Using Deep Learning and Google Street View Images to Assess Bus Stop Amenities

Yilong Dai[a], Luyu Liu[b], Kaiyue Wang[d], Meiqing Li[c], Xiang Yan[d,*]

[a]*Department of Computer & Information Science & Engineering, University of Florida, Gainesville, 32603, FL, USA*
[b]*Department of Geosciences, Auburn University, Auburn, 36849, AL, USA*
[c]*School of Public Administration, University of Central Florida, Orlando, 32801, FL, USA*
[d]*Department of Civil and Coastal Engineering, University of Florida, Gainesville, 32603, FL, USA*

Abstract

The assessment of bus stop amenities is important for providing fundamental data for public transit research, planning, and infrastructure enhancements. So far, public data on the amenities at bus stops have largely been unavailable. This study develops an automated, low-cost, and generalizable approach using Google Street View images and deep learning techniques to evaluate bus stop amenities. Leveraging the latest YOLOv8 model, transfer learning, and a dynamic prediction algorithm, our approach achieves efficient detection of shelters and benches with high accuracy and precision in major Florida cities. Results reveal highly heterogeneous spatial patterns for both shelters and benches within and across cities. Additionally, we conducted several tests to evaluate the transferability of the system to other urban contexts, which shows that highly accurate feature detection results can be achieved through model fine-tuning on a small sample of local data. In summary, the proposed system offers a scalable and efficient solution for large-scale real-time assessment of bus stop amenities, which can inform public transportation research and planning, especially for future transit infrastructure improvements.

Keywords: Bus stop amenities, Computer Vision, Google Street View,

*Corresponding author
Email address: `xiangyan@ufl.edu` (Xiang Yan)

Artificial Intelligence, Transportation Infrastructure, Public Transit

1. Introduction

Bus stops are the initial points of interaction between the public and a transit system. For many, bus stops serve as gateways to opportunities, connecting them to jobs, healthcare, groceries, recreation, and more. The amenities provided at bus stops significantly influence riders' experience and the public's perception of public transportation. Proper bus stop amenities, such as shelters, seating, signage, lighting, and accessibility features, can enhance riders' experience and safety [1]. Studies have shown that the availability of bus stop amenities can promote bus ridership [2]. Also, bus stop amenities can increase the attractiveness of fixed-route services to passengers with disabilities [3, 4], thereby potentially reducing reliance on more expensive paratransit options.

Nevertheless, public data on the amenities available at a given bus stop are largely unavailable. Most U.S. transit agencies do not maintain or release information of bus stop amenities [5, 6], mainly due to the lack of manpower and funds to conduct manual inspection for a large number of bus stops. For some transit agencies that have developed such a dataset, their lists generally cover a very limited set of amenity types (e.g., shelter and seating) and in some cases, and often include a significant amount of inaccurate or missing data due to infrequent updates. Although the General Transit Feed Specification (GTFS) data are gaining more popularity with detailed and comprehensive bus stop information, bus stop level amenities are rarely covered. This presents a missed opportunity for research projects in which obtaining objective measures of riders' experiences is important. Moreover, the lack of bus stop level amenity conditions along transit networks prevents transit agencies and local jurisdictions from making informed decisions on prioritizing and implementing enhancements.

The availability of Google Street View (GSV) images and recent advances in computer vision (CV) technologies (e.g., objective detection) offer a promising approach for developing a comprehensive and up-to-date database of bus stop amenities. GSV is a widely available and frequently updated data source, and if coupled with computer vision algorithms it can achieve accurate detection of bus stop amenity features. It can facilitate large-scale bus stop assessment without the need for labor-intensive manual inspections. To realize this potential requires one to tackle two key technical challenges: one is to efficiently extract the amenities present at each stop from GSV images, and the other is to train a computer vision model that can detect various bus stop amenities with high accuracy and precision.

To address these challenges, we develop the Transit Amenities Assessment System (TAAS), which offers an automated, low-cost, and generalizable approach that uses GSV and computer vision for bus stop assessment. Transit agencies with GSV image coverage can apply TAAS to efficiently create detailed bus stop amenity datasets for their service areas. Considering the potential GTFS data quality issues (e.g., the bus stop coordinates are not precise) and the high cost for automatic extraction of GSV images through Google API, we have introduced a dynamic prediction process in TAAS. As we will discuss in detail below, the dynamic parameter adjustment mechanism incorporated in the system not only enhances feature detection accuracy but also ensures efficient use of computational resources.

