BusStopCV: A Real-time AI Assistant for Labeling Bus Stop Accessibility in Streetscape Imagery

Minchu Kulkarni¹,² Chu Li³ Jaye Ahn¹,² Katrina Ma³ Zhihan Zhang¹ Michael Saugstad¹ Yochai Eisenberg⁴ Valerie Novack⁵ Brent Chamberlain⁵ Jon E. Froehlich¹
¹Allen School of Computer Science, University of Washington ²MHCI+D Program ³Human-Centered Design and Engineering, University of Washington ⁴Disability and Human Development, University of Illinois Chicago ⁵Landscape Architecture & Environmental Planning, Utah State University

Figure 1: The BusStopCV human+AI data collection workflow: a real-time YOLOv8 CV model automatically detects bus stop features such as shelters and benches in streetscape imagery (left). Users can verify detections via lightweight click interactions (middle) or manually label features not detected by the model (right). Verified bounding boxes turn from dashed to solid lines. See demonstration video in supplementary materials.

ABSTRACT
Public transportation provides vital connectivity to people with disabilities, facilitating access to work, education, and health services. While modern navigation applications provide a suite of information about transit options—including real-time updates about bus or train arrivals—they lack data about the accessibility of the transit stops themselves. Bus stop features such as seating, shelters, and landing areas are critical, but few cities provide this information. In this demo paper, we introduce BusStopCV, a Human+AI web prototype for scalably collecting data on bus stop features using real-time computer vision and human labeling. We describe BusStopCV’s design, custom training with the YOLOv8 model, and an evaluation of 100 randomly selected bus stops in Seattle, WA. Our findings demonstrate the potential of BusStopCV and highlight opportunities for future work.

KEYWORDS
urban accessibility, bus stops, accessible transit systems, computer vision, crowdsourcing

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ASSETS ’23, October 22–25, 2023, New York, NY, USA
© 2023 Copyright held by the owner/author(s).
https://doi.org/10.1145/3597638.3614481

1 INTRODUCTION
Public transportation systems provide many benefits, such as mitigating traffic congestion [9], reducing pollution [9], enhancing economic growth [6], and improving health and well-being [16]; however, significant barriers continue to limit how people with disabilities use public transportation [26]. While a city’s entire transit system requires careful assessment—from getting on a train or bus to finding accessible and safe seating—the transit stop itself is often overlooked, perceived as a mere waiting point instead of an integral part of mobility [1].

Modern navigation tools such as Google Maps [8] and Apple Maps [4] offer real-time bus information but fall short in providing data on essential bus stop accessibility features like seating availability, shelter provisioning, conditions of the landing area, and sidewalk connectivity [5]. Some cities publish open data about bus stop features; however, this practice is bespoke and limited by data collection costs and a lack of data standards [5]. In this demo paper, we introduce a new human+AI approach [3] to semi-automatically gather data on bus stops using real-time computer vision (CV) and
Our work builds on recent efforts in automatic streetscape analysis using deep learning [2, 7, 14, 15, 17, 25] as well as prior human + AI streetscape labeling tools for urban accessibility [12, 13]—all of which demonstrate the potential of emerging CV models in semiautomatically identifying accessibility features in street scenes. In contrast to this growing literature, which uses offline computer vision for analysis, we introduce a novel tool, BusStopCV, that automatically identifies bus stop features in real-time using a custom-trained YOLOv8 model running in the browser. Human labelers can then verify and correct automatic detections and add manual labels. By running a real-time CV model directly in the browser, BusStopCV operates analogously to emerging real-time CV systems in augmented reality (e.g., on wearable headphones); however, with BusStopCV, the pixel data is streamed from precaptured streetview panoramas rather than a live camera.

Most related to our approach is the manual labeling tool, Bus Stop CSI [10, 11], which showed how minimally trained crowd workers could label bus stop landmarks with 87% accuracy in interactive streetview imagery. While promising, Bus Stop CSI focused on identifying navigational landmarks for blind and low-vision pedestrians and did not include AI-assisted labeling. We draw upon Bus Stop CSI and related tools (e.g., Project Sidewalk [20]) for a similar streetview labeling interface with the addition of real-time AI. As an initial prototype, we focused on four bus stop features: shelters, seating, signage, and trash cans, which were informed by transit accessibility literature [5, 18, 21] and our own formative interviews with 29 people. We will include landing area size, shade availability, sidewalk connectivity, and beyond in our future work.

