Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery

Galen Weld[1], **Esther Jang**[1], **Anthony Li**[2], **Aileen Zeng**[1], **Kurtis Heimerl**[1], **and Jon E. Froehlich**[1]
[1]Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, USA
[2]Department of Computer Science, University of Maryland, College Park, USA
{gweld, infrared, aileenz, kheimerl, jonf}@cs.washington.edu, antli@umd.edu

ABSTRACT
Recent work has applied machine learning methods to automatically find and/or assess pedestrian infrastructure in online map imagery (*e.g.,* satellite photos, streetscape panoramas). While promising, these methods have been limited by two interrelated issues: small training sets and the choice of machine learning model. In this paper, aided by the recently released Project Sidewalk dataset of 300,000+ image-based sidewalk accessibility labels, we present the first examination of deep learning to automatically assess sidewalks in Google Street View (GSV) panoramas. Specifically, we investigate two application areas: automatically *validating* crowdsourced labels and automatically *labeling* sidewalk accessibility issues. For both tasks, we introduce and use a residual neural network (ResNet) modified to support both image and non-image (contextual) features (*e.g.,* geography). We present an analysis of performance, the effect of our non-image features and training set size, and cross-city generalizability. Our results significantly improve on prior automated methods and, in some cases, meet or exceed human labeling performance.

Author Keywords
Neural networks, accessibility, sidewalks, computer vision

ACM Classification Keywords
I.2.10. Artificial Intelligence: Vision and Scene Understanding; I.2.6. Artificial Intelligence: Learning

INTRODUCTION
Sidewalks should benefit all of us. They provide a safe, environmentally-friendly conduit for moving about a city. For people with disabilities, sidewalks can have a significant impact on independence [47], quality of life [38], and overall physical activity [17]. While mapping tools like Google and Apple Maps have begun offering pedestrian-focused features, they do not incorporate sidewalk routes or information on sidewalk accessibility [23], which limits their utility and disproportionately affects people with disabilities. A key challenge is data: Where does it come from? How is it collected?

Traditionally, sidewalk audits—which gather data on the presence and quality of sidewalks—are performed via in-person inspections by city transit departments or community volunteers. However, these audits are expensive, labor intensive, and infrequent. [1] Moreover, the resulting data is in disparate formats, is not typically open (*i.e.,* published online), and is not intended for end-user tools [23, 50]. To expand who can collect sidewalk data and to improve data granularity and freshness, researchers have introduced smartphone-based tools [15, 46, 52] as well as instrumented wheelchairs [35, 39, 51, 57], both of which capture sidewalk information *in situ* as it's experienced. However, these tools have been limited by low adoption, small geographic coverage, and high user burden (*e.g.,* requiring users to take out their phones, load an app, take a picture, annotate it, and upload it) [20, 23].

To partially address these scalability issues, researchers have begun developing automated methods for sidewalk assessment using machine learning and online imagery (*e.g.,* satellite photos [10, 8], panoramic streetscape imagery [31, 32, 59]). While still early, these complementary approaches promise to dramatically decrease manual labor and cost. However, they have been limited by two interrelated issues: small training sets and the choice in machine learning model—both of which negatively impact performance. In this paper, we attempt to address both of these issues.

We present the first examination of deep learning methods to automatically assess sidewalk accessibility in terms of *curb ramps*, *missing curb ramps*, *surface problems*, and *sidewalk obstructions* from widely available streetscape imagery. Our work is enabled by the recently released Project Sidewalk open dataset, which contains a corpus of 300,000+ image-based sidewalk accessibility labels collected via remote crowdsourcing in Google Street View (GSV) [55] (Figure 1). Specifically, we investigate two application tasks using GSV panoramas: automatically *validating* crowdsourced labels and automatically *labeling* sidewalk accessibility issues.

Our research questions include:

- **R1:** How well does our machine learning approach perform across our two tasks (validation and labeling)?
- **R2:** What is the impact of additional, non-image related training features on performance?
- **R3:** How does classification accuracy change as a function of training data amount?
- **R4:** How well does our model generalize across cities?

To address these questions, we trained two sets of deep convolutional neural networks using ResNet-18 [33]—one set for

[1]As one example, the Seattle Department of Transportation completed their first ever sidewalk assessment in 2016, which took 14 interns nearly a year to complete. [1]

ASSETS'19, October 28–30, 2019, Pittsburgh, PA, USA
 ISBN 978-1-4503-6676-2/19/10…$15.00
DOI: 10.1145/3308561.3353798

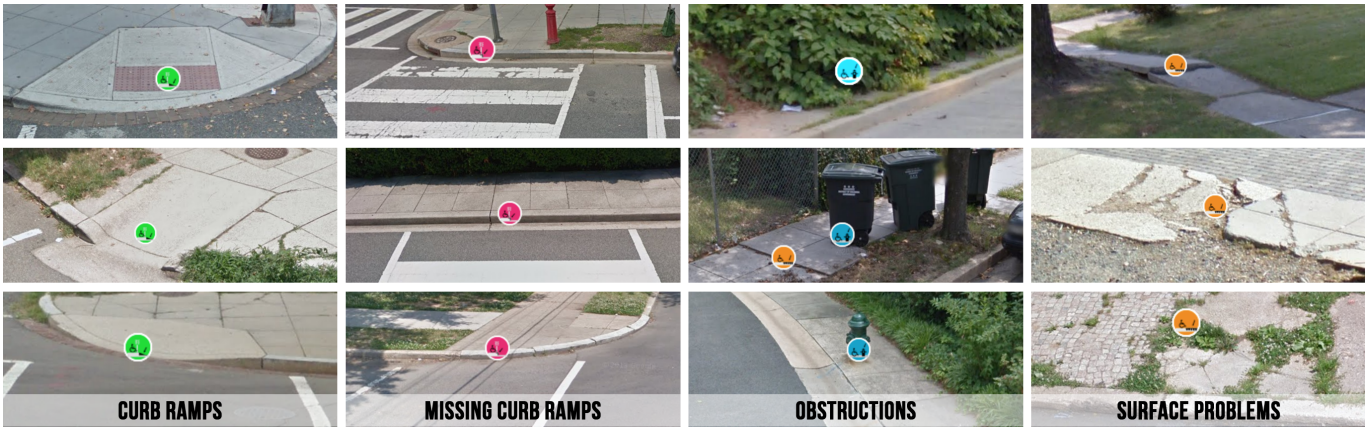


Figure 1: Examples of the four label types used to train and test our deep learning models for semi-automatic sidewalk assessment: *curb ramps*, *missing curb ramps*, *obstructions*, and *surface problems*. Figure adapted from Project Sidewalk [55].

each task. We experimented with three input feature types: *image features* cropped from a GSV panorama, *positional features* encoding the position of items within a panorama, and *geographic features* encoding the position of a panorama within a city's street network (addressing R2).

While direct comparisons with prior work are challenging, our results show labeling performance that meets or exceeds human crowdworkers [30] (R1). To complement our quantitative findings, we also qualitatively examine our results, identifying common sources of error, such as imagery limitations (shadows or poor resolution) and contextual limitations (*e.g.,* predicting missing curb ramps requires inferences about where pedestrians are intended to cross streets).

Our work contributes to a larger, overarching research agenda aimed at developing fast and accurate semi-automatic sidewalk assessment tools to help transform how city governments and citizens alike track, perceive, and use pedestrian infrastructure. We envision four specific use cases: first, to gather the data necessary to create accessibility-aware mapping tools [23, 28]; second, to help increase transparency about urban accessibility and, relatedly, to hold policy makers and governments accountable for making promised changes and meeting federal accessibility regulations (*e.g.,* ADA [3, 2]); third, to provide a fast and low-cost method to help city governments—particularly those without full-time sidewalk or ADA staff—track, triage, and fix identified issues; and finally, to provide granular geolocated sidewalk accessibility data to enable more informed decision-making for policy makers and urban planners.

In summary, the contributions of this paper include: (1) a new deep learning approach to automatically *validate* crowdsourced labels of sidewalk accessibility problems in GSV imagery; (2) a related approach to automatically *find* and *label* sidewalk problems in GSV imagery; and (3) a preliminary analysis of how these techniques generalize to other cities.

RELATED WORK

Residual Neural Networks in Computer Vision

Recent advances in deep learning and neural networks have dramatically improved computer vision performance, including for facial recognition [48], scene reconstruction [12], and even image translation [34]. Our approach uses Residual Neural Networks (ResNets) [33, 60, 61], a specific flavor of neural network with *shortcut connections* between inner network layers that allow them to avoid some sources of overfitting. ResNets achieve state-of-the-art results in many applications [60, 33]. While we do not provide any additions to the core methods present in the vision literature, we believe our contribution to be a novel application of these techniques in a particularly important domain.