This paper is organized as follows: we first describe the research background and review relevant literature on bus stop amenities assessment. Then, we introduce TAAS, describing the system framework, data needs, as well as its workflow that includes a transfer learning process, a dynamic prediction mechanism, and model evaluation and testing. To demonstrate the practical deployment of TAAS, we apply it to assess transit stop amenities (focusing on shelters and benches) in five major Florida cities. In addition, we have conducted a series of scalability and transferability tests to evaluate the applicability of our proposed approach across various deployment contexts and study areas.. We conclude the paper by discussing the results and present potential implications for future research and practice.

2. Research Background

2.1. Assessment of Bus Stop Amenities

Two notable efforts have been conducted in recent years to develop inventories of bus stop amenities. One is the Operation Bus Stop Census initiative led by MARTA Army (a citizen-led advocacy organization in Atlanta) that accomplished the collection of data on available amenities at over 3,200 bus stops (approximately one-third of all MARTA bus stops) with the help of over 300 volunteer surveyors [7]. The data collection phase lasted almost one year and was conducted both in-person and remotely via GSV. The other is the Bus Stop Census of San Francisco conducted by Marcel Moran, who inspected all 2,964 bus stops in San Francisco by bike in a three-month period [5]. While these efforts are laudable, the data collection approaches are too labor-intensive to be scalable to multiple cities, regions, or across the country. Also, these approaches do not offer a convenient or efficient way to update the bus stop census database in real time.

2.2. Application of GSV and Deep Learning to Transportation Infrastructure Assessment

The availability of GSV images and recent advances in computer vision technologies (e.g., objective detection) offer a promising approach to enable the development of a national bus stop census database. Previous studies have demonstrated the validity of using GSV images and for detecting various transportation infrastructure elements such as traffic signs [8], signalized intersections [9], bikeways [10], marked crosswalks [11], sidewalks [12], walkability [13], and bikeability [14]. In terms of bus stop amenities, some recent studies have applied AI methods to identify shelter, seating or signage from street-view images [15, 6, 16].

Despite the comprehensive coverage of types of infrastructure or amenities, research gaps remain in deploying the computer vision models at scale and then using the results to inform practice. First, the previous models rely on a limited sample of images, often from a single region, limiting its transferabililty to other cities. Second, they rely heavily on manual annotation to create training labels. This, combined with the cost of high-Resolution GSV API access, has made the existing automated bus stop assessment approaches only marginally more efficient than manual audits. This study aims to advance the automated bus stop assessment method by exploring the potential of leveraging AI-powered image recognition for efficiently and accurately identifying a variety of bus stop amenities, especially shelters and benches, across different geographical contexts.

3. The Transit Amenities Assessment System

3.1. System Framework

To address the research gaps discussed above, we introduce the *Transit Amenities Assessment System* (TAAS), an *automated*, *low-cost*, and *generalizable* system to assess various bus stop amenities. Figure 1 shows the workflow of the framework, encapsulating the stages of data collection, model training, dynamic site identification, and the ensuing accuracy analysis. The TAAS system includes three main components:

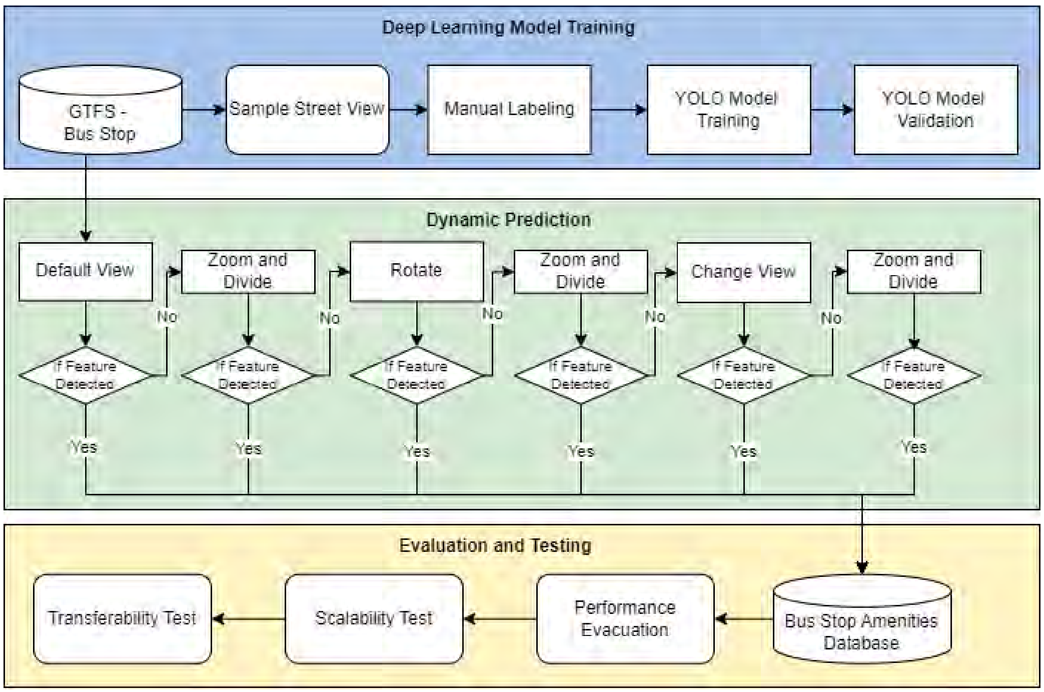

Figure 1: The Transit Amenities Assessment System pipeline.

First, using deep learning techniques, the system leverages the state-of-the-art YOLO (You Only Look Once) model (YOLOv8) to achieve the object detection function [17]. We implement transfer learning to enhance detection accuracy and efficiency, allowing the model to leverage pre-trained weights and adapt to new tasks with less data and computational cost. This approach has been shown to improve performance in various transportation and infrastructure applications [18, 19, 20].

Second, we introduce a dynamic prediction process to address three prevailing issues that cannot be addressed by the deep learning model itself when deploying TAAS for automated bus stop assessment. This process handles one image and one bus stop at a time, starting with the assumption of a perfect situation with no technical challenges encountered. Initial parameters are calculated and the corresponding image is analyzed. If no target is detected, the process adjusts parameters to account for another specific challenge of the current bus stop until all challenges have been addressed or the target is detected.

Finally, users of TAAS can validate the results inferred by the dynamic prediction process with the ground truth data and use different performance metrics to quantitatively assess the model performance. Meanwhile, during the dynamic identification process, we have introduced a mechanism to measure the API parameters for each step (e.g., Zoom and Divide, Rotate, Change View) to improve the results of feature detection at each bus stop. Google charges a fee for each API call, which constitutes the primary expense of using TAAS. The system can provide a high-fidelity break-down of the API usage and the cost-efficiency in each step. Based on the API usage measurement, one can determine how to balance object detection accuracy and project costs when using the TAAS. We have also designed a series of transferability test that can be used for assessing the system's ability to apply to other contexts.

3.2. Data

Our proposed approach primarily use two publicly available datasets: GTFS and GSV data. The GTFS data is the *de facto* standard format to transmit and broadcast transit service supply and schedule data, which consists of several relational database tables that detail the transit system's stops, trips, routes, arrival and departure times, and other schedule-related information. Here we primarily extract from the GTFS data information regarding bus stops, which includes the bus stop coordinates and stop ID. For the GSV data, we use the official GSV application programming interface (API) to fetch static street view images as our primary dataset. This low-cost, high-capacity API allows researchers to capture non-interactive, medium-resolution images from the panorama database. It is noteworthy that the GSV API returns static images, not the interactive images within a web-based interface. Each street view image is returned via an HTTP request specified by various parameters, including the coordinates of the panorama, field of view (FoV, the zoom level of the image), pitch (up or down angle of the camera), and heading (left and right angle of the camera).

3.3. Detailed Workflow of the Transit Amenities Assessment System

As shown in Figure 1, the TAAS includes three key steps: deep learning model training, dynamic prediction, and model evaluation and testing. To illustrate the workflow in the following subsections, we will focus on describing the detection of bus shelters and benches to exemplify the effectiveness of the system for detecting bus stop amenities. The system can be used to detect other features (e.g., signage, boarding pads, accessibility features) as needed.