To evaluate BusStopCV, we conducted two studies: first, a technical performance evaluation of the custom-trained YOLOv8 model across 661 bus stop images, demonstrating a F1 Score of ~0.9 for shelters, seating, signage, and trash cans, respectively. Second, a pilot study with one user who used the tool to label 100 bus stop locations randomly sampled in Seattle, WA. Here, BusStopCV was able to automatically identify 87.5% of shelters, 90.7% of seating, 85.23% of signage, and 82.86% of trash cans. The user only needed to correct 12 false positives (0.13 per bus stop) and 28 false negatives (0.31 per bus stop).

In summary, our contributions include: (1) A novel human + AI tool that uses real-time computer vision in the browser to facilitate labeling bus stop features in streetscape imagery; and (2) Initial studies demonstrating the potential of our approach (e.g., high accuracy and responsiveness) while highlighting key areas for improvement. The BusStopCV tool is open source and available on Github.

2 THE BUSSTOPCV PROTOTYPE

The overarching goal of our work is to develop a scalable crowdsourcing system that allows non-experts to quickly and accurately label bus stop accessibility features. The collected geo-located data can then be used to populate city databases (e.g., [24]), be integrated into modern mapping tools (e.g., so users can query the existence of bench availability or shelters), and support future transit planning and inclusive infrastructural development.

To design BusStopCV, we first reviewed literature on bus stop features [5, 18, 21] and synthesized key attributes contributing to accessibility including: (1) urban wayfinding and navigation: the presence and type of signage, connectivity to accessible sidewalks; (2) safety: lighting, high visibility; (3) street furniture: newspaper

---

BusStopCV online repository: https://github.com/ProjectSidewalk/BusStopCV
boxes, poles, and trash cans, which can provide navigational landmarks [10] but can also impede travel if not appropriately positioned (4) comfort: seating, shelter, tree shade; (5) landing areas: accessible landing areas with accessible travel paths to the sidewalk.

To gain further insights on the needs and opportunities for transit accessibility tools, we performed formative interviews with 23 professionals, including metropolitan planners, occupational therapists, travel trainers, community service providers, realtors, ADA compliance professionals, and people who identified as having a disability. Additionally, six people with disabilities (P WD), who use public transportation to travel to or look for work, participated in a focus group about end product utility. All participants confirmed that currently available tools and data (e.g., Google Maps) were inadequate for incorporating accessibility into transportation planning or for trip-planning purposes. Our analysis identified the desire for up-to-date transit accessibility data and a general acceptance of using AI models for collecting data. Participants identified desired features for future development, such as accessible design of destination entrances and safe drop-off spaces for ramp deployment.

Informed by these experiences and prior work in streetscape labeling tools [10, 12, 20, 25], we iteratively designed BusStopCV, starting with sketches and Figma mockups before implementing the tool in JavaScript (frontend) and Java (backend). Unlike previous streetscape labeling tools, we centered the human+AI workflow around an AI-first approach—leveraging recent advances in CV to automatically identify and highlight bus stop accessibility features in real-time. Human input is then utilized for manual verification and correction (Figure 1). As initial work, we incorporate four label types—shelter, seating, signage, trash can—with plans to expand.

To use BusStopCV, users are “virtually transported” to bus stop locations in an immersive GSV-based labeling UI (Figure 2). As users explore the environment in GSV, our real-time CV processes the image and automatically labels bus stop features (Figure 2) with a bounding box and confirmation widget. Users can confirm (√) or deny (×) each detection, and add their own labels at any point to correct false negatives. As the user pans, labels appear to “stick” to the underlying labeled feature, and the CV model is rerun on the current field of view to potentially add new automatic labels. Once all automatic labels have been verified and false negatives corrected (e.g., by adding manual labels), BusStopCV produces a geo-located list of bus stop locations and access features, which can be used in transit analytics and route planning tools.

### 2.1 The BusStopCV Model

To construct our CV subsystem, we first custom-trained a YOLOv8 model by randomly selecting 200 Seattle bus stops drawn from the King County GIS (KCGIS) Open Data repository [24]. We ensured that all bus stops were marked active, geographically distributed, and included a variety of bus stop types with various shelters, seating, and signage. We (virtually) visited each location in GSV and took screenshots from varying distances and angles to ensure a diverse training sample. In total, we collected 661 images from 92% of the selected bus stop locations (one image per panorama). The remaining 8% of the bus stops were either completely occluded, did not have a bus stop, or failed to load the GSV panorama.

To label the training images, we used the image annotation tool Roboflow [19]. We traced the outline of the four label classes in each image and exported the data to YOLOv8 format. In total, our labeled dataset contained 1,707 annotations across the four label types: 433 shelters, 383 seats, 536 signs, and 355 trash cans.