Computer Vision for Urban Features

The computer vision community has studied urban scenes in a variety of contexts (*e.g.,* for self-driving vehicles [18, 24, 16], for tracking urban change [11]). To train these machine-learning systems, significant effort has been devoted to developing labeled street-level imagery (*e.g.,* the *CityScapes Dataset* [18]). However, these data (and the models trained with them) focus primarily on street features relevant to vehicles, not pedestrians. Existing literature on detecting curbs and curb ramps does so largely through the lens of parking [16, 24] using real-time data generated from sensing devices that are widely deployed on autonomous vehicles. While several commercial services gather urban data [6, 4] and apply computer vision [5], these are not targeted explicitly at identifying accessibility features, are not open source, and have not been evaluated scientifically, so it is not possible to directly compare the performance of these systems with our own. In general, using CV techniques to automatically detect sidewalk features for people with differing mobility remains underexplored.

Most related to our work is that of Sun *et al.*, which identifies missing curb ramps in GSV panoramas using a Siamese trained fully convolutional context network (SFC) that evaluates the context around regions of the image [59]. While promising, this work is limited by a small training dataset (1087 labeled intersections from *Tohme* [32]), focuses only on missing curb ramps, and achieved only marginal results (recall of less than 30%).

Geographic Information Systems

Geographic Information Systems (GIS) researchers have worked with pedestrian infrastructure, including the development of an algorithm for combining sidewalk data with street-grid data [37], computing distances of sidewalk paths using manually-labeled satellite imagery [36], and exploring correlations between census data and sidewalk data with the goal of improving community health outcomes [27]. This work, however, focuses on performing computations with existing databases of sidewalk features, not on gathering new data about sidewalks. These existing databases are either provided by local governments or laboriously created by researchers manually labeling map data. Our research extends the prior work by contributing new automated approaches to assess sidewalks using streetscape imagery.

Uses of Google Streetview in Computer Vision

Google Streetview panoramas, our primary source of imagery, have been used to semi-automatically track urban greenery [44, 42, 43], predict real-estate prices [40], and train convolutional neural networks (CNNs) to detect changes in urban areas over time [56]. Some preliminary work exists on using GSV to identify street-level infrastructure for pedestrians. For example, Ahmetovic *et al.* [9, 7] identify striped ("zebra") crosswalks with a recall of 90% and precision of 60% using a combination of GSV panoramas and satellite imagery. Our research complements and extends prior work by using GSV and deep learning methods to semi-automatically assess sidewalk features relevant to accessibility: *curb ramps*, *missing curb ramps*, *obstructions*, and *surface problems*.

Hybrid Crowdsourcing and Computer Vision

While recent developments in deep learning have improved automatic object detection performance, results still vary and are highly context-dependent. Thus, researchers often combine automated solutions (which are fast but noisy) with human work (which is slow and expensive but can perform better than machines for some problems)—so called hybrid crowdsourcing + computer vision systems. For example, Hara *et al.* developed *Tohme* [32], which combined manual labeling with computer vision for semi-automatically identifying curb ramps and missing curb ramps in streetscape imagery. Their hybrid system performed almost as well as an all-manual approach while reducing the manual labor time cost by 13% [32]. Ahmetovic *et al.* also explored the use of hybrid automated + crowd-sourced labels for crosswalk detection and found that humans were able to improve recall from 77% to 93% and improve precision from 94% to 97%, but with a significant increase in cost and time [9]. Though our current work explores a purely automated solution, future systems could incorporate our models to reduce manual labor or improve accuracy.

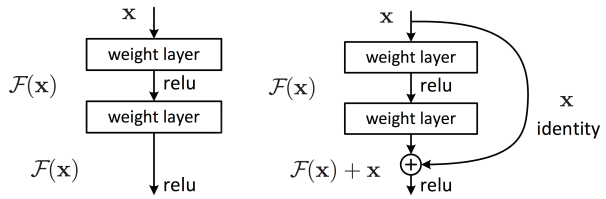
APPROACH

We explore two applications of deep learning to streetscape images: automatically *validating* human labels on pre-labeled GSV panoramas and automatically *labeling* GSV panoramas to locate and classify sidewalk accessibility problems. For both tasks, we use a deep learning approach called a *residual neural network* (ResNet), which are increasingly common in computer vision due to their relatively fast training and high performance [33]. While we construct individual models for each task, both networks are trained on the same three input features: (i) *image crops* from GSV panoramas; (ii) *positional* features describing the position of a point in a scene; and (iii) *geographic* features describing the location of a scene within the broader context of a city's geography. Below, we describe our neural network architecture, our training and test datasets, and expand on our two tasks. Our system, implemented in PyTorch, is open source and publicly available.[2]

Architecture

Residual Neural Networks

While traditional deep neural networks can degrade in performance with more layers due to overfitting, residual networks (ResNets) allow the use of deeper layers to increase accuracy. More concretely, the inner layers of ResNet-18 are arranged into two-layer "blocks" as shown in Figure 2b where a "shortcut" connection adds the identity of the input to the block output. Given the direct mapping of identity to output, the network weights only have to learn the remainder of the underlying function mapping input to output, which is referred to as the "residual." For our experiments, we extend the PyTorch reference implementations of ResNet-18 by adding layers that incorporate *positional* and *geographic* features.



(a) Plain block **(b)** ResNet block

Figure 2: Diagram of the inner layers of (a) a plain deep convolutional neural network *vs.* (b) a residual neural network. Figure adapted from [33]. ReLU (Rectified Linear Unit) is defined as $\texttt{relu}(x) = \texttt{max}(0, x)$

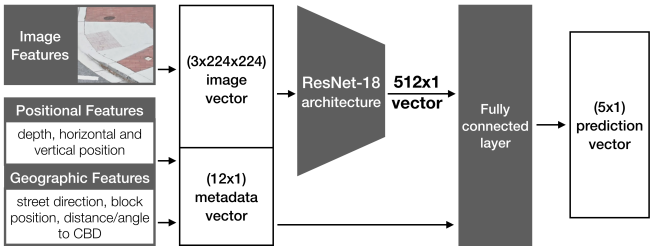


Figure 3: The structure of our modified ResNet-18 architecture, which, in addition to pixel-based imagery, incorporates extra features such as position in scene, depth information, and geographic data.

Extending Feature Inputs to ResNets

Typically, ResNet-based neural networks only take images as input. In our case, however, we wanted to leverage additional contextual knowledge such as the *position* of a crop within a streetscape panorama, the *geographic* location of the panorama within the city, or the relative location of a panorama

[2] https://github.com/ProjectSidewalk/sidewalk-cv-assets19

on a street (*i.e.,* distance from the intersection). To incorporate both image and non-image input features into the ResNet model, we perform a two-stage input process. First, we feed a 224x224x3 image vector through a series of convolutional layers derived from ResNet to obtain a 512x1 vector—essentially treating intermediate ResNet layers as a method for dimensionality reduction (top pathway in Figure 3). We then append our additional vector of contextual features (12x1) to this 512x1 vector to generate a final input vector of size 524x1. This combined vector is fed into the last layer of the neural network, which outputs a prediction vector of 5x1. Here, rows correspond to a prediction confidence for each of the five classes. To determine the final prediction, we simply compute the `argmax` of the prediction vector.

Transfer Learning Using ImageNet

Finally, to improve performance, we use a popular machine learning technique called transfer learning [49] to initialize the ResNet model with weights learned from pre-training on the ImageNet image corpus [19]. Prior work has shown how transfer learning with ImageNet can significantly boost performance by leveraging commonalities between images of real-world scenes (*e.g.,* [45]). In preliminary experiments, we found the addition of transfer learning to significantly improve our results; for example, by improving auto-validations of curb ramps by over 60%.

Input Features

We have three categories of input features intended to capture both the appearance and geographic structure of a specific point within a GSV panorama:

- **Image Features**. A 224x224 pixel RGB image cropped from a larger GSV panorama, encoding the visual appearance around the point in a scene. This image is automatically proportioned based on its distance from the GSV camera and then downsampled to 224x224x3 before being input to the neural network.
- **Positional Features**. A 7x1 vector encoding the position of the point in the scene, including: (i) the distance of the point to the GSV vehicle as calculated by LiDAR (*i.e.,* the depth position of the point in the scene); (ii) the sine and cosine of the angle between the point and the street axis (*i.e.,* the horizontal position of the point in the scene); (iii) the sine and cosine of the angle between the point and the horizon (*i.e.* the vertical position of the point in the scene).
- **Geographic Features**. A 5x1 vector, which includes: (i) the sine and cosine of the axis of the street relative to true North; (ii) distance and bearing of the panorama to the center of the city; (iii) and the absolute and relative position of the panorama within the street segment (*i.e.,* distance from the nearest intersection)

Extracting Image Crops from Point Labels

Because our human-supplied labels are *x,y* point labels on panoramas (rather than bounding boxes), we needed to derive an approach for automatically sizing the crops made around those points. The challenge is compensating for differing apparent sizes of objects at different distances (*i.e.,* optical perspective). For example, a curb ramp 1-meter wide and 5 meters from the camera will *appear* larger than an identical curb ramp 50 meters away. To incorporate optical perspective and to auto-size our crops accordingly, we use the GSV depth data, specifically: size (in pixels) $= \frac{4}{15} * \texttt{distance} + 200$. This depth-proportioned algorithm was derived empirically using the bounding box dataset in [32].

