TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
2019, Vol. 2673(8) 499–507

DOI: 10.1177/0361198119842820
journals.sagepub.com/home/trr
SAGE

Developing an Aerial-Image-Based Approach for Creating Digital Sidewalk Inventories

Ji Luo[1], Guoyuan Wu[1], Zhensong Wei[1], Kanok Boriboonsomsin[1], and Matthew Barth[1]

Abstract

To support active mobility, extensive work has been focused on planning, maintaining, and enhancing infrastructure, such as sidewalks. A significant amount of these efforts has to go on the setup and maintenance of sidewalk inventory on a certain geographic scale (e.g., citywide, statewide). To address the stated problem, this paper proposes the development of an aerial-image-based approach that can 1) extract the features of sidewalks based on digital vehicle road network; 2) overlay the initial sidewalk features with aerial imagery and extract aerial images around the sidewalk area; 3) apply a machine learning algorithm to classify sidewalk images into two major categories, that is, concrete surface present or sidewalks missing; and 4) construct a connected sidewalk network in a time-efficient and cost-effective manner. A deep convolutional neural network is applied to classify the extracted sidewalk images. The learning algorithm gives 97.22% total predication rate for the test set and 92.6% total predication rate in the blind test. The proposed method takes full advantage of available data sources and builds on top of the existing roadway network to digitize sidewalks.

Ever-increasing transportation activities have raised a range of public concerns such as increasing traffic congestion and degraded air quality. As one of the promising remedies, active mobility such as walking and bicycling are advocated to provide various environmental, public health, and economic benefits (*1*, *2*). To promote the active transportation mode, extensive work has been focused on planning and developing pedestrian- and bicyclist-related programs which require infrastructure such as sidewalks and associated information as the cornerstones. Significant efforts were made for the setup, maintenance, and evaluation of the sidewalk inventory on a relatively large geographic scale (e.g., citywide, statewide). Based on a previous study (*3*), this paper applied a more advanced method to create such a sidewalk inventory. The sidewalk information potentially 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, an accurate (connected) sidewalk network becomes necessary.
- Crowdsourcing-based sidewalk inventory maintenance and update: as aforementioned, a large-scale sidewalk inventory will facilitate the maintenance or improvement of existing sidewalks and planning of new sidewalk construction. For example, based on the updated digitized sidewalk network database, active travelers and traffic engineers can identify or report damaged sidewalks and share the locations in a timely and cost-effective manner.

Conventionally, traffic engineers and researchers have to rely on field measurements to conduct sidewalk survey and assessment (*4*, *5*), which is rather resource

[1]College of Engineering, Center for Environmental Research & Technology, University of California, Riverside, Riverside, CA

Corresponding Author:
Address correspondence to Ji Luo: ji.luo@ucr.edu

Figure 1. A comparative example on pedestrian navigation using (*a*) existing roadway network and (*b*) potential sidewalk network (from National Center for Sustainable Transportation [*3*]).

consuming, regarding both time and cost. Recently, a few studies attempted to collect the pedestrian volume and bicycle lane inventory with crowd-sourced information (*6*, *7*), or synthesize digital sidewalk inventory based on commonly available open data (*8*). However, most of the existing methods for sidewalk system digitization are neither comprehensive nor cost-effective at city-wide scales.

On the other hand, thanks to the rapid advances in computational capability and explosion of data availability, machine learning techniques have shown great potential for image recognition and classification (*9*). To address the above issues, we propose an aerial-image-based sidewalk digitization method which is more time-efficient and cost-effective than existing methods. The basic idea is to take full advantage of existing vehicle roadway networks to reconstruct an initial (connected) sidewalk network. Then, the 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.

The paper is organized as follows: First is an overview of the proposed methodology for sidewalk digitization. Then the paper will illustrate how to construct a connected sidewalk network (as an initial) by mapping from an existing roadway network in details. A convolutional neural network (CNN) based image processing algorithm will be applied to determine if there is a sidewalk present or not in a satellite image. Next, it presents the application of the method to the surrounding communities of University of California Riverside, and evaluates the performance of the method. The last section concludes this paper with discussion on potential future work.

