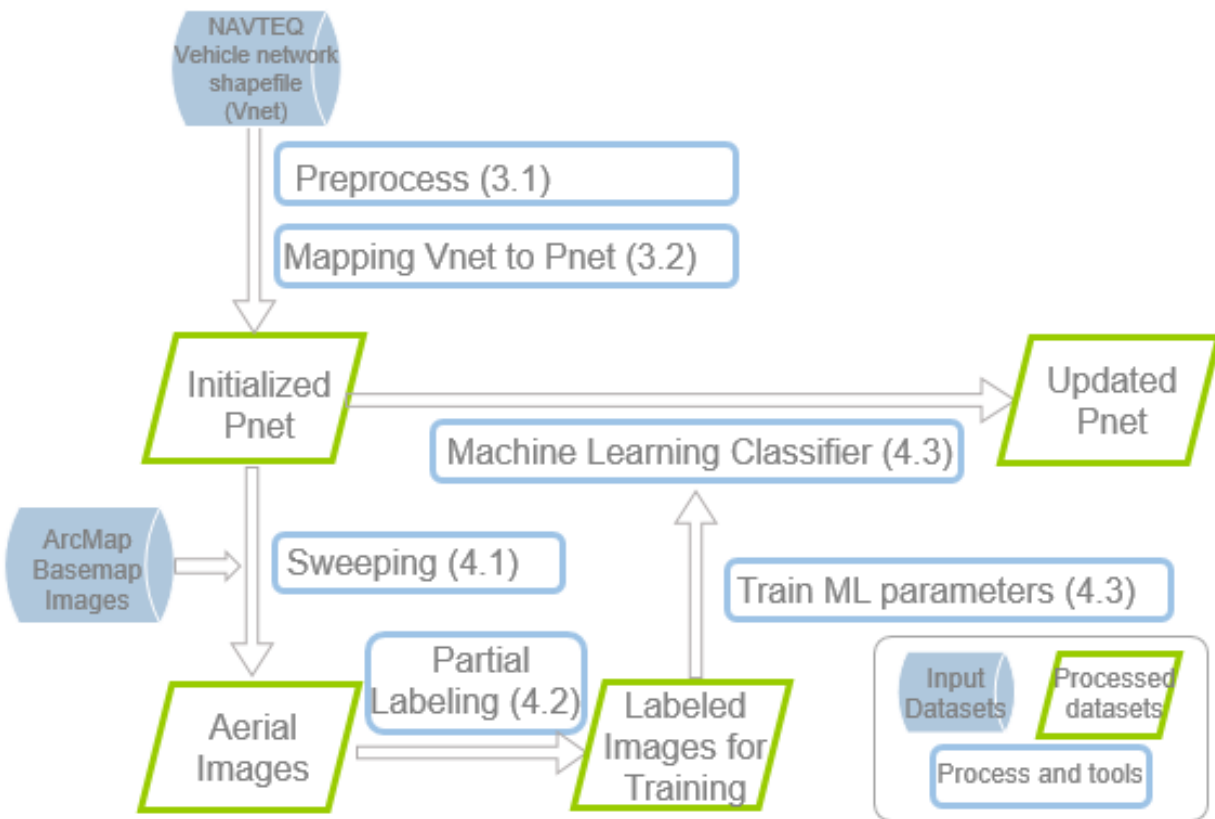


Figure 2. Flow chat of overall methodology.

3. Sidewalk Network Mapping

3.1 Preprocessing

The preprocessing of Vnet aims at filtering out unnecessary roadway links, representing the road curvatures with straight segments, and extracting the graph table of the network. We recommend using ESRI ArcMap software for the geographic processing and all features are projected to Universal Transverse Mercator (UTM) coordinate system (xv , xvi). The preprocessing includes the following steps:

1. We assume that sidewalks do not exist by side of freeways and freeway ramps. Therefore, freeways and ramps are removed from the Vnet. Conventional highways in the State of California could be included in the Vnet because they are accessible to pedestrians.
2. Step Two is to prevent duplicating sidewalk links. If one section of arterial is represented with two edges (e.g. an arterial has an island in the middle and the vehicle traffic of two directions are represented with two parallel links), only one consistently edge along the road should be kept and the other one edge should be removed. This method will result in a shift of the roadway centerline and potentially cause trouble to capture the correct sidewalk image in Section 4.1. A better solution is proposed in Section 5.3.
3. Generalize the filtered Vnet, and split links at vertices to make each link a straight segment. In Figure 3a, the original curved link is split into 11 segments using 'Split line at vertices' ($xvii$) tool in ArcMap. In Figure 3b: After using 'Generalize' tool ($xviii$) on the original link, the geometry of the curve is simplified and it only produces 6 links using 'Split line at vertices' tool, hence reducing the number of links to process. Users need to specify a generalization tolerance when using this tool; the larger the tolerance is, the more simplified the results are. The generalization not only reduces the number of links up to 20% but also preserve the geometry of the network.