Data

We have three distinct datasets to *train*, *validate*, and *test* our task-specific ResNet models. For *training* and *validation*, we use the Project Sidewalk dataset [55], which consists of 205,385 image-based sidewalk accessibility labels (Figure 1) across 58,034 GSV panoramas from Washington, D.C. collected via remote crowdsourcing. We randomly partitioned these panoramas and their corresponding labels into training, validation, and test sets following an 80/10/10 split (Table 1). This three-way split is a common machine learning evaluation method that reduces overfitting by evaluating final performance on a previously unseen dataset (the test set) [53].

For *testing*, we manually created a ground-truth dataset by labeling a subset of the test dataset (224 of the 5774 panoramas). This subset was produced to ensure at least 100 examples of each label type. For labeling, two authors worked collaboratively to comprehensively label all visible accessibility problems in each panorama. Consensus was achieved on every placed label. Each panorama required approximately 4-5 minutes to label completely. Any accessibility problems which were completely obscured or too distant to be identified with certainty were left unlabeled. The size of the final ground truth dataset is shown in Table 1.

	Project Sidewalk		Ground Truth	
	Train	**Val**	**Test**	**Total**
Number of Panos	46,463	5,797	224	52,484
Missing Curb Ramp	15,692	1,872	135	17,699
Surface Problem	7,003	897	224	8,124
Obstruction	17,519	2,305	142	19,966
Curb Ramp	119,799	14,731	662	135,192
All Labels	**160,013**	**19,805**	**1,163**	**180,981**

Table 1: Training, Validation, and Test (Ground Truth) set sizes, which contain panoramas and image-based sidewalk accessibility labels. The training and validation sets, used for developing models, were created with crowdsourced labels from Project Sidewalk. The test set, used to evaluate the models, was created from labels manually added by the authors.

Tasks

We investigate two applications of our ResNet model to streetscape imagery: automatically *validating* crowdsourced labels and automatically *labeling* sidewalk accessibility issues. For *validation*, we input previously supplied manual sidewalk accessibility labels—in the form of an image crop around the label plus positional and geographic features—to our ResNet model. Here, our model outputs its own prediction, which could be used to confirm the human label or flag it for further review. For *labeling*, we input an entire GSV panorama and our model automatically finds and classifies sidewalk accessibility problems. Again, here low-confidence predictions could be crowdsourced for manual validation (similar to [32]).

Because the tasks are different, we train separate ResNet models for each. Just as with humans [32], finding and labeling problems is significantly harder than validating existing labels because the former requires a full scan of an image rather than assessing just a pre-cropped portion.

Validation Task

For *validation*, we first extract crops around each human-supplied point label as well as corresponding positional and geographic features. We then input these features into our model for classification, which outputs a 5x1 vector of confidence scores, one for each of the four label types: *curb ramp*, *missing curb ramp*, *obstruction*, and *surface problem*, plus an additional *null* confidence used to predict the absence of a label. To evaluate performance, the model's precision and recall are computed on the ground truth dataset.

Center Crop Generation. For the validation task, image crops are generated using the techniques described above: an auto-sized crop is extracted centered on each *x,y* point label (Table 2). To predict the *absence* of an accessibility problem, we also generate a set of *null* crops via uniform random sampling of unlabeled (lacking a structure of interest) *x,y* panorama coordinates. If a randomly-sampled *null* crop overlaps a crop containing a label, it is discarded. *Null* crops are sampled only between the horizon and 1600 pixels above the bottom of the panorama, which removes extraneous 'sky' and 'road' imagery. These ranges were determined by the maximum y-axis range of all crowdsourced labels used for training.

Labeling Task

While the *validation* task is focused on validating manually labeled panoramas, the *auto-labeling* task is focused on automatically finding and classifying sidewalk accessibility problems in GSV panoramas. Here, we use a standard *sliding window* approach [22], which breaks the panorama into small, overlapping crops that are then passed into the neural network for classification.

The neural network outputs a 5x1 prediction vector for each crop, from which we compute a single predicted class by taking the `argmax` to find the class with the greatest confidence. Crops with a predicted class of *null* are ignored. The remaining predictions are then clustered using *non-maximum suppression* [22]. That is, overlapping predictions for a given label type are grouped together, and the prediction with the highest confidence is kept, while weaker predictions are suppressed. We used an overlap threshold of 150 pixels determined via qualitative evaluation. A final culling step is then applied to remove predictions with low confidence, adjustable with a tunable hyperparameter, γ—a larger value of γ results in fewer false positives, improving precision, while a smaller γ results in fewer false negatives, improving recall. After culling, the model outputs the final predicted labels.

To evaluate performance, we compare the model-predicted labels to the author-supplied labels on panoramas from the ground truth dataset. To determine which predictions are 'correct' and which are not, each label from the ground truth dataset is assigned a (width, height) proportional to its depth in the panorama, using the same algorithm as is used for producing crops. If the distance between the predicted label and the ground truth label is $\leq \texttt{max}(\text{width, height})$ of the ground truth crop, the predicted label is marked *correct*. Each time a prediction is marked as correct, the corresponding ground truth label is marked as 'accounted for' and no other predictions can be marked as correct with it. This prevents double-counting of correct predictions: the number of correct predictions can never exceed the number of ground truth labels.

	Centered Crops	Sliding Window Crops
Curb Ramp	130,314	159,039
Missing Ramp	17,173	21,552
Obstruction	19,583	25,162
Sfc Problem	7,592	9,933
Null Crop	49,248	139,389
Total	**223,910**	**355,075**

Table 2: Number of training crops for the validation and labeling models produced using the center-crop and sliding window techniques, respectively.

Sliding Window Crop Generation. For the labeling task, models are trained on crops produced using a sliding window (Table 2). Each panorama in the training set is partitioned into a regular 'grid' of overlapping crops with the centers a *stride length* distance apart. Any crop from the grid which contains a label from the dataset within $\frac{\text{stride}}{\sqrt{2}}$ pixels of its center is labeled accordingly; any crop containing multiple labels of different types is ignored. Sliding window crops always overlap, so labeled items from the dataset often appear in multiple crops. Therefore, the set of sliding window crops is larger than those created for training the *validation* model using the centered-cropping technique. Crops containing no labeled sidewalk problems are assigned a label of *null.* As the vast majority of sliding window crops from any panorama are *null*, all but three uniformly randomly sampled *nulls* per panorama are discarded in order to prevent dataset imbalance.

Stride length is an adjustable hyperparameter, set heuristically in our experiments at 100 pixels in both the vertical and horizontal direction. A smaller stride increases computation time as it produces more crops but also results in fewer false negatives, since the probability of producing a crop centered around a given label is increased. The window size at each crop location is determined using the same depth-proportioned algorithm as with the centered crops.

RESULTS

Overall, our models appear to substantially improve upon the performance of previous machine learning-based sidewalk assessment approaches (*e.g.,* Tohme [32] and Sun *et al.* [59])—though direct comparisons are difficult due to a lack of open datasets and software. Perhaps more impressively, our precision and recall on the auto-labeling task meets or exceeds the performance of a group of five human labelers in Hara *et al.* [30]. Below, we examine the performance of our ResNet-based models on our two tasks (addressing R1), analyze the effect of our contextual (non-image) input features (R2) and training data amount (R3) on performance, and investigate the generalizability of our models across cities (R4).