3.3.1. Deep Learning Model Training and Validation

This step starts with retrieving corresponding street view images from Google Maps according to the coordinates of transit stops. For stops with clear, complete images, we directly downloaded images using the GSV API and included them in the dataset. For images that bus stops that were not captured via the API or the automatic requested image not fully covering the amenities, we manually adjusted and captured the images on the Google Street Map website before adding them to the dataset. As we obtain these images from different sources, the collection of original images can have different resolutions and potential dataset pollution. To address this issue, we standardize all training images by resizing the images manually captured from Google Street View. We then enhance the resolution of all images by using OpenCV's super-resolution model to ensure similar adjacent pixel patterns across all images, optimizing the training effect of the model. This same enhancement technique will also be applied during prediction, ensuring matched adjacent pixel distribution patterns between the training set and the prediction set.

To prepare the training set, we choose Roboflow, a tool known for its fast annotation and efficient preprocessing [21, 22], and its compatibility with YOLOv8, for manual labeling training, validation, and testing samples. We save the annotations in YOLO format including the class and bounding box coordinates. Then we download the pre-trained YOLOv8 weights for object detection (yolov8l.pt) trained on the Common Objects in Context (COCO) dataset as the initial weight for transfer learning [23]. These pre-trained weights are good at handling complex environments and detecting multiple classes of objects, and they require moderate computing and storing ability. We use 100 epochs for training, which is an empirically chosen value, to ensure the model learns the data features adequately without over-fitting, providing a balance between sufficient training and practical training time.

3.3.2. Dynamic Prediction and Parameter Adjustment

While it is intuitive to apply deep learning model to the GSV images collected from Google API, there are multiple drawbacks of the model and the API, which limit our ability to automate the collection of street view images and generate high-fidelity predictions. Specifically, the technical challenges are as follows:

Challenge 1: Low Image Resolution. Despite GSV static API's low costs and high capacity, it has a maximum image size of 640x640 pixels, which poses a significant constraint for acquiring high-resolution panoramic views. On one hand, the returned images can be too blurry when the observation point is far from the bus stop and a large FoV is used, as shown in figure 2 A. On the other hand, a too close-up photo may not capture the whole picture of the bus stops and miss some amenities. To address this issue, we need to optimize the parameter settings to find a best position to capture the complete view of bus stops and amenities without compromising the resolution.

Challenge 2: Coordinate Discrepancy. The reported bus stop location in the GTFS data may be inaccurate due to GPS errors or data misalignment. Therefore, calculating heading and FoV with simple triangle math leads to potential discrepancies in results. These errors include wrong camera heading angles (Figure 2 B) and too far-away or close-up view; in some extreme cases, the returned images may be from the interior of a building or a totally different street across the intersection corner. Meanwhile, in the GTFS data, a single pair of latitude and longitude coordinates is used to denote the location of a bus stop. But different amenities, especially for bus stop sign, shelter and bench, can naturally have different position. This physical separation means that using one single coordinate point to represent all elements is fundamentally flawed. Hence, attempting to detect all these elements with the GTFS data may results in missing one or more targeted facilities in the retrieved images.

Challenge 3: Object Obstructions. Given that GSV primarily uses cars (i.e., specialized panoramic vehicles) for image capture, it is inevitable to encounter obstructions between the view point and the targeted bus stop as illustrated in Figure 2 C and D. These obstructions may include vehicles, trees, other structures and glare when facing the sun. Notably, vehicles are the frequent type of obstructions, and they may become more challenging to avoid when GSV Car and the obstruction vehicles move in the same direction and continuously obstruct the bus stop in multiple panoramic images along the road.