a. a curved link is split into 11 straight links

b. after generalization the curved link is split into 6 links

Figure 3. ‘Generalize (18)’ and ‘Split Line at Vertices (17)’ tools used to simplify and preserve link geometry

- In this step, we need to build the Vnet graph based on the preprocessed Vnet shapefile. The graph is a mathematical representation of the road network. A relatively fast method is to apply ArcMap’s ‘Network Analysis’ tool (xix, xx). The ‘Network Analysis’ tool will output the unique junctions that connect the entire network, and the coordinates of the junction points can be extracted using ‘Add XY’ function in ArcMap (xxi). Then ‘Spatial Join’ tool (xxii) can be utilized to match each link with its two nodes. Because each link is already processed to be a straight segment as described in the last section, each link should have and only have two nodes. The example link table and node table are shown in Table 1.

Table 1. Example Network Graphs

a. Example Link Table

| Vnet LinkID | Node1ID | Node2ID | Road Width (m) |
|-------------|-----------|-----------|----------------|
| 1 | 2586 | 2601 | 10 |
| 2 | 2601 | 2617 | 10 |
| 3 | 1822 | 1932 | 10 |
| 4 | 2085 | 2176 | 10 |
| 5 | 2070 | 2085 | 10 |
| ... | ... | ... | ... |
| $Link_j$ | $Node1_j$ | $Node2_j$ | $Width_j$ |

b. Example Node Table

| Vnet NodeID | NodeX | NodeY |
|-------------|----------|---------|
| 1 | 460188.2 | 3758112 |
| 2 | 460268.5 | 3758106 |
| 3 | 460323 | 3758103 |
| 4 | 460331.3 | 3758089 |
| 5 | 460356.3 | 3758100 |
| ... | ... | ... |
| $Node_i$ | X_i | Y_i |

- Assign road width and other critical values to roadway links. Detailed information about Vnet road width and number of lanes is not widely available for local roadways in California, therefore the values need to be estimated. In this study, the road width was estimated based on the road names provided by NAVTEQ Streets (xxiii). The roadway names are recorded with the name and the road type, for example, 'Magnolia' and 'AVE', 'Van Buren' and 'BLVD'. Based on the suffix of road, a general road width was assigned to each link. For example, roads with 'AVE', 'CIR', 'CT', 'DR', 'CN', 'PL', 'VLG', 'WAY', 'TRL', 'TER' were given a 10-meter road width, and roads with 'BLVD' were given an 18-meter width. A number of exceptions were made based on the knowledge of the local streets and survey of Google Maps. This step should be built on local road condition and there are various resources available for surveying, e.g. Google Maps and Google Earth.

3.2 Mapping Vnet to Pnet

With the preprocessed Vnet, we can now prepare to map the preliminary sidewalk nodes. We assume that sidewalk segments are present on both sides of a Vnet link, and the sidewalk links are all connected with sidewalk nodes in Pnet as in Vnet. As shown in Figure 4a, our goal is to calculate the coordinates of four preliminary Pnet nodes ($P1$, $P2$, $Q1$, $Q2$) based on the two Vnet nodes ($X1$, $Y1$, $X2$, $Y2$) of one roadway link ($Link_j$) and the roadway link's half width (d , roadway centerline to the edge of sidewalk) which is estimated in Section 3.1.

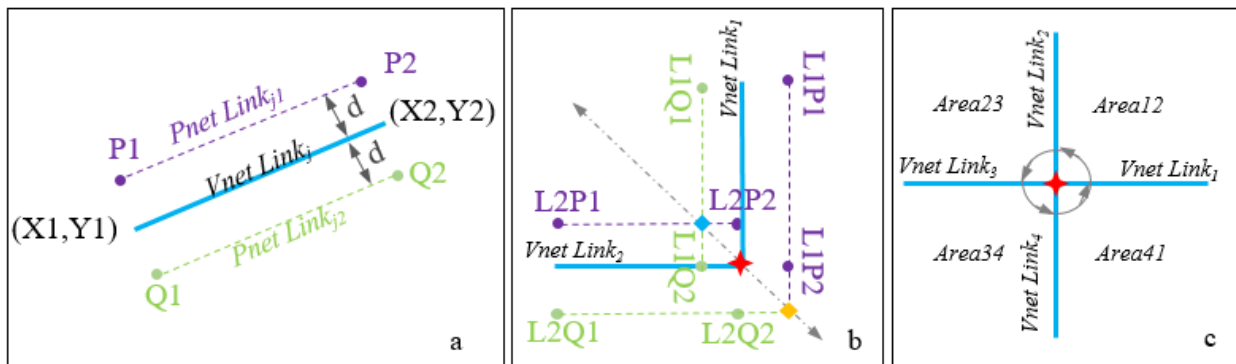


Figure 4. Schematics of mapping the preliminary sidewalk nodes

$X1$, $Y1$, $X2$, $Y2$ are Cartesian coordinates (Universal Transverse Mercator) of the link nodes. We should guarantee that point $P1$ and $P2$ will always be at a predictable side of their original Vnet link in order to calculate the correct merged sidewalk node that is shown as a blue/orange diamond in Figure 4b. In this case, we specify that a Pnet link's $P1$ and $P2$ must fall within the upper or right-hand side of its original Vnet link, as shown in Figure 4a and 4b. In order to control the relative position of $P1$, $P2$, $Q1$, $Q2$, the following steps are applied. When $X1$ equals $X2$ and $Y1$ is smaller than $Y2$, or when $X1$ is larger than $X2$, we swap the values of $(X1, Y1)$ with