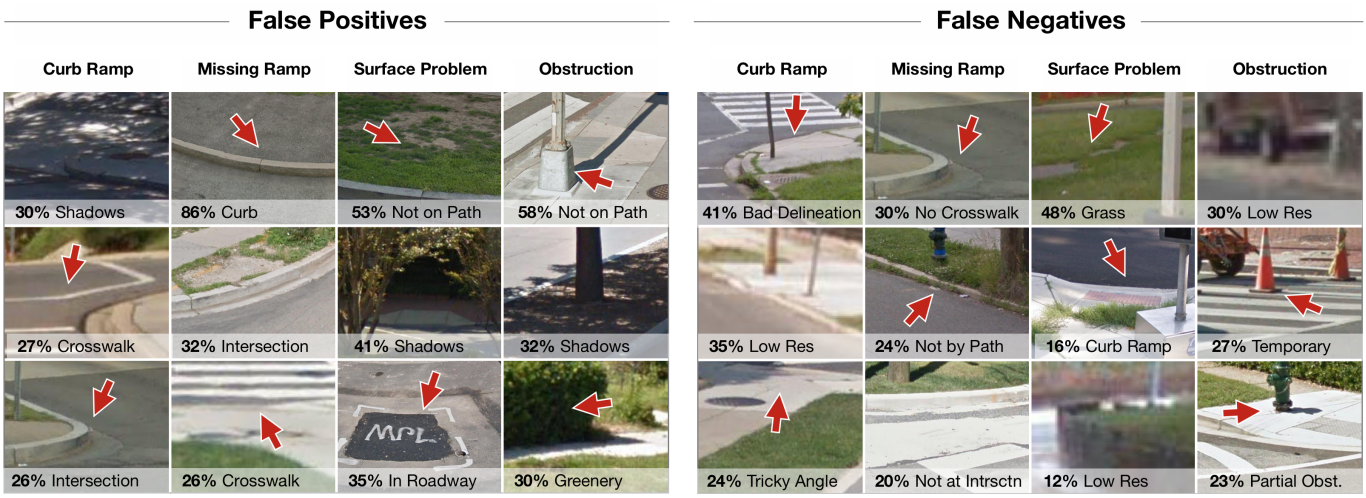
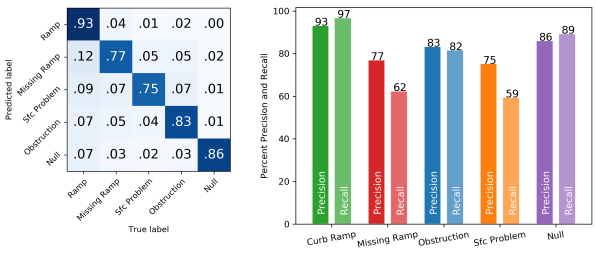


Figure 4: To qualitatively assess our auto-validation model, we randomly selected and manually reviewed approximately 50 false positive and 50 false negative errors per label type (414 assessments total). We present the top three most common errors for both categories above. For example, 30% of falsely identified curb ramps contained inconsistent lighting due to shadows, which confused our model.

R1: Evaluating Auto-Validation Performance

Overall, the average precision and recall of the validation model was 81.3% and 77.2%, respectively. The model performed best on *curb ramps*, with a precision of 93.2% and recall of 96.8% and *obstructions* with 83% precision and 82% recall while *surface problems* were the most challenging, with a precision of 75.32% and a recall of 59.45%. Surface problems vary widely in appearance (*e.g.,* cracks, grass, upended concrete) and location (*e.g.,* anywhere on a street segment), which confounds our model. Importantly, the auto-validation model performs well on *null* crops containing no labeled feature; correctly identifying almost 90% of these instances.

To better understand performance, we created a confusion matrix (Figure 5a), which helps highlight inter-class errors. This matrix is normalized by dividing each box by the number of actual examples of that type, such that each column sums to 1, sans rounding; the numbers down the diagonal are therefore the recall for that label type. The most common confusion of *missing curb ramps* for *curb ramps* is likely due to image and contextual similarities: they both appear at the interface between road and sidewalk and occur at street corners.



(a) Confusion matrix with recall on the diagonal.

(b) Bar plot showing precision and recall for each label type.

Figure 5: Model performance on the validation task.

Qualitatively Assessing Auto-Validation Errors

To qualitatively assess auto-validation performance, we randomly selected and manually reviewed 50 false positive and 50 false negative errors per label type (a total of 414 manual inspections). A *false positive* error occurs when our model predicts a label but one does not exist. A *false negative* is when our model does *not* predict a label but one actually exists. A single researcher inductively analyzed the auto-validation results with an iteratively created codebook—a code set was produced for each label type. Two additional researchers checked the application of these codes for verification.

In analyzing false positives, we observed three key sources of error: camera or imagery limitations, contextual limitations (*i.e.,* a correct classification required contextual knowledge not yet captured by our model), and/or confounding object similarity. For imagery limitations, 30% of falsely identified *curb ramps* and 41% of *surface problems* were due to shadows or poor lighting conditions. For contextual limitations, 53% of *surface problem* errors and 58% of *obstruction* errors were due to correctly identifying objects that were *not* on the pedestrian pathway—a challenging contextual distinction to make. Finally, for object similarity, 86% of *missing curb ramp* false positives were due to misidentifying a normal curb as a missing ramp. Inferring *missing curb ramps* is also challenging due to contextual limitations: that is, determining whether a location actually requires a ramp—a problem that humans also struggle with [55].

For false negatives, common sources of errors included: low resolution imagery, poor object delineation, challenging camera angles, and partial occlusions or obstructions. For example, over 30% of false negatives for *curb ramps* and *obstructions* were due to resolution issues. For poor delineation, 41% of false negative *curb ramps* were due to ramps blending into the scene and 21% were due to 'tricky' camera angles. Finally, for partial occlusions, 23% of false negative *obstructions* were due to co-existing objects in the imagery obscuring the target.

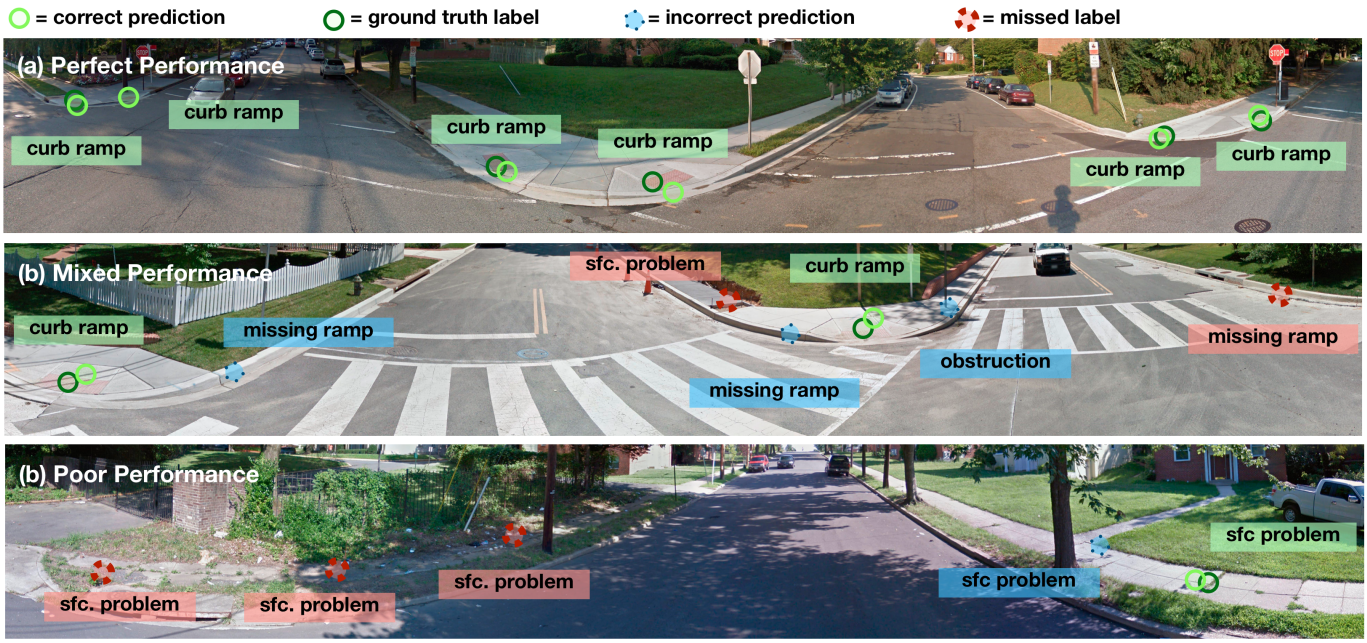


Figure 6: Example results from the labeling task, shown with predicted and ground truth labels. **(a)** Perfect performance. All features detected. **(b)** Mixed performance. Ramps were accurately detected, but the missing ramp was missed, as was the surface problem. Several false positives were predicted. **(c)** Poor performance. One surface problem was detected, but a long section of damaged, and dirty sidewalk pavement was missed. A false surface problem was added.

