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

Minchu Kulkarni\textsuperscript{1,2}, Chu Li\textsuperscript{1}, Jaye Ahn\textsuperscript{1,2}, Katrina Ma\textsuperscript{3}, Zhihan Zhang\textsuperscript{1}, Michael Saugstad\textsuperscript{1}, Yochai Eisenberg\textsuperscript{4}, Valerie Novack\textsuperscript{5}, Brent Chamberlain\textsuperscript{5}, Jon E. Froehlich\textsuperscript{1}

\textsuperscript{1}Allen School of Computer Science, University of Washington \hspace{1em} \textsuperscript{2}MHCI+D Program \hspace{1em} \textsuperscript{3}Human-Centered Design and Engineering, University of Washington \hspace{1em} \textsuperscript{4}Disability and Human Development, University of Illinois Chicago \hspace{1em} \textsuperscript{5}Landscape Architecture & Environmental Planning, Utah State University

1 INTRODUCTION

Public transportation systems provide many benefits, such as mitigating traffic congestion \cite{9}, reducing pollution \cite{9}, enhancing economic growth \cite{6}, and improving health and well-being \cite{16}; however, significant barriers continue to limit how people with disabilities use public transportation \cite{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 \cite{1}.

Modern navigation tools such as Google Maps \cite{8} and Apple Maps \cite{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 \cite{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 \cite{5}. In this demo paper, we introduce a new human+AI approach \cite{3} to semi-automatically gather data on bus stops using real-time computer vision (CV) and

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
Figure 2: The BusStopCV interface showing automatic detections in dotted bounding boxes, user-confirmed automatic detections in solid bonding boxes, and a manually applied label (trash can) as a circular target.
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 a common image annotation tool called Roboflow [19]. In Roboflow, 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 [25]. 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 evaluated the effectiveness of our model compared 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 pano loads or the user pans, we send a downsampled (640x640) image of the user’s current street scene 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/labeling, 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 sessions.

<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>100%</td>
<td>0.64%</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>0</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>0</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.
additional bus stops in Seattle, WA—32 had at least one shelter, according to official KCGIS data [24].

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 Plus 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. We plan to add these in future work.

**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 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 availability.

**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

<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>5</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.
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