Methodology Overview

By a general definition, sidewalks are to accommodate pedestrians at a level of service equal to that of vehicles using the roadway (*10*). 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 not straightforward 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 does not exist) categories as a first attempt. In this paper, unless otherwise noted, the phrases "paved" and "concrete-surfaced" are interchangeable, and "missing" means that the sidewalk surfaces do not exist.

The overall method is illustrated in Figure 2. In this study, we proposed to map the features of sidewalks based on the roadway network as the first step. The roadway network data applied in this study, such as roadway shapefiles, should include road link attributes and position coordinates (*11*). Secondly, a Python script was written to sweep each sidewalk link in the initialized sidewalk network and extract the aerial image within that area. In parallel, we manually classified a large number of aerial images (e.g., paved or missing sidewalk) of sidewalk network and set up a machine learning algorithm to learn from the labeled images. We trained the machine learning classifier to be able to achieve a reasonable prediction rate (a comparison is presented in the Case Study section). Then the classifier could be used to predict the surface attributes of the extracted image using the trained machine learning algorithm. Here, we name the vehicle roadway network as "Vnet," and pedestrian sidewalk network as "Pnet" for convenient reference.

Mapping Vnet to Pnet

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

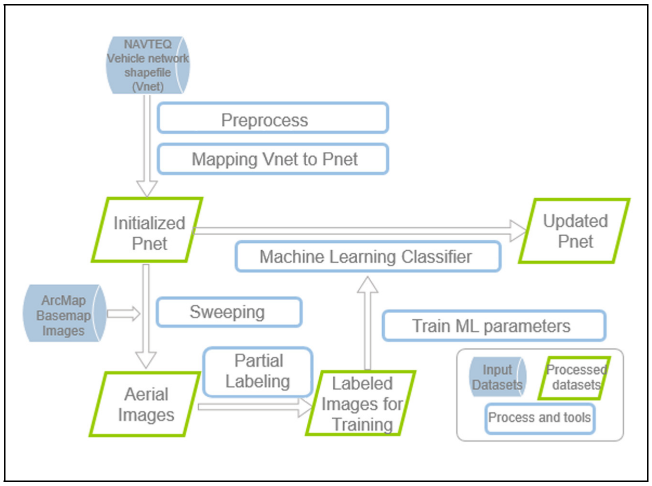

Figure 2. Flow chart of overall methodology, adopted from National Center for Sustainable Transportation (*3*).

network. Geographic processing software was applied, and all features were projected to universal transverse mercator (UTM) coordinate system (*12*, *13*). The preprocessing includes the following steps:

(1) To remove freeways and ramps from the Vnet, as we assume that sidewalks do not exist by the side of freeways and freeway ramps.
(2) To prevent duplicating sidewalk links, we remove one edge and keep only the other consistent edge along the road, if one section of arterial is presented with two edges. For example, for an arterial that has an island in the middle and the vehicle traffic of two directions are represented with two parallel links, only one consistent edge will be kept. This step will introduce errors in the sweeping step (see Image Sweeping section), and potential improvement will be discussed in future works.
(3) To simplify link geometry, the filtered Vnet links are generalized (*14*, *15*). This step not only preserves the geometry of the network but also reduces the number of links by up to 20%.
(4) To build the Vnet graph based on the preprocessed Vnet shapefile. An example link table and node table are shown in Table 1.
(5) To 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 streets in California, therefore in this study, the road width was estimated based on the road names provided by NAVTEQ Streets (*16*). 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, or TER were given a 10-m road width, and roads with BLVD were given an 18-m width. A number of exceptions were made based on the knowledge of the local streets and survey of Google Maps.

Table 1. Example Network Graphs (from National Center for Sustainable Transportation [*3*])

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

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 3*a*, our goal for the mapping from Vnet to Pnet 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 ($Link_i$) and the roadway link's half-width (d, roadway centerline to the edge of sidewalk) which was estimated as described in Equation 5 below.