R1: Evaluating Auto-Labeling Performance

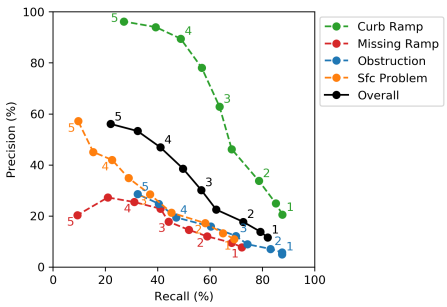


Figure 7: Precision-recall curve showing model performance on the labeling task. γ values are shown next to data points.

As expected, for the auto-labeling task, our overall performance drops to 47.0% for precision and 41.2% for recall (with γ=4). *Curb ramps* perform best with a precision and recall of 89.8% and 48.9% while *missing curb ramps* perform worse with a precision=25.5% and recall=31.0%. In comparison to auto-validation, the drop in performance makes sense: unlike auto-validation where our model is supplied a pre-cropped image around a human-supplied label, the auto-labeling task searches through an entire panorama attempting to find and classify problems. To more deeply examine performance and the effect of our hyperparameter γ, we created a precision-recall curve (Figure 7) showing performance for each label type, varying by γ. An elbow appears for most of the curves at $\gamma \approx 4$, which optimizes both precision and recall. Figure 6 shows representative examples of our labeling results illustrating good, mixed, and poor performance, computed with $\gamma = 4$. Below, we situate our results in the literature.

Comparison to Human Performance

Comparisons to prior work are difficult due to differing datasets, ground truth, and the specific labeling tasks. Prior labeling systems, both manual and automatic, make differing tradeoffs between precision and recall. By varying γ, we attempt to achieve a fair comparison and find that in some cases our automated system's performance meets or even exceeds that of human labelers. For example, in Hara *et. al.* [30], crowdworkers were asked to find and draw bounding boxes around *missing curb ramps*, *sidewalk obstructions*, *surface problems*, and *prematurely ending sidewalks* in a manually curated GSV-based image dataset. To evaluate performance, Hara examined overall pixel-level overlap (intersection over union area) of crowdworker-placed labels *vs.* researcher-placed ground truth. On the multi-class labeling task, single crowdworker labels (one labeler per image) achieved an overall precision of 34% and recall of 26%, while a majority-vote solution with five labelers per image achieved 37% precision and 46% recall. At $\gamma = 3.5$, our purely automated model achieves an overall precision of 38.6% and recall of 49.7%, surpassing the majority-vote human labeler on both metrics (Figure 7). For more subjective labels such as surface problems, we can achieve 58% recall at 17% precision ($\gamma = 2$), which compares favorably to the 26% human accuracy (precision and recall were not reported) with five-person majority vote in [30].

More recently, Hara *et al.*[32] found that crowdworkers using the *svLabel* tool were able to manually find and label curb ramps in GSV panoramas with a precision and recall of 85% and 89%, respectively. In comparison, with $\gamma = 4$, our auto-labeling model achieves a precision and recall of 89.8% and 48.9% for curb ramps. By varying $\gamma = 0$ to emphasize recall,

our model can find 89.8% of curb ramps (recall) but at a cost of 18.5% precision. While here our auto-labeling performance does not yet reach human levels, the task in [32] was far simpler as it focused on a single accessibility feature (curb ramps) *vs.* our multi-class labeling problem.

Comparison to Automated Labeling in Prior Work

In addition to manual labeling, we also compare our performance to two state-of-the-art automated sidewalk labeling systems (Table 3). In [32], Hara *et al.* introduced *svDetect*, which combines a *Deformable Part Model*[21] with a *Support Vector Machine (SVM)* to automatically detect curb ramps in GSV panoramas. svDetect resulted in 67% recall and 26% precision (where correctness was measured as 20% overlap with ground truth bounding boxes). In comparison, with $\gamma = 2$, our multi-class labeling model recalls 78% of curb ramps at 33% precision—an increase of 18% and 27%, respectively. More recently, Sun *et al.* [59] applied a Siamese neural network [14] to identify *missing curb ramps* in GSV panoramas and achieved a 27% recall (precision is not reported). For *missing curb ramps*, our ResNet-based neural network achieves more than double the recall (58.6%) at a precision of 12%—a substantial improvement.

		Tohme[32]	**Our Model**	**Change**
Curb Ramp	*precision*	26%	33.7%	+30%
	recall	67%	78.7%	+17%
		Sun *et al.*[59]	**Our Model**	**Change**
Missing Ramp	*precision*	not reported	12.0%	N/A
	recall	27%	58.6%	+117%

Table 3: Improvements over prior work in fully automated ML-based methods for labeling curb ramps and missing curb ramps in GSV panoramas. Our model's precision and recall are generated with γ=2.

R2: Effect of Contextual Input Features On Performance

Our modified ResNet-18 architecture incorporates non-image features such as the position of a crop in a scene (*i.e.,* positional features) as well as the relative geographic location of a panorama in the city (*i.e.,* geographic features). To investigate the effect of these additional "context" features on performance, we experimented with three different input feature compositions: *image-only*, *image + positional*, and *all features* (*image + positional + geographical*). For the experiments, models were trained on the center-crop training set and evaluated on our ground truth dataset. The results are shown in Table 4.

Overall, there are only marginal differences in performance: recall improves from 79.6% to 80.1% while precision drops from 80.3% to 79.7%; however, each label type is impacted differently. For example, finding *surface problems* jumps from 48.5% with *image-only* features to 56.7% with *all features* while recall decreases for *obstructions*: from 73.1% to 69.8%. Both *curb ramps* and *missing curb ramps* benefit from the context features (increasing 3-5%), perhaps because of predictable geographic patterns—*e.g.,* they tend to occur on corners at the end of street segments. Relatedly, because *null* crops are randomly sampled, there are no positional or geographic patterns to leverage, so the additional features do not help recall (indeed, performance decreases by 2% for *nulls*).

	Precision			**Recall**		
	Image	Img. + Position	All	Image	Img. + Position	All
Overall	80.3	79.5	79.7	79.6	80.0	80.1
Curb Ramp	81.5	80.1	79.7	90.7	93.2	93.6
Missing Ramp	80.2		80.6	50.7		51.8
Obstruction	84.9	84.9	85.4	73.0	71.9	69.8
Sfc Problem	79.3		73.5	48.5	50.8	56.7
Null	75.6		79.3	89.4		

Table 4: Changes in precision and recall with the addition of positional features and geographic features. Color shading indicates increase or decrease in performance from baseline (image features only).

R3: Performance as a Function of Training Set Size

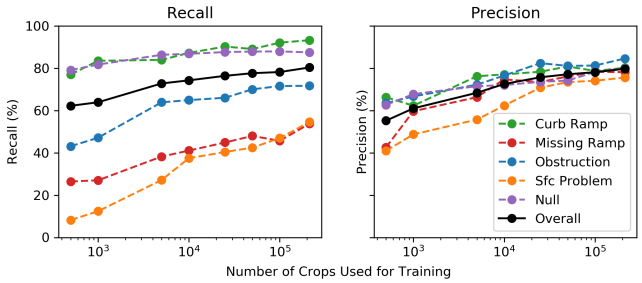


Figure 8: Performance overall and by feature type as the size of the training set increases. Note the log scale on the *x* axis.

As noted in the *Introduction*, previous work applying computer vision to sidewalk assessment has used relatively small datasets. For example, Hara *et al.*'s used 2,877 *curb ramp* image crops [32] and Sun *et al.* used 647 *missing curb ramps* [59]. To explore the effect of dataset size on performance, we trained our auto-validation model on increasingly large, randomly sampled subsets of our training dataset. Our results are shown in Figure 8. Unsurprisingly, performance is positively correlated with training set size. With only 1,000 crops, our overall precision and recall was 61% and 63.9%, which improves to 79.7% and 80.4% with the full training set (213,000 crops). Interestingly, even at the maximum training set size, a plateau is not yet reached, particularly for the worst performing classes (*surface problems* and *missing curb ramps*)—which suggests that even more training data would be beneficial.

R4: Exploring Cross-City Generalizability

While the results presented above were focused on Washington DC, ideally a model trained on one city's streetscape images would generalize to other cities. However, as noted in [32, 55], sidewalk infrastructure can vary in quality and design across geographic areas and neighborhood types (*e.g.,* suburban residential vs. downtown commercial)—impacting visual appearance. To examine the cross-city generalizability of our models, we use recent open datasets from two new Project Sidewalk deployment cities: Seattle, WA (a major US city on the west coast with 750,000 residents) and Newberg, OR (a small town outside of Portland, OR with 22,000 residents).