$(X2, Y2)$ to guarantee that the $(X1, Y1)$ is at the left side or at the top of $(X2, Y2)$. Then the coordinates of $P1, P2, Q1,$ and $Q2$ are calculated by the following formulas:

$$P1x = X1 - d/L(Y2 - Y1), P1y = Y1 + d/L(X2 - X1) \quad (1)$$

$$Q1x = X1 + d/L(Y2 - Y1), Q1y = Y1 - d/L(X2 - X1) \quad (2)$$

$$P2x = X2 - d/L(Y2 - Y1), P2y = Y2 + d/L(X2 - X1) \quad (3)$$

$$Q2x = X2 + d/L(Y2 - Y1), Q2y = Y2 - d/L(X2 - X1) \quad (4)$$

d is the assumed distance from road centerline to the edge of the sidewalk. L is the link length. Since all the links are processed to be straight segments, L can be calculated as:

$$L = \sqrt{(X1 - X2)^2 + (Y1 - Y2)^2} \quad (5)$$

Next, we need to process the preliminary sidewalk links at intersections. As shown in Figure 4b, when two Vnet links intersect at a Vnet node (marked by a red star), the Pnet links generated based on previous rules (shown in purple and green broken lines) will also intersect. We aim at calculating the 'merged sidewalk node' (marked in blue and orange diamonds) based on the Pnet nodes, which also form the intersections nodes at street crossings.

The general method is to iterate through all the nodes in Vnet. For each Vnet node (e.g., the red star in Figure 4c), we rank all the connected Vnet links by the relative angle to X-axis. Referring to Figure 4c, the four links can be ranked as $Link1, Link2, Link3,$ and $Link4$. In $Area12$, we can locate which Pnet link of $Link1$ will intersect with the Pnet link of $Link2$ based on the coordinates and the angle bisector formed between $Link1$ and $Link2$. Then, the intersection coordinates can be computed, and the coordinates of preliminary Pnet nodes can be updated. The pseudocode for this process in Matlab can be shown as:

```
function ProcessPreliminaryPnetNodes(VnetGraph);
%calculate the coordinates of Pnet Node (P1,P2,Q1,Q2) for all Vnet links in the Vnet
initialize Pnet Node based on Equation (1) to (5)
Pnet Node= Vnet Link ID, Vnet Node ID, new Pnet LinkID, new Pnet NodeID, P1x, P1y, P2x, P2y, Q1x,
Q1y, Q2x, Q2y

for each Vnet Node in VnetGraph
    search number of Vnet links n connected with Vnet Node i

    if n==1
        keep Pnet Nodes the same
    else
        rank n links by their relative angle to x-axis

        for j=1:n
            extract coordinates of (P1,P2,Q1,Q2) for link(j) and (P1,P2,Q1,Q2) for link(j+1)
```

search the area between link(j) and link(j+1)
locate the intersection of Pnet of link(j) and Pnet of link(j+1) based on coordinates of
(P1,P2,Q1,Q2)
calculate the coordinates of Pnet intersection: interX, interY
mark the intersection point as jth crossing point for crosswalk generation
update the Pnet ID of the intersected P or Q point to be the same
update the coordinates of the intersected P or Q points with interX, and interY

return Pnet Node

If we assume there is a total of J links ($link1, link2, \dots, linkj$) in Vnet Graph, then the mapping method will yield $4J$ preliminary sidewalk nodes, and $2J$ preliminary sidewalk links. Next, we assume there are m nodes in the Vnet, among which x out of m nodes are connected with only one link, and the other $(m-x)$ links are connected with j links ($n \geq 2$). Then after the execution of the pseudocode above, the total number of Pnet sidewalk nodes will become $2x + \sum_{m-x} j$.

An example output of the pseudocode above is illustrated in Figure 5, where the preliminary sidewalks are shown in green line, and sidewalk nodes are shown in blue dots.