$X1, Y1, X2, Y2$ are Cartesian coordinates (UTM) of the link nodes. To calculate the correct merged sidewalk node that is shown as a blue/orange diamond in Figure 3*b*, one condition is that we should guarantee that point $P1$ and $P2$ always fall into a predictable side (e.g., upper or lower side, left or right side) of their original Vnet link. In this case, we specify that a Pnet link's $P1$

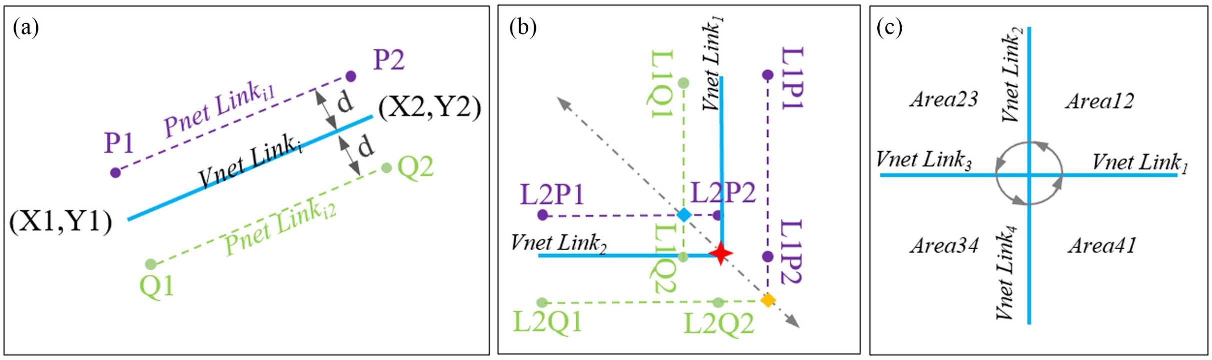

Figure 3. Schematics of mapping the preliminary sidewalk nodes (from National Center for Sustainable Transportation [3]).

and *P2* must fall within the upper or right-hand side of its original Vnet link, as shown in Figure 3*a* and *b*. 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 formulas:

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

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

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

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

where 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 3*b*, when two Vnet links intersect at a Vnet node (marked by a red star), the newly generated Pnet links (by following previous rules) as 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 intersection 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 3*b*), we rank all the connected Vnet links by the relative angle to X-axis. Referring to Figure 3*c*, 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 Equations 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 as shown in
Figure 3c
    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
```

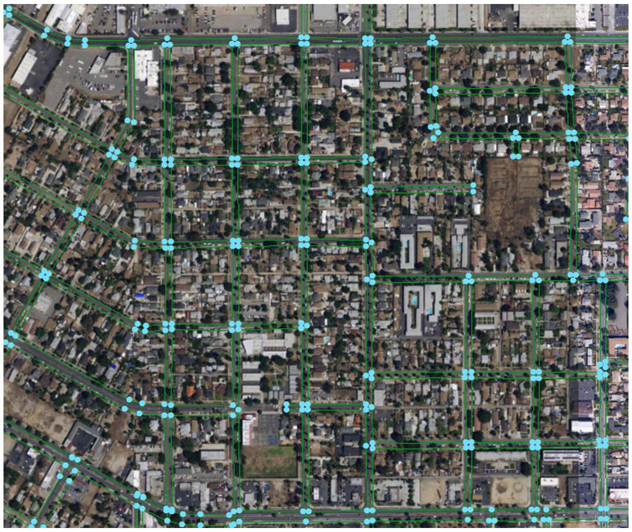

Figure 4. An example of mapped sidewalk based on vehicle roadway network (from National Center for Sustainable Transportation [*3*]).

If we assume there is a total of J links (*link1, link2*, …, *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 ($n \geqslant 2$). 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 4, where the preliminary sidewalks are shown in green line, and sidewalk nodes are shown in blue dots.

Development of Image Processing Algorithm Based on Machine Learning

With the initialized sidewalk network, our next goal is to identify whether a certain segment is concrete 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 Figures 5 and 6. To enable a large number of image collection, we developed an image sweeping method to capture the aerial images along the Pnet. After collecting and labeling a considerable number of images, a CNN, which is one of the machine learning methods, is applied to train the learning algorithm to identify the "sidewalk surface" feature.

Image Sweeping

To collect images that can be trained by a machine learning algorithm, we considered capturing a uniform size of images which contain desired objects to be recognized by the algorithm. As shown in Figure 4, collecting images along the Pnet links involves a large number of screenshot operations and it would be challenging to manually perform the task.

With the aid of ArcPy package (*17*), we developed a Python script and executed it in ArcMap-Python Environment. The Python script was able to command the map interface to zoom into each Pnet link at a designated map scale (1:300 in this study) and take screenshots at ESRI World Imagery Basemap with designated image size (e.g., 200 × 400 pixel). The pseudocode is provided below. For alternative imagery sources (e.g., drones), we recommend a minimum resolution of 1:500 scale and should be orthorectified and aligned with the digital road network.