In total, Seattle and Newberg have over 80,000 new GSV-based crowdsourced sidewalk image labels; however, for our experiments, we use a subset: the 9,535 labels created or validated by members of the Project Sidewalk team (to ensure

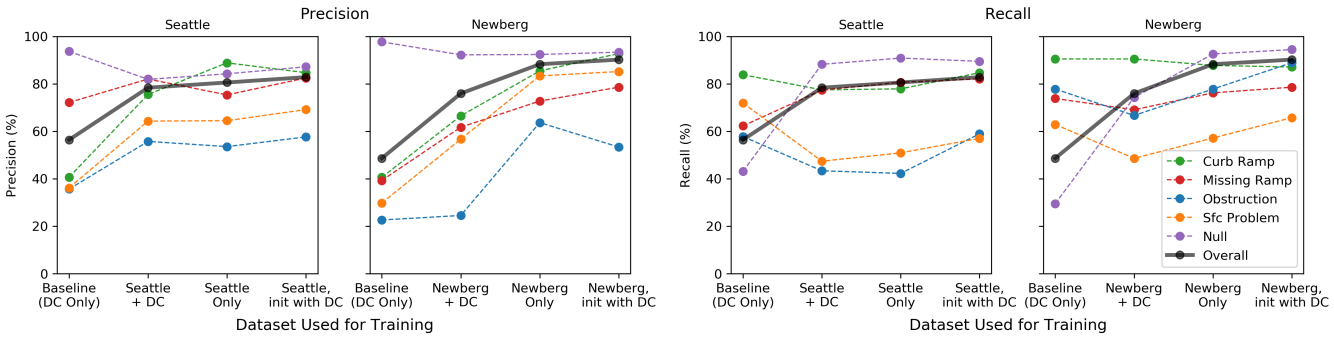


Figure 9: Precision and recall of our auto-validation model trained using four different approaches and tested on Seattle and Newberg.

maximum quality). For each city, an additional 1,500 *null* labels were randomly sampled from the GSV panoramas. For each label, we used center cropping (described in 3.4) to transform points to image crops. Crops were randomly partitioned into train and test sets with an 80/20 split (see Table 5).

	Seattle		Newberg	
	Train	**Test**	**Train**	**Test**
Number of Panos	3,101	775	3,199	801
Curb Ramp	1,648	412	1,814	453
Missing Ramp	1,194	298	588	147
Obstruction	633	158	307	76
Surface Problem	1,000	250	446	111
Null	1,202	298	1,199	301
All Labels	**5,677**	**1,416**	**4,354**	**1,088**

Table 5: Table of train and test set sizes for Seattle and Newberg

In total, we conducted four cross-city experiments—all use the auto-validation model but vary in training set composition. Each model was evaluated on the test set for its respective city (either Seattle or Newberg). The four models included:

1. **Baseline Model**: trained on DC only.
2. **DC + New City Model**: trained on both DC and the other city (either Seattle or Newberg).
3. **New City-only Model**: trained on only Seattle or only Newberg without any DC data.
4. ***(Best performing)* New City-only Model Initialized with DC**: same as New City-only, but initialized with the weights from the DC model ('pre-trained') before training on the new city data.

Results are presented in Figure 9. Both cities demonstrate similar trends in performance across the four models. The Baseline Model—which is trained on DC-only but tested on the new cities—resulted in 55.6% precision and 56.3% recall for Seattle and 46.0% and 48.6% for Newberg. Interestingly, some label types performed quite well—suggesting some uniformity across cities. For example, *curb ramps* achieved a recall of 83.8% in Seattle and 90.5% in Newberg (though precisions were at 40% for both cities). This high cross-city performance is perhaps because *curb ramps* are the only *designed* urban feature in our dataset, which likely resulted in visual and contextual consistency—the other label types are due to dilapidation (*e.g., surface problem*) or the lack of urban design (*e.g.,* missing curb ramps).

For the other three models, even training on a small amount of new city-specific data results in significant improvement over baseline, with Seattle+DC improving from 55.6% to 71.9% for precision and 56.4% to 78.4% for recall, and Newberg+DC improving from 46.0% to 60.3% for precision and 48.6% to 75.9% for recall.

Overall, the best performing method was to use DC data to learn neural network weights that were then used to initialize ("pre-train") a model that was further trained on each individual city's data (either Seattle or Newberg). Our results are promising: the Seattle model pre-trained on DC achieved 76.2% precision and 82.8% recall while the Newberg version achieved 80.6% precision and 90.2% recall. These results are competitive with the auto-validation model trained and tested on DC data (from the *R1: Evaluating Auto-Validation Performance* sub-section), with overall 81.3% precision and 77.2% recall. The results suggest that our models can perform well with only a small amount of new training data per city.

DISCUSSION

We discuss biases and potential dangers in automating sidewalk assessments, describe dataset limitations and how they may impact our results, enumerate future work relevant to machine learning, and reflect on possible uses for and impacts of automatically assessing sidewalks.

Biases in Automating Sidewalk Assessment

Any Artificial Intelligent (AI) or Machine Learning (ML) system contains intrinsic norms, values, and biases [25, 26]. Ours is no different. These exist at many levels, from the data collection [25] to the design and implementation of the machine learning approach itself [26]. In this paper, we leverage the Project Sidewalk dataset, which offers highly granular, geolocated sidewalk accessibility labels; however, the four label types (*curb ramps*, *missing curb ramps*, *obstructions*, and *surface problems*) do not comprehensively capture sidewalk accessibility. For example, no labels exist for crosswalks, accessible pedestrian signals (*e.g.,* audio-based stoplights), stairs, or accessible public transit stops.

Our ML approach may also introduce new biases. For instance, our system, though successful in a small number of cities, may not work in locations where sidewalk problems are less obvious and detectable or where streetscape imagery is not widely

available. Failures in our ML model may incorrectly inform policymakers that certain sidewalks are accessible, hampering appropriate sidewalk transportation and funding. We emphasize that 'on-the-ground' investigations should accompany the use of these ML tools to provide a mechanism for continued community involvement.

Dataset Limitations

Neural networks are notably data-hungry as their expansive parameter space requires a large amount of training data. Our project is enabled by the size and richness of the crowdsourced Project Sidewalk dataset. As expected, however, crowdsourced data is noisy, which can undermine ML model training. the task of sidewalk assessment for accessibility problems is inherently subjective and consistent labeling is difficult even for humans. For example, labelers may place labels differently for the same accessibility problem. Similarly, some sidewalk areas are occluded from view within panorama images, making assessment impossible. While consistent, detailed labeling rules and training help, labelers still face ambiguity—such as, "is this a *surface problem* or an *obstruction*?" "Are pedestrians intended to cross at this intersection?" Finally, in Project Sidewalk, crowdworkers are *not* asked to comprehensively label each visited GSV panorama—instead, their focus is on finding accessibility problems. So, once a problem is labeled, it need not be labeled in adjacent panoramas. This 'under-labeling' per panorama may contribute to artificially high false negative rates in our model evaluations.

Future Work: Data and Computer Vision Methods

In this paper, we trained models separately for the *auto-validation* and *auto-labeling* tasks, as this increased performance. For improved trainability and usability, future work should explore the development of a universal model that would be optimal for both tasks.

Collecting additional data will also improve the performance of our models, as shown by our investigations of performance as a function of training set size. We will need more large, labeled sidewalk accessibility datasets from cities around the world, for which we will continue to reach out to potential communities and partners.

Mechanisms for improving human-labeled data quality remain an open question for Project Sidewalk. Also, with more detailed training data, it may become possible to train our system to produce more accurate and nuanced labels, such as the severity of a surface problem or the presence of friction strips on curb ramps. The additional data needed for these tasks is already being collected by Project Sidewalk, with crowdworkers rating the severity of problems on a 1-5 scale. Including these ratings in our model is a logical first step towards increased label detail.

The performance of our neural network may transfer from one city to another to some degree, as shown by our Newberg and Seattle experiments, but will likely be lower due to differences between cities. The ability to transfer trained networks between cities could ease data collection requirements for new areas, but at the cost of potentially excluding communities with drastically different sidewalk infrastructure.

Impact on Sidewalk Accessibility

Our overarching long-term vision to be able to automatically assess the accessibility of cities at scale within hours. This would put a powerful accountability tool in the hands of accessibility advocates, for example when monitoring the adherence of cities to ADA mandates. However, automation of any task brings up a number of complex challenges and ethical considerations that merit further discussion.

While our machine learning approach demonstrates a significant improvement over the state-of-the-art on image-based automated assessment of sidewalk-level accessibility problems, it is important to assess how to use these results to meaningfully affect sidewalk accessibility. In addition to identifying concrete problems to be solved, the results could be aggregated into quantitative accessibility metrics on the street, neighborhood, or city level, as in the work proposing AccessScore [54]. This would allow for users to make more informed choices about neighborhoods, housing, and transportation. We also see potential future applications to the problem of routing users with differing mobility (*e.g.,*[58]), although this task is notably harder and requires more precision than our algorithm can yet provide. For example, even one missed obstruction could render a route impassible.