Using this dataset, we trained a YOLOv8 model from scratch (no pretraining) with the Ultralytics [22] platform. For the settings, we used an image batch size of 16, a learning rate of 0.01, three warmup epochs, and defaults for other hyperparameters [23]. We employed the Stochastic Gradient Descent optimizer and 150 epochs to optimize the model’s performance.

To train the model, we performed a random 70/20/10 split on the 661 images for training, validation, and testing. We compared our model to two pre-trained baseline models (YOLOv8n & YOLOv8l) using three standard CV metrics: Recall, Precision, and F1 score. As Table 1 shows, our model achieved impressive F1 scores of nearly 0.9 for all label classes while the baseline models were unable to detect any shelters, signage, or trash cans.

After validating model performance, we integrated the custom-trained YOLOv8 model into BusStopCV using the Open Neural Network Exchange (ONNX) runtime standard. Importantly, we aimed to implement a real-time CV engine such that the human and AI could work seamlessly together. Whenever the GSV panorama loads or the user pans, we send a downsampled (640x640) image of the user’s current streetscape view to the model, which then returns a list of identified features, bounding boxes, and confidence intervals. We used an IoU threshold of 0.7 and a confidence threshold of 0.4. During informal experiments, we found a processing time of 250-500ms. The size of the model was 12.2 MB.

### 2.2 Pilot Study

To examine the full BusStopCV workflow with CV detections and human verification/labelling, we conducted a pilot study with a single participant drawn from our research team who had not used the tool before. For the study dataset, we randomly selected 100 additional bus stops in Seattle, WA—32 had at least one shelter, according to official KCGIS data [24].

<table>
<thead>
<tr>
<th>Method</th>
<th>Shelter Precision</th>
<th>Shelter Recall</th>
<th>Shelter F Score</th>
<th>Seating Precision</th>
<th>Seating Recall</th>
<th>Seating F Score</th>
<th>Signage Precision</th>
<th>Signage Recall</th>
<th>Signage F Score</th>
<th>Trash Can Precision</th>
<th>Trash Can Recall</th>
<th>Trash Can F Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>BusStopCV</td>
<td>92.11%</td>
<td>87.5%</td>
<td>0.897</td>
<td>88.64%</td>
<td>90.7%</td>
<td>0.897</td>
<td>97.4%</td>
<td>85.23%</td>
<td>0.909</td>
<td>93.55%</td>
<td>82.86%</td>
<td>0.879</td>
</tr>
<tr>
<td>YOLOv8n</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>100%</td>
<td>20.51%</td>
<td>0.34</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
<tr>
<td>YOLOv8l</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>100%</td>
<td>25.64%</td>
<td>0.408</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
<td>0</td>
<td>0</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 1: The technical performance results of the custom-trained YOLOv8 BusStopCV model on bus stop feature task on precision, recall, and F1 score.
To begin the study, the participant was instructed to verify and/or manually label all bus stop features—shelters, seating, signage, and trash cans—at each bus stop. If a bus stop was not immediately visible when the GSV scene loaded, participants were told to ignore errant automatic detections from the model and explore the area until the bus stop was found. If multiple bus stops were visible, we asked that only the closest bus stop be labeled. Finally, we asked that each feature be labeled only once (e.g., the user need not label the same bus stop feature from multiple GSV panoramas).

Once a bus stop was fully labeled, the participant clicked the “next stop” button (Figure 2). All labeling was conducted in a single study session on a 13-inch MacBook Pro with a 2.3GHz Quad-Core Intel Core i7 (2560x1600 resolution) and an Intel Iris Pro 1536MB graphics card running macOS Ventura 13.4 and Chrome v114.0.5735.198 (x86_64) in a maximized window state. The pilot study lasted roughly one hour and 15 minutes (approximately 1.2 bus stops/minute).

**Study Results.** In total, the participant successfully labeled 89 out of 100 initial locations, providing a total of 218 inputs (2.45 per bus stop): 178 positive verifications (true positives), 12 negative verifications (false positives), and 28 manual labels (false negatives). For the 11 locations that were unlabeled, either the bus stop was not visible because of occlusion (e.g., a stationary bus), it could not be found or the GSV panoramas failed to load. For 86 of the 89 labeled bus stops, the participant was able to label all features from a single GSV panorama; for the other three, the participant needed to move around with GSV navigation to label from alternative views (e.g., to avoid occlusion).

In terms of performance, BusStopCV automatically identified ~80% of all features with a remarkably low false positive and false negative rate. Across the 89 labeled stops, the user only needed to correct 12 false positives (0.13 per bus stop) and 28 false negatives (0.31 per bus stop) with seating having the highest false positive (FP) rate (10.4%) and trash can the highest false negative (FN) rate (16.2%). To further examine FP and FN performance, we conducted a qualitative assessment of all 40 errors—a representative subset is shown in Figure 3). Common sources of error include poor lighting (e.g., shadows), occlusion (e.g., a vehicle blocking the bus stop), complex backgrounds, and inter-class similarity.