```
function 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 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 a rather 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 min.

Image Labeling

After collecting a large number of aerial images along the initial sidewalk network (i.e., Pnet), we labeled a subset of them for training the proposed machine learning algorithm. 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 5, a label of "paved present" is assigned to a group of images which clearly present concrete-surfaced sidewalk segments. On

Figure 5. Screenshots of "paved present" sidewalk sections from aerial image at 1 to 300 scale (from National Center for Sustainable Transportation [*3*]).

Figure 6. Screenshots of "missing" sidewalk sections from aerial image at 1 to 300 scale (from National Center for Sustainable Transportation [*3*]).

the other hand, in Figure 6, the label of "missing sidewalk" is assigned to a group of images which can be identified as a lack of paved sidewalk segments.

Generally, the more labeled images there are, the better the training results will be. Usually, image classification requires a large number of training samples. The labeled images will be critical inputs for the next training step.

Training Machine Learning Parameters

The model used for training the parameters is a CNN (*18*, *19*), with architecture shown in the figure below. It is assumed that for each layer the batch size equals 1. The CNN structure was chosen because CNN is known for outstanding performance in the realm of image classification.

An image is usually represented with a vector of values, which corresponds to the value of color (e.g., red, green, and blue [RGB]) of each pixel in the image. For example, if one image is of size 200 $\times$ 400 pixels, the image will be represented with a 200 $\times$ 400 $\times$ 3 matrix. At the same time, this matrix will be associated with a label, in this case, "1" for sidewalk present or "2" for missing sidewalk. As shown in Figure 7, the input to the network is the labeled 3-channel RGB image, and the output is a 2-entry vector indicating the probability of whether the image belongs to case 1, with [1,0] as label, or case 2, with [0,1] as label. The network loss is defined as a cross-entropy loss:

$$\boldsymbol{L} = -\sum_i \boldsymbol{y_i log x_i} \tag{6}$$

With x_i and y_i as the entry of label and network output, respectively. The network consists of multiple convolutional layers (conv), with a kernel size of 3 $\times$ 3 and

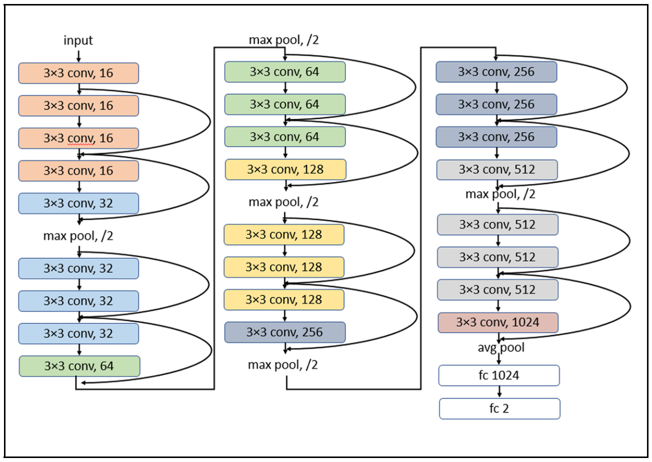

Figure 7. Flowchart of CNN applied in the learning algorithm (adopted from He et al. (*19*) with descriptions below).