This paper's approach only leverages a portion of the data available through Project Sidewalk, which now include additional tags on labels such as the presence or absence of friction strips on curb ramps, differentiation between types of obstacles such as street poles or cars, and severity metrics. Incorporating all of these details into automatic classifiers in a manner similar to this work could also result in more usable and accurate metrics for routing and other applications.

CONCLUSION

The overarching goal of our work is to develop fast and accurate sidewalk assessment methods using machine learning to help transform how city governments and citizens alike track, maintain, and use pedestrian infrastructure. In this paper, we have demonstrated a promising deep learning approach for *auto-validating* and *auto-labeling* sidewalks in streetscape imagery. Our ResNet models significantly improve upon the performance of previous automated systems and, in some cases, meet or exceed human labeling performance.

ACKNOWLEDGEMENTS

This work was supported by an NSF grant (IIS-1302338) and a Sloan Research Fellowship. We would like to thank Kavi Dey and Michael Saugstad for their code contributions, Joseph Redmon for his valuable technical guidance, Kevin Jamieson for his support in the early stages, and Richard Anderson for his mentorship.

REFERENCES

1. 2017. Seattle Sidewalk Survey Update. (Aug 2017). https://sdotblog.seattle.gov/2017/08/28/seattle-sidewalk-survey-update/

2. 2019a. 2010 ADA Standards for Accessible Design. (Jul 2019). https://www.ada.gov/2010ADAstandards_index.htm

3. 2019b. Chapter R3: Technical Requirements. (Jul 2019). https://www.access-board.gov/guidelines-and-standards/streets-sidewalks/public-rights-of-way/proposed-rights-of-way-guidelines/chapter-r3-technical-requirements
4. 2019. Cyclomedia.com. (Jul 2019). https://www.cyclomedia.com/us
5. 2019. Mapillary.com. (Jul 2019). https://www.mapillary.com/
6. 2019. Wegoto: Geolocated data acquisition. (Jul 2019). https://www.wegoto.eu/wegoto_data.php?lang=gb
7. Dragan Ahmetovic, Cristian Bernareggi, Andrea Gerino, and Sergio Mascetti. 2014. ZebraRecognizer: Efficient and precise localization of pedestrian crossings. In *Proceedings - International Conference on Pattern Recognition*. Institute of Electrical and Electronics Engineers Inc., 2566–2571. DOI: http://dx.doi.org/10.1109/ICPR.2014.443
8. Dragan Ahmetovic, Roberto Manduchi, James M. Coughlan, and Sergio Mascetti. 2015. Zebra Crossing Spotter: Automatic Population of Spatial Databases for Increased Safety of Blind Travelers. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15*. ACM Press, New York, New York, USA, 251–258. DOI: http://dx.doi.org/10.1145/2700648.2809847
9. Dragan Ahmetovic, Roberto Manduchi, James M. Coughlan, and Sergio Mascetti. 2017a. Mind Your Crossings. *ACM Transactions on Accessible Computing* 9, 4 (apr 2017), 1–25. DOI: http://dx.doi.org/10.1145/3046790
10. Dragan Ahmetovic, Roberto Manduchi, James M. Coughlan, and Sergio Mascetti. 2017b. Mind Your Crossings: Mining GIS Imagery for Crosswalk Localization. *ACM Transactions on Accessible Computing* 9, 4 (apr 2017), 1–25. DOI: http://dx.doi.org/10.1145/3046790
11. Pablo F. Alcantarilla, Simon Stent, Germán Ros, Roberto Arroyo, and Riccardo Gherardi. 2018. Street-view change detection with deconvolutional networks. *Autonomous Robots* 42, 7 (01 Oct 2018), 1301–1322. DOI: http://dx.doi.org/10.1007/s10514-018-9734-5
12. Amir Arsalan Soltani, Haibin Huang, Jiajun Wu, Tejas D Kulkarni, and Joshua B Tenenbaum. 2017. Synthesizing 3d shapes via modeling multi-view depth maps and silhouettes with deep generative networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1511–1519.
13. Anke M. Brock, Jon E. Froehlich, João Guerreiro, Benjamin Tannert, Anat Caspi, Johannes Schöning, and Steve Landau. 2018. SIG: Making Maps Accessible and Putting Accessibility in Maps. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. ACM Press, New York, New York, USA, 1–4. DOI: http://dx.doi.org/10.1145/3170427.3185373
14. Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature Verification Using a "Siamese" Time Delay Neural Network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems (NIPS'93)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 737–744. http://dl.acm.org/citation.cfm?id=2987189.2987282
15. Carlos Cardonha, Diego Gallo, Priscilla Avegliano, Ricardo Herrmann, Fernando Koch, and Sergio Borger. 2013. A Crowdsourcing Platform for the Construction of Accessibility Maps. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility (W4A '13)*. ACM, New York, NY, USA, 26:1—-26:4. DOI: http://dx.doi.org/10.1145/2461121.2461129
16. Mingmei Cheng, Yigong Zhang, Yingna Su, Jose M. Alvarez, and Hui Kong. 2018. Curb Detection for Road and Sidewalk Detection. *IEEE Transactions on Vehicular Technology* 67, 11 (nov 2018), 10330–10342. DOI: http://dx.doi.org/10.1109/TVT.2018.2865836
17. KM Christensen, JM Holt, and JF Wilson. 2010. Effects of Perceived Neighborhood Characteristics and Use of Community Facilities on Physical Activity of Adults With and Without Disabilities. *Preventing chronic disease* 7, 5 (2010), A105–A105. https://europepmc.org/articles/pmc2938399
18. Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. (apr 2016). http://arxiv.org/abs/1604.01685
19. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.
20. Chaohai Ding, Mike Wald, and Gary Wills. 2014. A Survey of Open Accessibility Data. In *Proceedings of the 11th Web for All Conference (W4A '14)*. ACM, New York, NY, USA, 37:1—-37:4. DOI: http://dx.doi.org/10.1145/2596695.2596708
21. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. 2010. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 9 (Sep. 2010), 1627–1645. DOI: http://dx.doi.org/10.1109/TPAMI.2009.167
22. David. Forsyth and Jean. Ponce. 2003. Detecting Objects in Images. In *Computer Vision: A Modern Approach* (2 ed.). Prentice Hall, Chapter 17, 693. http://luthuli.cs.uiuc.edu/

23. Jon E. Froehlich, Anke M. Brock, Anat Caspi, João Guerreiro, Kotaro Hara, Reuben Kirkham, Johannes Schöning, and Benjamin Tannert. 2019. Grand Challenges in Accessible Maps. *Interactions* 26, 2 (Feb. 2019), 78–81. DOI:http://dx.doi.org/10.1145/3301657

24. Orazio Gallo, Roberto Manduchi, and Abbas Rafii. 2008. Robust curb and ramp detection for safe parking using the Canesta TOF camera. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 1–8. DOI: http://dx.doi.org/10.1109/CVPRW.2008.4563165

25. Megan Garcia. 2016. Racist in the machine: The disturbing implications of algorithmic bias. *World Policy Journal* 33, 4 (2016), 111–117.

26. Sara Hajian, Francesco Bonchi, and Carlos Castillo. 2016. Algorithmic Bias: From Discrimination Discovery to Fairness-aware Data Mining. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 2125–2126. DOI: http://dx.doi.org/10.1145/2939672.2945386

27. William Hansen, Ned Kalapasev, Amy Gillespie, Mary Singler, and Marsha Ball. 2009. Development of a Pedestrian Walkability Database of Northern Kentucky Using Geographic Information Systems (GIS). *Journal of Physical Activity and Health* 6, 3 (may 2009), 374–385. DOI:http://dx.doi.org/10.1123/jpah.6.3.374

28. Kotaro Hara, Christine Chan, and Jon E Froehlich. 2016. The Design of Assistive Location-based Technologies for People with Ambulatory Disabilities: A Formative Study. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1757–1768. DOI: http://dx.doi.org/10.1145/2858036.2858315

29. Kotaro Hara, Victoria Le, and Jon Froehlich. 2012. A feasibility study of crowdsourcing and google street view to determine sidewalk accessibility. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '12*. ACM Press, New York, New York, USA, 273. DOI: http://dx.doi.org/10.1145/2384916.2384989

30. Kotaro Hara, Vicki Le, and Jon Froehlich. 2013a. Combining crowdsourcing and google street view to identify street-level accessibility problems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press, New York, New York, USA, 631. DOI: http://dx.doi.org/10.1145/2470654.2470744