Because BusStopCV’s YOLOv8 model runs in real-time, we were also interested in examining UI responsiveness and perceived lag. We found an average inference time—the time it took to process a GSV view and return a list of detections as bounding boxes—of 528ms. The participant stated that the inference delay “was noticeable but just fast enough to be tolerable.”

### 3 DISCUSSION AND CONCLUSION

In this paper, we introduced a new human+AI workflow and tool for rapidly and accurately labeling accessibility bus stop features in streetscape imagery. Unlike prior streetscape labeling tools [10, 11, 20], BusStopCV adopts an AI-first labeling approach, which minimizes human visual search and labeling time. Through our technical performance evaluation, we showed how our custom-trained YOLOv8 model significantly improves bus stop feature detection and our pilot study demonstrated the effectiveness and potential of the BusStopCV tool. Below, we reflect on limitations and opportunities for future work.

**Limited label types.** Currently, BusStopCV is configured to detect four features: shelters, seating, signage, and trash cans. However, we recognize and acknowledge that other elements such as lighting, the presence of a landing area, and sidewalk connectivity [5, 18, 21] are integral to an accessible bus stop.

**CV Implementation Paradigms.** When designing BusStopCV, we discussed two primary CV implementations: a pre-processed approach where our CV model would analyze all bus stop streetscape

<table>
<thead>
<tr>
<th>Label Type</th>
<th>N (TP + FN)</th>
<th>TP</th>
<th>FN</th>
<th>FP</th>
<th>Avg. Conf. (TP)</th>
<th>Avg. Conf. (FP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shelter</td>
<td>40</td>
<td>35</td>
<td>5</td>
<td>3</td>
<td>95.2%</td>
<td>69.3%</td>
</tr>
<tr>
<td>Seating</td>
<td>43</td>
<td>39</td>
<td>4</td>
<td>5</td>
<td>73.8%</td>
<td>51.0%</td>
</tr>
<tr>
<td>Signage</td>
<td>88</td>
<td>75</td>
<td>13</td>
<td>2</td>
<td>83.4%</td>
<td>64.5%</td>
</tr>
<tr>
<td>Trash Can</td>
<td>35</td>
<td>29</td>
<td>6</td>
<td>2</td>
<td>91.6%</td>
<td>81.0%</td>
</tr>
</tbody>
</table>

Table 2: Pilot study results for BusStopCV’s custom-trained YOLOv8 model showing number of correct labels (N), True Positive (TP), False Positive (FP), False Negative (FN), and Average Confidence. All FPs and FNs were correctly rectified by the user.
images a priori and a real-time CV approach. The former reduces computation on the frontend and increases responsiveness—as the bounding boxes would be pre-computed and each bus stop would be analyzed once and only once regardless of user panning. However, the latter is more flexible, enabling our approach to work on any newly visited bus stop location. Future work should examine these tradeoffs, including frontend hardware requirements (2020 MacBook Pro used in pilot study) and Internet bandwidth.

Enhancing Model Ability. The current version of BusStopCV lacks the ability to maintain label recognition as the user navigates through different views or steps forward in GSV. This means that the system does not identify the same bus stop features when the viewing angle or position changes. Enhancing the model’s ability to ‘understand’ and ‘remember’ label information from contiguous panoramas can provide a more seamless user experience.

3.1 Conclusion
As early work, this study opens up more questions than it answers. Does our human+AI workflow present the best solution for labeling urban accessibility features and how can it be improved? Given the current constraints on feature selection, how can we expand to incorporate a wider array of features, and accommodate a broad range of bus stop designs? One potential solution lies in integrating BusStopCV into a wider ecosystem of tools, such as Project Sidewalk [20], allowing for a more comprehensive analysis of urban spaces. Additionally, how might we best facilitate the recognition of features and their relationships to one another in a way that impacts a diverse range of users (e.g., individuals with vision or mobility disabilities)? Despite the challenges and unanswered questions, we remain hopeful that our tool can help urban planners, policymakers, and advocacy groups foster a more inclusive and equitable urban future. Given the expansive availability of GSV accessibility data, this tool has significant potential to fill the gap in missing bus stop accessibility data.

4 ACKNOWLEDGMENTS
This research was supported by NSF awards #2125087 and #2236277. We are grateful to Jacqueline Kish at Shirley Ryan AbilityLab for help with the interviews.

REFERENCES