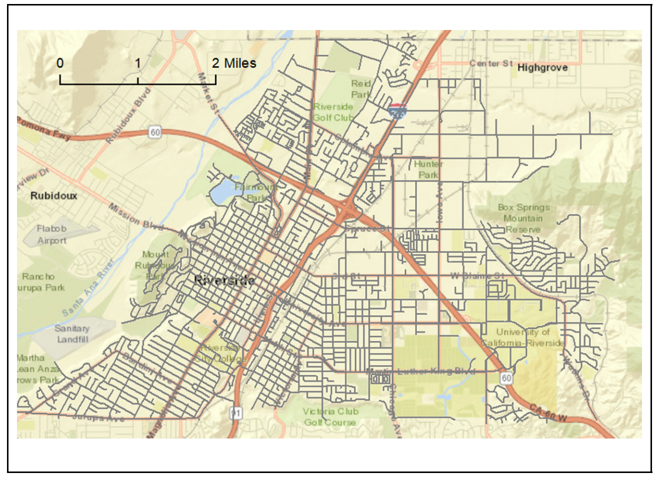

Figure 8. Example roadway network in the case study (from National Center for Sustainable Transportation [*3*]).

stride equals 1, followed by rectified linear unit (ReLU) as nonlinear activation function. Residual blocks which add padded input of one layer to the output of next layer are used to increase the training speed of the network (*19*). The learnable parameters include filters, weights, and biases in the convolutional layers and fully connected layers (fc) are updated using an adaptive moment estimation (Adam) (*20*) optimizer with a learning rate of 1×10^{-4}.

Case Study and Performance Evaluation

Preprocessing and Initial Mapping

For a case study, we selected an area surrounding University of California, Riverside shown in Figure 8. There are 4,385 roadway links in the Vnet. The preprocessing and initial mapping were performed by following the methods described earlier. The initial mapping created 14,806 sidewalk links, including 386 cul-de-sac, 8,388 sidewalk segments, and 6,032 crosswalks. The initialized sidewalk map can be viewed at https://arcg.is/184PbK.

Sweeping and Labeling Images

In this project, image sweeping and labeling are the most time-consuming processes. There are 8,388 sidewalk segments and their aerial images need to be captured (crosswalks are not swept because crosswalks are usually a part of vehicle roadways). Sweeping 8,388 images took approximately 12 h.

We marked 792 images with “paved” label and 1040 images with “missing” label. Each image captured an area 50 × 16 m and was of size 603 × 192 pixels. Among the 1,832 labeled images, 1,472 images (632 paved and 840 missing) were used as training dataset, 180 images (80 paved and 100 missing) for validation, and 180 images (80 paved and 100 missing) for testing. All images were chosen randomly from the labeled images.

Training Results

Following the training of a deep CNN, we blind-tested its prediction performance by feeding it with the labeled images in a new test set. The training results are shown in Table 2. The results demonstrated that the learning algorithm is able to identify the two categories of sidewalk using the presented framework at a total prediction rate of 97.2%. Table 2 also listed the prediction performance of a simple logistic regression (LR) method applied in the previous related research (*3*). It is evident that CNN algorithm has significantly higher prediction rate than the LR algorithm.

Furthermore, with the learning algorithm, we classified all the images that were collected from the sweeping method. Then 1,041 new images (aside from the training, validation, and test sets) were labeled manually to validate the predication results of the learning algorithm. We found that 964 images were correctly classified. The results indicated that our learning algorithm can achieve a satisfactory prediction rate of 92.6% in the blind test.