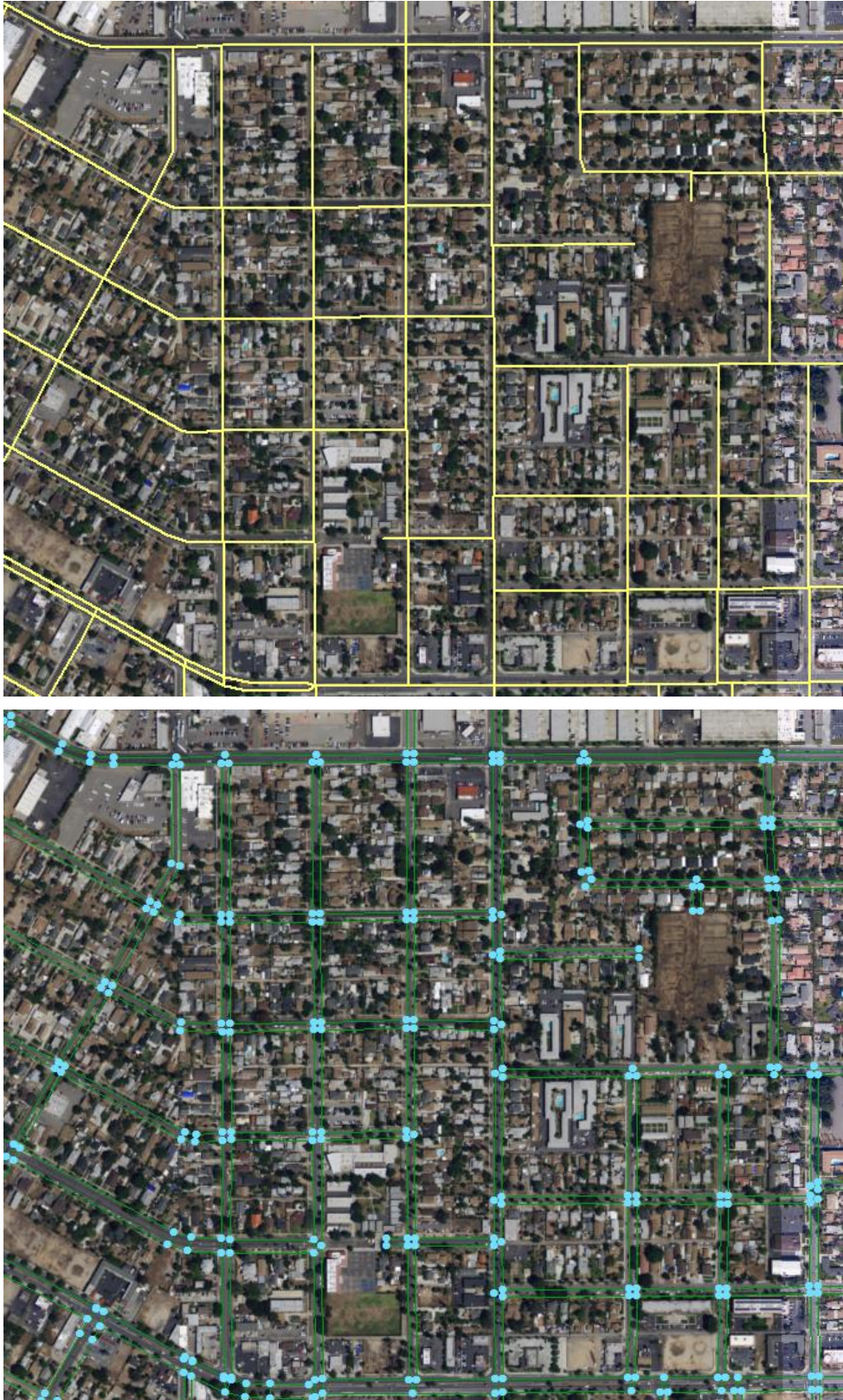


Figure 5. An example of initialized sidewalk calculated based on vehicle roadway network.

4. Development of Image Processing Algorithm Based on Interactive Machine Learning

With the initialized sidewalk network, it becomes important to be able to identify whether a certain segment is concreted or not. When viewing the aerial images of the street at a zoomed-in level, most unobstructed paved and missing sidewalk sections can be easily identified with human eyes, as shown in Figure 7 and Figure 8. To enable a large number of image collection, we developed an image sweeping method to capture the aerial images along the Pnet. In order to train the learning algorithm to identify the 'sidewalk surface' feature, we utilized a regularized logistic regression learning algorithm in our first attempt. The overview of the training process is illustrated in Figure 6.

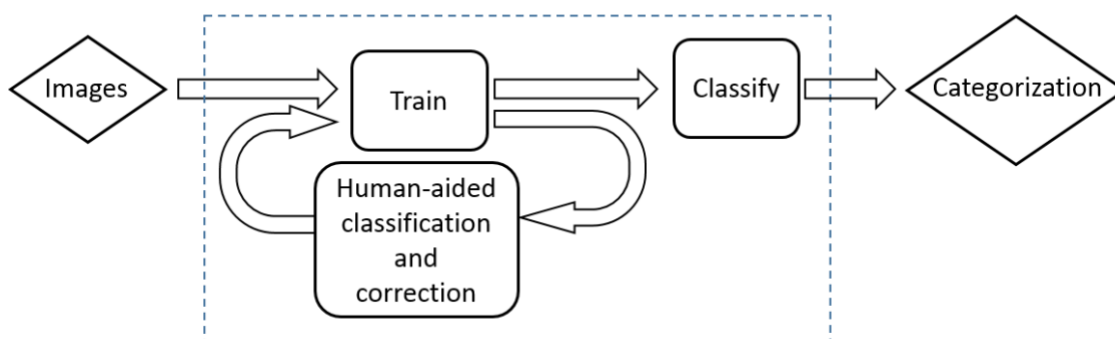


Figure 6. An overview of machine learning training process

4.1 Sweeping Images

In order to collect images that can be trained by a machine learning algorithm, the images should be of the same size and capture the objects that are desired to be recognized by the algorithm. As shown in Figure 5, collecting images along the Pnet links involves a large number of screenshot operations and it will not be practical to manually perform the task.