31. Kotaro Hara, Jin Sun, Jonah Chazan, David W. Jacobs, and Jon Froehlich. 2013b. An Initial Study of Automatic Curb Ramp Detection with Crowdsourced Verification Using Google Street View Images. *HCOMP* (2013). https://www.semanticscholar.org/paper/An-Initial-Study-of-Automatic-Curb-Ramp-Detection-Hara-Sun/25f26ef200b520bf9a834604f007fb818dd9ec8a

32. Kotaro Hara, Jin Sun, Robert Moore, David Jacobs, and Jon Froehlich. 2014. Tohme. In *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14*. ACM Press, New York, New York, USA, 189–204. DOI: http://dx.doi.org/10.1145/2642918.2647403

33. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 770–778. DOI:http://dx.doi.org/10.1109/CVPR.2016.90

34. Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

35. Yusuke Iwasawa, Kouya Nagamine, Ikuko Eguchi Yairi, and Yutaka Matsuo. 2015. Toward an Automatic Road Accessibility Information Collecting and Sharing Based on Human Behavior Sensing Technologies of Wheelchair Users. *Procedia Computer Science* 63 (jan 2015), 74–81. DOI:http://dx.doi.org/10.1016/J.PROCS.2015.08.314

36. Ian Janssen and Andrei Rosu. 2012. Measuring sidewalk distances using Google Earth. *BMC Medical Research Methodology* 12, 1 (dec 2012), 39. DOI: http://dx.doi.org/10.1186/1471-2288-12-39

37. Bumjoon Kang, Jason Y. Scully, Orion Stewart, Philip M. Hurvitz, and Anne V. Moudon. 2015. Split-Match-Aggregate (SMA) algorithm: integrating sidewalk data with transportation network data in GIS. *International Journal of Geographical Information Science* 29, 3 (mar 2015), 440–453. DOI: http://dx.doi.org/10.1080/13658816.2014.981191

38. Corinne E. Kirchner, Elaine G. Gerber, and Brooke C. Smith. 2008. Designed to Deter: Community Barriers to Physical Activity for People with Visual or Motor Impairments. *American Journal of Preventive Medicine* 34, 4 (apr 2008), 349–352. DOI: http://dx.doi.org/10.1016/J.AMEPRE.2008.01.005

39. Reuben Kirkham, Romeo Ebassa, Kyle Montague, Kellie Morrissey, Vasilis Vlachokyriakos, Sebastian Weise, and Patrick Olivier. 2017. WheelieMap: An Exploratory System for Qualitative Reports of Inaccessibility in the Built Environment. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, 38:1—-38:12. DOI: http://dx.doi.org/10.1145/3098279.3098527

40. Stephen Law, Brooks Paige, and Chris Russell. 2018. *Take a Look Around: Using Street View and Satellite Images to Estimate House Prices*. Technical Report. https://doi.org/10.475/123

41. Anthony Li, Manaswi Saha, Anupam Gupta, and Jon E. Froehlich. 2018. Interactively Modeling and Visualizing Neighborhood Accessibility at Scale. Association for Computing Machinery (ACM), 444–446. DOI: http://dx.doi.org/10.1145/3234695.3241000

42. Xiaojiang Li, Carlo Ratti, and Ian Seiferling. 2017. Mapping Urban Landscapes Along Streets Using Google Street View. (2017). https://www.semanticscholar.org/paper/Mapping-Urban-Landscapes-Along-Streets-Using-Google-Li/ef541adc3c536c7d43ff1e625f5a8a5b7dba3455

43. Xiaojiang Li, Chuanrong Zhang, Weidong Li, Robert Ricard, Qingyan Meng, and Weixing Zhang. 2015. Assessing street-level urban greenery using Google Street View and a modified green view index. *Urban Forestry & Urban Greening* 14, 3 (2015), 675–685. DOI: http://dx.doi.org/10.1016/j.ufug.2015.06.006

44. Yi Lu. 2018. The Association of Urban Greenness and Walking Behavior: Using Google Street View and Deep Learning Techniques to Estimate Residents' Exposure to Urban Greenness. *International Journal of Environmental Research and Public Health* 15, 8 (jul 2018), 1576. DOI: http://dx.doi.org/10.3390/ijerph15081576

45. Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the Limits of Weakly Supervised Pretraining. *CoRR* abs/1805.00932 (2018). http://arxiv.org/abs/1805.00932

46. Andrew May, Christopher J. Parker, Neil Taylor, and Tracy Ross. 2014. Evaluating a concept design of a crowd-sourced âĂŸmashup' providing ease-of-access information for people with limited mobility. *Transportation Research Part C: Emerging Technologies* 49 (dec 2014), 103–113. DOI: http://dx.doi.org/10.1016/J.TRC.2014.10.007

47. Christopher Mitchell. 2006. Pedestrian Mobility and Safety: A Key to Independence for Older People. *Topics in Geriatric Rehabilitation* 22, 1 (2006), 45–52.

48. Ali Mollahosseini, David Chan, and Mohammad H Mahoor. 2016. Going deeper in facial expression recognition using deep neural networks. In *2016 IEEE Winter conference on applications of computer vision (WACV)*. IEEE, 1–10.

49. Emilio Soria Olivas, Jose David Martin Guerrero, Marcelino Martinez Sober, Jose Rafael Magdalena Benedito, and Antonio Jose Serrano Lopez. 2009. *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes*. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA.

50. OpenSidewalks.com. OpenSidewalks. (????). https://www.opensidewalks.com/

51. Catia Prandi, Paola Salomoni, and Silvia Mirri. 2014. mPASS: Integrating people sensing and crowdsourcing to map urban accessibility. In *2014 IEEE 11th Consumer Communications and Networking Conference (CCNC)*. IEEE, 591–595. DOI: http://dx.doi.org/10.1109/CCNC.2014.6940491

52. Catia Prandi, Paola Salomoni, Marco Roccetti, Valentina Nisi, and Nuno Jardim Nunes. 2016. Walking with Geo-Zombie: A pervasive game to engage people in urban crowdsourcing. In *2016 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 1–5. DOI: http://dx.doi.org/10.1109/ICCNC.2016.7440545

53. Stuart J. Russell and Peter Norvig. 2010. *Artificial intelligence: a modern approach*. Prentice Hall.

54. Manaswi Saha, Kotaro Hara, Soheil Behnezhad, Anthony Li, Michael Saugstad, Hanuma Maddali, Sage Chen, and Jon E. Froehlich. 2017. A Pilot Deployment of an Online Tool for Large-Scale Virtual Auditing of Urban Accessibility. Association for Computing Machinery (ACM), 305–306. DOI: http://dx.doi.org/10.1145/3132525.3134775

55. Manaswi Saha, Michael Saugstad, Hanuma Teja Maddali, Aileen Zeng, Ryan Holland, Steven Bower, Aditya Dash, Sage Chen, Anthony Li, Kotaro Hara, Jon Froehlich, and An-Thony Li. 2019. Project Sidewalk: A Web-based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data at Scale. (2019). DOI: http://dx.doi.org/10.1145/XXXXXX.XXXXXX

56. Ken Sakurada and Takayuki Okatani. 2015. Change Detection from a Street Image Pair using CNN Features and Superpixel Segmentation. British Machine Vision Association and Society for Pattern Recognition, 61.1–61.12. DOI:http://dx.doi.org/10.5244/c.29.61

57. Daniel Sinkonde, Leonard Mselle, Nima Shidende, Sara Comai, Matteo Matteucci, Daniel Sinkonde, Leonard Mselle, Nima Shidende, Sara Comai, and Matteo Matteucci. 2018. Developing an Intelligent PostGIS Database to Support Accessibility Tools for Urban Pedestrians. *Urban Science* 2, 3 (jun 2018), 52. DOI: http://dx.doi.org/10.3390/urbansci2030052

58. Adam D. Sobek and Harvey J. Miller. 2006. U-Access: a web-based system for routing pedestrians of differing abilities. *Journal of Geographical Systems* 8, 3 (sep 2006), 269–287. DOI: http://dx.doi.org/10.1007/s10109-006-0021-1

59. Jin Sun and David W. Jacobs. 2017. Seeing What is Not There: Learning Context to Determine Where Objects are Missing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1234–1242. DOI:http://dx.doi.org/10.1109/CVPR.2017.136

60. Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *undefined* (2016). https://www.semanticscholar.org/paper/Inception-v4

61. Zifeng Wu, Chunhua Shen, and Anton van den Hengel. 2019. Wider or Deeper: Revisiting the ResNet Model for Visual Recognition. *Pattern Recognition* 90 (jun 2019), 119–133. DOI: http://dx.doi.org/10.1016/j.patcog.2019.01.006