The CNN is implemented in TensorFlow (*21*). In relation to computation performance, the training process took approximately 20 min with 8,000 iterations, corresponding to about 54 epochs. The convergence plot is shown in Figure 9, indicating that convergence was achieved with approximately 8,000 iterations. The training, validation, and testing of the network were performed on a PC with four-core 3.60 GHz CPU, 16GB of RAM, and Nvidia GeForce GTX 1080Ti GPU.

Table 2. Training Result

Number of images	Case 1 ("present")	Case 2 ("missing")	Total
Training sample	632	840	1472
Validation sample	80	100	180
Testing sample	80	100	180
Testing positive: CNN	77	98	175
Testing negative: CNN	3	2	5
Testing prediction rate: CNN	96.3%	98%	97.2%
Testing positive: LR	41	66	107
Testing negative: LR	39	34	73
Testing prediction rate: LR	51.3%	66%	59.4%

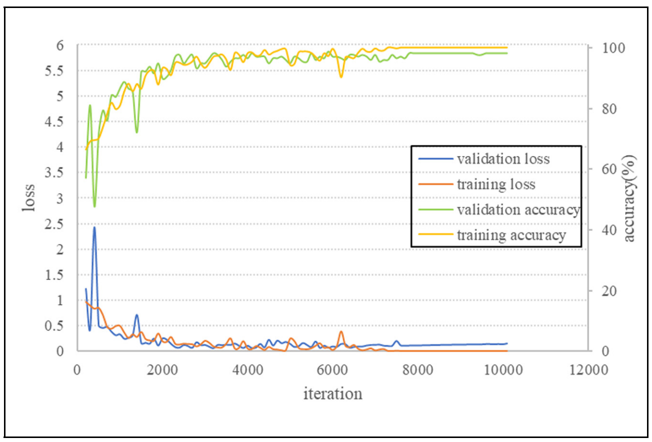

Figure 9. Convergence plot of the training process.

Discussion and Future Directions

In this study, we proposed a method that takes full advantage of available data sources and builds on top of the existing roadway network to create a digital sidewalk inventory. The learning algorithm is able to achieve a satisfactory prediction rate of 92.6% in the blind test, especially, it can successfully identify areas with poor sidewalk conditions as shown in Figure 6. The mapping approach could potentially lay the foundation for sidewalk inventory and improved active traveler applications. What is more important is that there is potential for future improvement:

1. Improve the preprocessing methods and the Pnet initialization process. For example, arterials with two road edges could be merged into one link to capture the correct centerline location. The road width could be interpolated based on city-wide roadway classification or image processing techniques such as edge detection.
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. However, the number of labeled images is more important than the size of the image. We think the following measures are worth trying to further increase the prediction rate: 1) try a map scale of 1:400, 2) reduce the section length from 50 m to 30 m, and 3) increase the number of labeled images to 2,000.
3. Try to classify the captured images into more detailed categories, rather than only two categories. For example, green space, curb cut, parking lot/driveway. Also, there are many obstructed images (e.g., sidewalk shaded by trees, shadows, or unable to identify objects) in the collected image set and those images should be identified as well.

Acknowledgments

We would like to acknowledge our funding agencies. The work was sponsored by U.S. Department of Transportation Grant No. DTRT13-G-UTC29, and California Department of Transportation Contract No. 65A0527. We thank Mr. Alexander Vu from the Center for Environmental Research and Technology for his technical support. We greatly appreciate Mr. Nathan Mustafa from City of Riverside and Mr. Dustin Foster from Caltrans for their valuable inputs. Last but not least, we would like to thank our project manager Mr. Patrick Tyner and Ms. Lauren Iacobucci for their support throughout the project.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: JL and GW; data collection: JL and GW; analysis and interpretation of results: JL, GW, and ZW; draft manuscript preparation: JL, GW, ZW, KB, and MB. All authors reviewed the results and approved the final version of the manuscript.