With the aid of ArcPy package (xxiv), we are able to develop a Python script and run it in ArcMap Python Environment. The Python script will zoom into each Pnet link at a designated map scale, and take screenshots at designated image size. The pseudocode is as follow:

```
SweepingImage(PnetGraph);
```

```
import arcpy, arcpy.mapping, pyautogui, numpy  
set map scale, image size and other paramters
```

```
for each Pnet Link in PnetGraph:
```

```
    extract the two Pnet node coordinates (Px1, Py1, Px2, Py2) of the Pnet Link
```

```
    calculate the angle between the Pnet link and x positive axis based on (Px1, Py1, Px2, Py2)
```

rotate the map frame with respect to the angle in order to position the link parallel to x axis
pan the map frame at the designated scale and center the Pnet Link
calculate the screen pixel location based on the screen resolution, screen size, and the Pnet Link location
screenshot and save an image of the designated size

Even though the script has reduced manual operation time, it is the most time-consuming process in this study. For example, with a desktop computer of Intel Core i3-2120 CPU (@3.3GHz, 8 GB RAM), capturing 1,000 screenshots will take approximately 88 minutes.

4.2 Labeling Images

After collecting a large number of aerial images of the initialized sidewalk segments, we need to label a subset of them in order to develop the classifier. In this study, labeling images means that we assign a category for a group of images which share similar attributes. Specifically, as shown in Figure 7, a label of ‘paved sidewalk’ is assigned to a group of images which clearly present concrete-surfaced sidewalk segments. On the other hand, in Figure 8, the label of ‘missing sidewalk’ is assigned to a group of images which can be identified as a lack of paved sidewalk segments.



Figure 7. Screenshots of paved sidewalk sections from ESRI aerial image at 1:300 scale.



Figure 8. Screenshots of unpaved roadside sections from ESRI aerial image at 1:300 scale.

Generally, the more well-labeled images there are, the better the training results will be. Usually, image classification requires a large number of training samples. For example, Coates *et al.* collected more than 20,000 photograph text samples in order to train a character recognition program (xxv). The images are usually represented with a vector of values, which corresponds to the grey scale value of each pixel in the image. For example, if one image is of size 200-by-400 pixel, the image will be represented with an 80,000-element vector. At the same time, this vector will be associated with a label, in this case, '1' for 'paved sidewalk' or '2' for 'missing sidewalk'. The labeled images will be critical inputs for the next training step.

4.3 Training Machine Learning Parameters

With a large number of labeled images, we will be able to use a large portion (e.g. 75%) of the images for training samples and use the rest of the labeled images (25%) for cross validation. The learning algorithm first calculates a cost function which quantifies the "error" between the predicted and labeled results, as shown in Equation (6),

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left[-y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2 \quad (6)$$

$$x = X * \theta \quad (7)$$

$$h_{\theta}(x) = \frac{1}{1+e^{-x}} \quad (8)$$

where m denotes the number of training samples, for example, 1,000 images. i denotes each sample from 1, 2, ..., to m . n is the number of elements in one training sample, e.g. 80,000 in the aforementioned example. j denotes each element, e.g., 1, 2, ..., to 80,000. X is the matrix of input training samples, e.g., a 1000-by-80000 matrix that represents the 1,000 images. y is the labeled vector, e.g., a vector of m -by-1 which is comprised of number '1' and number '2', where '1' stands for 'paved sidewalk' and '2' means 'missing sidewalk'. *Lambda* λ is the regularization parameter that usually ranges from 0.1 to 3. In Equation (7), ϑ (*theta*) is the parameter set that we aim to train and aim to yield good prediction results. *theta* is initialized to be all zeros.

Next, for each training sample (image) in the X , the learning algorithm will update the values in *theta* with the aim that the cost function $J(\vartheta)$ will decrease (or not increase) after every iteration. The pseudocode of learning algorithm (25, xxvi) can be expressed as:

```

function RegLogisticRegression (X, y):
  initialize  $\vartheta$ 
  set lambda
  set number of max_iteration
  while iteration < max_iteration
  for each class // in this study, there are two classes: 'paved' and 'missing'
    calculate  $J(\vartheta)$ 
    update  $\vartheta$  to reduce  $J(\vartheta)$ 
  return  $\vartheta$ 

```

5. Case Study and Performance Evaluation

5.1 Preprocessing and Initial Mapping

For a case study, we selected an area surrounding UC Riverside shown in Figure 9. There are 4,385 roadway links in the Vnet shown in Figure 9. The preprocessing and initial mapping were performed smoothly with the processes described in Section 3. The initial mapping created 14,806 sidewalk links, including 8,774 sidewalk segments and 6,032 crosswalks. The preprocess took approximately 2 hours and the initial mapping process only took a few minutes. The initialized sidewalk map can be viewed at <https://arcg.is/184PbK>.

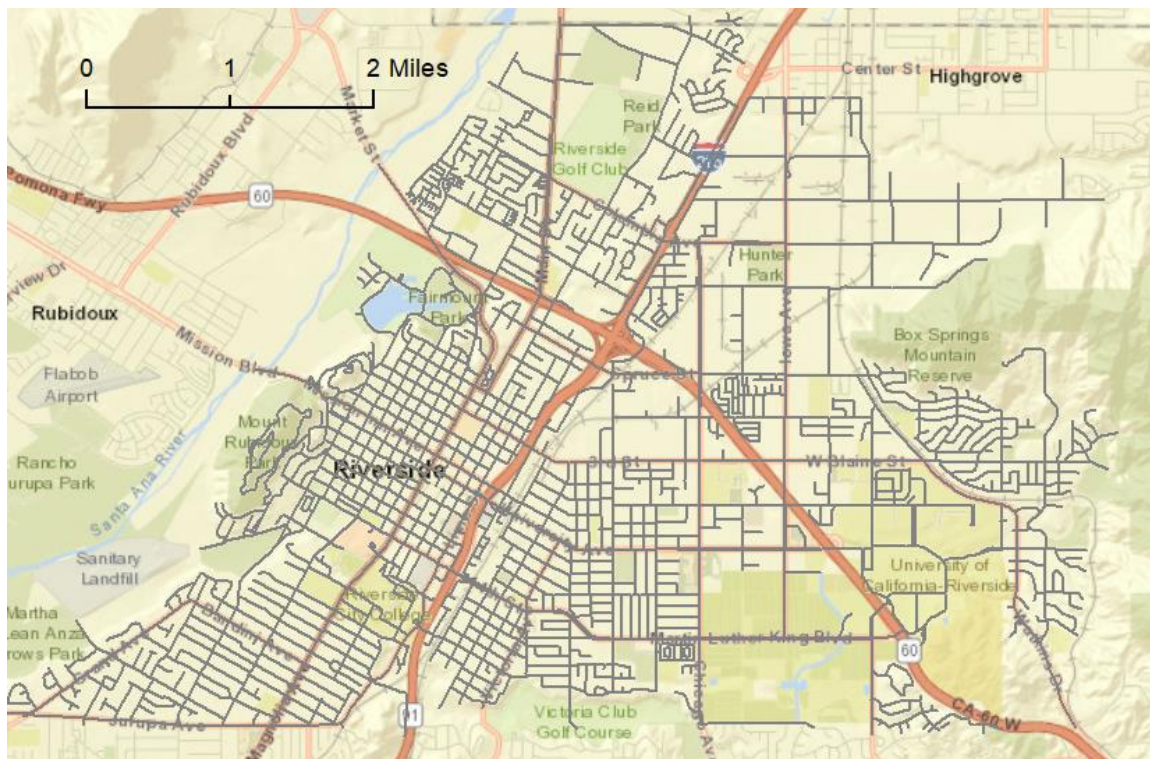


Figure 9. Example roadway network in the case study.

5.2 Sweeping, Labeling Images, and Training

As stated in Section 4.1, image sweeping is the most time-consuming process in this project. There are 8,774 sidewalk segments and their aerial images need to be captured (crosswalks are not swept because crosswalks are usually a part of vehicle roadways and do not have a concrete surface like regular sidewalks). Sweeping 8,774 images took approximately 12 hours.

We marked 804 images with “paved” label and 1051 images with “missing” label. Each image captured a 50-by-16-meter area and was of size of 603-by-192 pixels. Specifically, we applied Matlab to read all the images, convert them to gray-scale, and reshape each image into a

115,776-element vector, where each element represents the intensity of the gray-scale color of the pixel in the image.

Among the 1,855 labeled images, we planned to use 1,392 images (603 ‘paved’ and 789 ‘missing’) as training examples and 463 images (201 ‘paved’ and 262 ‘missing’) as cross validation. Recall Equation (7), the input X is a 1392-by-115776 matrix that represents the 1,392 images for training. Then, we constructed a vector of the labels, with ‘1’ representing ‘paved’, and ‘2’ as ‘missing’ based on previous labeling results. Then we input the training matrix (X) and the label vector (y) into a regularized logistic regression learning algorithm to calculate a parameter matrix (ϑ , size of 115,776-by-2) to predict category for images in cross validation set.

In cross validation, we used 463 images (201 ‘paved’ and 262 ‘missing’), the parameter ϑ calculated based on previous 1,392 training samples, and the regularization parameter λ as 1.5, yielded a prediction rate of 62%. It means that 62% of the labeled cross validations images can be correctly predicted. We think this prediction rate could be largely improved by several future steps as shown in the next section.