References

1. 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.
2. Hanson, S., and A. Jones. Is there Evidence that Walking Groups Have Health Benefits? A Systematic Review and Meta-Analysis. *British Journal of Sports Medicine*, Vol. 49, No. 11, 2015, pp. 710–715.
3. National Center for Sustainable Transportation. *Developing an Interactive Machine-learning-Based Approach for Sidewalk Digitalization*. 2018. https://ncst.ucdavis.edu/project/developing-an-interactive-machine-learning-based-approach-for-sidewalk-digitalization/. Accessed March 1, 2019.
4. Champaign Urbana Urbanized Area Transportation Study. *Sidewalk Network Inventory and Assessment – for the Champaign Urban Urbanized Area*. Illinois Department of Transportation, Ill., 2016.
5. 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.
6. CitiLab. *Citilabs' Streetlytics Provides Complete Hour-by-Hour Pedestrian Volume Analytics for the Entire United States*. http://www.citilabs.com/citilabs_blog/citilabs-leverages-location-data-create-first-ever-nationwide-analytics-platform-pedestrian-movements/?utm_campaign=Company%20News&utm_source=hs_email&utm_medium=email&utm_content=62537380&_hsenc=p2ANqtz-8sefRcesPCVz4C1UlOAjjwvbJUFiQdqyPx9SQk9BFgR0uUgGQ0inYlZIQRauzFywegd6X_DJZzDfslfkmzSaKNwFxvkQ&_hsmi=62537380. Accessed March 1, 2019.
7. Bao, J., T. He, S. Ruan, Y. Li, and Y. Zheng. Planning Bike Lanes Based on Sharing-Bikes' Trajectories. *Proc., 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2017, pp. 1377–1386.
8. Bolten, N., A. Amini, Y. Hao, V. Ravichandran, A. Stephens, and A. Caspi. Urban Sidewalks: Visualization and Routing for Individuals with Limited Mobility. *Proc., 1st International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, ACM, 2015, pp. 122–125.
9. He, K., X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification. *Proc., IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.
10. California Department of Transportation. *Chapter 100: Basic Design Policies, Highway Design Manual*. 2015. http://www.dot.ca.gov/design/manuals/hdm/chp0100.pdf. Accessed March 1, 2019.
11. Environmental Systems Research Institute. *Shapefile Technical Description*. http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf. Accessed March 1, 2019.
12. Environmental Systems Research Institute. *Project*. http://desktop.arcgis.com/en/arcmap/10.3/tools/coverage-toolbox/project.htm. Accessed March 1, 2019.
13. 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. Accessed March 1, 2019.
14. ESRI ArcGIS for Desktop. *Generalize*. http://desktop.arcgis.com/en/arcmap/10.3/tools/editing-toolbox/generalize.htm. Accessed March 1, 2019.
15. ESRI ArcGIS for Desktop. *Methods for Splitting Line Features*. http://desktop.arcgis.com/en/arcmap/10.3/manage-data/editing-existing-features/ways-to-split-a-line-feature.htm#ESRI_SECTION1_84606E86D9DE4FA5A95AEBC095657767. Accessed March 1, 2019.
16. NAVTEQ. *NAVSTREETS Street Data Reference Manual v4.4*. 2012.
17. ESRI ArcGIS Pro. *ArcPy Reference*. http://pro.arcgis.com/en/pro-app/arcpy/main/arcgis-pro-arcpy-reference.htm. Accessed March 1, 2019.
18. LeCun, Y., Y. Bengio, and G. Hinton. Deep Learning. *Nature*, Vol. 521, No. 7553, 2015, p. 436.
19. He, K., X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *Proc., IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
20. Kingma, D. P., and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.
21. Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard. Tensorflow: A System for Large-Scale Machine Learning. *Symposium on Operating Systems Design and Implementation*, No. 16, 2016. pp. 265–283.

The Standing Committee on Information Systems and Technology (ABJ50) peer-reviewed this paper (19-02287).