5.3 Training Results and Next Steps

It was mentioned that the overall prediction rate for the 463 cross validation (201 ‘paved’ and 262 ‘missing’) images was 62%. A further analysis was performed to find which images were classified correctly and which ones were predicted incorrectly. Specifically, based on the two categories (‘paved’, ‘missing’) that we used, we evaluated the following factors:

1. Paved Positive: ‘paved’ images which are classified correctly, 105 images
2. Paved Negative: ‘paved’ images which are classified incorrectly, 96 images
3. Missing Positive: ‘missing’ images which are classified correctly, 184 images
4. Missing Negative: ‘missing’ images which are classified incorrectly, 78 images.

Therefore, the prediction rate for ‘paved’ category is 52.2%, while the prediction rate for ‘missing’ category is 70.2%. The training experiments can be summarized in Table 2.

Table 2. Summary of the Image Training Experiments #1

| Number of Images | Category 1 ('paved') | Category 2 ('missing') | Total |
|-----------------------|----------------------|------------------------|-------|
| Training Sample | 603 | 789 | 1392 |
| Cross Validation (CV) | 201 | 262 | 463 |
| CV positive | 105 | 184 | 289 |
| CV negative | 96 | 78 | 174 |
| CV Prediction Rate | 52.2% | 70.2% | 62.4% |

The prediction difference between the two categories could be a result of the difference between training sample size. To further look into the difference, we performed another training experiment #2. In experiment #1, we used more images in category 2 than category 1. In experiment #2, we kept the same category 1 ('paved') training sample and cross validation set. However, we reduced the number of images of training sample and cross validations in category 2 ('missing') to the same number as of in category 1. Then we tested the algorithm with the same lambda (1.5) and other parameters, and the results were summarized in Table 3.

Table 3. Summary of the Image Training Experiments #1

| Number of Images | Category 1 ('paved') | Category 2 ('missing') | Total |
|-----------------------|----------------------|------------------------|-------|
| Training Sample | 603 | 603 | 1206 |
| Cross Validation (CV) | 201 | 201 | 403 |
| CV positive | 120 | 117 | 237 |
| CV negative | 81 | 84 | 165 |
| CV Prediction Rate | 59.7% | 56.5% | 59.0% |

Based on the comparison between experiment one and two, we came to several observations. Firstly, the overall prediction rate increased with the increasing number of training images. Secondly, the classifier predicted less than 60% of the cross validation samples when there were only 600 training images. This prediction rate should be improved in order to generate meaningful results. Thirdly, the prediction rate for category 2 reached 70.2% percent when there were 789 training samples. We believe that labeling more images will further increase the prediction rate.

Last but not least, we applied all the labeled images in the classifier and went through all the images associated with each sidewalk link. The updated yet still preliminary sidewalk network can be found at <http://arcg.is/rmTyX>. We think the map accuracy could be largely improved by the following future steps:

1. Improve the preprocessing steps in Section 3.1 and the Pnet initialization process. For example, arterials with two road edges should be merged into one link in order to capture the correct centerline location. The road width could be interpolated based on city-wide roadway classification. In addition, the curvature of the sidewalk sections at intersections should be taken into consideration, and the sidewalk length should be further reduced to increase the accuracy.
2. Reduce the image size and focus on increasing the number of labeled images. In the case study, the images were captured at a scale of 1:300 and it resulted in a large image with 115,776 pixels. As shown in our experiment, the number of labeled images are more important than the size of the image. We think the following measures are worth trying in order to increase the prediction rate: 1) try a scale of 1:400, 2) cut the section

length from 50 meters to 30 meters, and 3) increase the number of labeled images to 2,000.

3. Select only clear images for labeling to start training. The analysis of the negative-predicated images in 'paved' category showed that those images were associated with noisy surroundings such as trees, tree shadows, houses, and parked vehicles.
4. Try to categorize the captured images into more detailed categories, rather than only two categories, and label them accordingly.
5. Explore more advanced machine learning algorithms such as a convolutional neural network for classifying images. Perform more sensitivity analyses and evaluate the impacts from different machine-learning parameters.

5.4 Future Work

In addition to the future steps to improve the accuracy of this mapping method, there are several future directions of this project.

1. As mentioned in the last section, it is critical to explore more advanced algorithms to improve the geographic accuracy of initialization and the accuracy of image classification.
2. With an accurate sidewalk network, a future application could be modeling the real-world pedestrian volume and the interactions between sidewalk users and the traffic.
3. In Section 3.2, we mentioned that the lack of roadway width data is a limitation for pedestrian network initialization. We think the image sweeping and categorization method developed in this research is a potential tool to extract detailed information from vehicle roadway surface.
4. Similarly, this process can be applied to categorize bicycle infrastructure, for example, Class I, II, III, or IV facility.

References

- i* Pucher, J., and L. Dijkstra. Promoting safe walking and cycling to improve public health: lessons from the Netherlands and Germany. *American journal of public health*, Vol. 93, No. 9, 2003, pp. 1509-1516.
- ii* Ehrenfeucht, R., and A. Loukaitou-Sideris. Planning urban sidewalks: Infrastructure, daily life and destinations. *Journal of Urban Design*, Vol. 15, No. 4, 2010, pp. 459-471.
- iii* Bao, J., T. He, S. Ruan, Y. Li, and Y. Zheng. "Planning Bike Lanes based on Sharing-Bikes' Trajectories", In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017. pp. 1377-1386.
- iv* Champaign Urbana Urbanized Area Transportation Study. *Sidewalk Network Inventory and Assessment – for the Champaign Urban Urbanized Area*. February 2016. <https://ccrpc.org/wp-content/uploads/2016/02/SidewalkNetworkInventoryAssessment.pdf>
- v* Champaign Urbana Urbanized Area Transportation Study. *Sidewalk explorer*. <https://sidewalks.cuuats.org/>
- vi* Frackelton, A., A. Grossman, E. Palinginis, F. Castrillon, V. Elango, and R. Guensler. Measuring walkability: Development of an automated sidewalk quality assessment tool. *Suburban Sustainability*, Vol. 1, No. 1, 2013, p. 4.
- vii* Ai, C., and Y. Tsai. Automated Sidewalk Assessment Method for Americans with Disabilities Act Compliance Using Three-Dimensional Mobile Lidar. *Transportation Research Record: Journal of the Transportation Research Board*, No. 2542, 2016, pp. 25-32.
- viii* Guensler, R., A. Grossman, A. Frackelton, V. Elango, Y. Xu, C. Toth, A. Akanser, F. Castrillon, E. Palinginis, and R. Sadana. Automated Sidewalk Quality and Safety Assessment System (Regional University Transportation Center Subproject). 2015. <https://ntl.bts.gov/lib/56000/56800/56849/12-16.pdf>.
- ix* Krizhevsky, A., I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. pp. 1097-1105.
- x* Raghavan, H., O. Madani, and R. Jones. Active learning with feedback on features and instances. *Journal of Machine Learning Research*, Vol. 7, No. Aug, 2006, pp. 1655-1686.
- xi* Fails, J. A., and D. R. Olsen Jr. Interactive machine learning. In *Proceedings of the 8th international conference on Intelligent user interfaces*, ACM, 2003. pp. 39-45.
- xii* California Department of Transportation. *Chapter 100: Basic Design Policies, Highway Design Manual*. 2015. <http://www.dot.ca.gov/design/manuals/hdm/chp0100.pdf>.
- xiii* Environmental Systems Research Institute. *Shapefile Technical Description*. <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

-
- xiv Environmental Systems Research Institute. *Choose basemap*.
<http://doc.arcgis.com/en/arcgis-online/create-maps/choose-basemap.htm>.
- xv Environmental Systems Research Institute. *Project*.
<http://desktop.arcgis.com/en/arcmap/10.3/tools/coverage-toolbox/project.htm>.
- xvi ESRI ArcGIS for Desktop. *Universal Transverse Mercator Coordinate System*.
<http://desktop.arcgis.com/en/arcmap/10.3/guide-books/map-projections/universal-transverse-mercator.htm>.
- xvii ESRI ArcGIS for Desktop. *Split line at vertices*.
<http://desktop.arcgis.com/en/arcmap/10.3/tools/data-management-toolbox/split-line-at-vertices.htm>.
- xviii ESRI ArcGIS Resource Center. *Generalize*.
<http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/001v00000006000000.htm>.
- xix ESRI ArcGIS for Desktop. *Network Analysis*.
<http://desktop.arcgis.com/en/arcmap/10.3/tools/network-analyst-toolbox/an-overview-of-the-network-analyst-toolbox.htm>.
- xx ESRI ArcGIS for Desktop. *Network Analysis Tutorial*.
<http://help.arcgis.com/en/arcgisdesktop/10.0/pdf/network-analyst-tutorial.pdf>.
- xxi ESRI ArcGIS for Desktop. *Add XY Coordinates*.
<http://desktop.arcgis.com/en/arcmap/10.3/tools/data-management-toolbox/add-xy-coordinates.htm>.
- xxii ESRI ArcGIS for Desktop. *Spatial Join*.
<http://desktop.arcgis.com/en/arcmap/10.3/tools/analysis-toolbox/spatial-join.htm>.
- xxiii NAVTEQ. *NAVSTREETS Street Data Reference Manual v4.4*. 2012.
- xxiv ESRI ArcGIS Pro. ArcPy Reference. <http://pro.arcgis.com/en/pro-app/arcpy/main/arcgis-pro-arcpy-reference.htm>.
- xxv Coates, A., B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In Document Analysis and Recognition (ICDAR), 2011 International Conference on, IEEE, 2011. pp. 440-445.
- xxvi Coursera. *Machine Learning by Stanford Online*. <https://www.coursera.org/learn/machine-learning/home/welcome>.