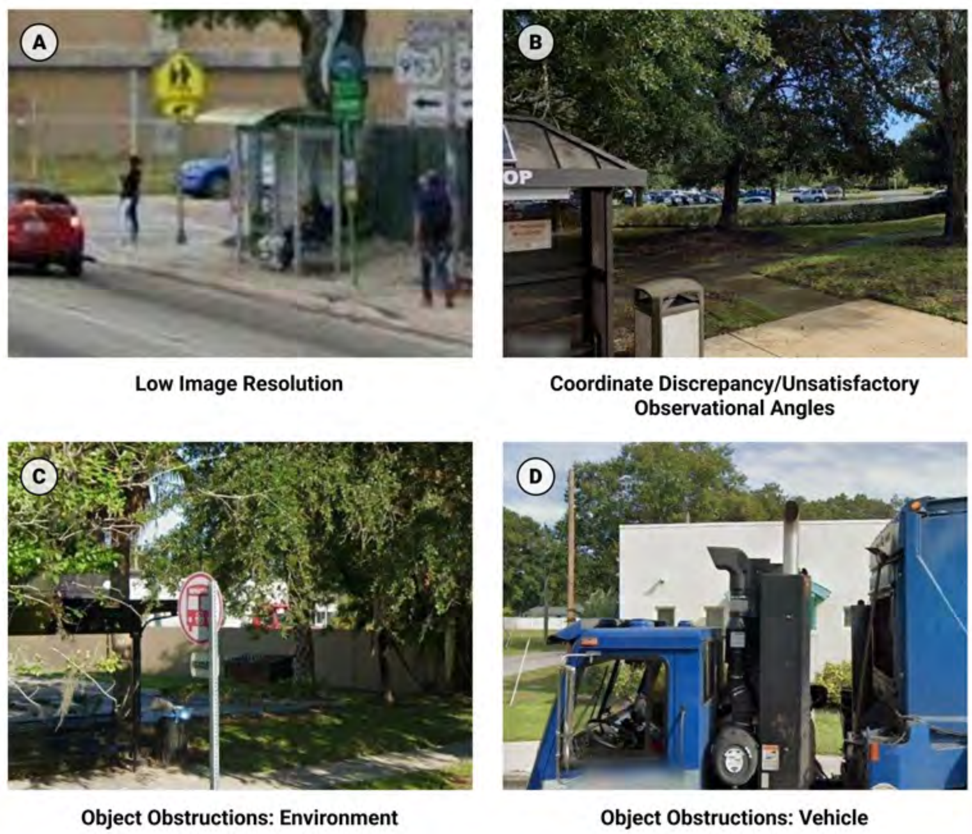

Figure 2: Illustrations of technical challenges.

To fully automate the inference process, we develop a pipeline to dynamicly adjust the camera parameters and engage in real-time prediction. This approach aims to emulate and automate the manual adjustment process used to rectify ground truth errors, thereby extending its applicability to subsequent research endeavors.

Step 1: Naive Prediction. In Step 1, we assume that the stop has not encountered any technical challenges and directly predict based on the original image. We use the GSV API to obtain the panoramic ID (panoID) for observing the city bus stop from the specified location. We adjust FoV based on the distance from the panoID location to the target location and download the street view image. Next, we enhance the image resolution by a factor of 2 using OpenCV's super-resolution model to achieve better details of the feature. We then predict the features using the trained YOLOv8 model. If a feature is detected, we save the prediction result and proceed to the next bus stop. This step requires that the feature is completely captured in the image and is close enough to the observation position to have a good resolution. If no feature is detected, we assume it is either because the shelter truly does not exist, or it was not detected due to other technical challenges, and move on to step 2. This rule requires very low false-positive rate of the YOLO model. We later empirically calculate the false-positive rate and overall accuracy of our trained deep learning model to confirm it satisfies this assumption in the analysis section.

Step 2: Divide and Zoom In. In Step 2, we assume that the stop's location is nearly or completely accurate, with potential false-negative results caused by the issue of low image resolution (Challenge 1). Under this assumption, we split the image into left and right sections and zoom in by adjusting the orientation and FoV without losing any angles. For the left section, the heading is adjusted by subtracting a quarter of the FoV and then taking the result modulo 360 degrees. For the right section, the heading is adjusted by adding a quarter of the FoV and then taking the result modulo 360 degrees. The new FoV for both sections is set to half of the original FoV. We then download the zoomed-in images and identify them separately. Figures 3 A and B demonstrate the process of this step. If the target is not detected in Step 2, we continue to address other challenges and move on to Step 3.

Step 3: Rotate. In Step 3, we assume a significant deviation between the target and actual locations, with potential false-negative results caused by the issue of coordinate discrepancy (Challenge 2). Therefore, this makes the distance and FoV adjustment in Step 2 moot. To address this, we keep the original observation point and adjust the viewing angle. We repeat the operations from Step 1 and Step 2 until a shelter is detected or all reasonable observation points are covered. Notice that we also execute the divide and zoom-in operation after rotating since the new images would still have the low-resolution issue. Figure 3 C shows an original view of a stop with a slight deviation from the actual facilities. Figure 3 D exemplifies a solution by adjusting the viewing angle.

Step 4: Change Viewpoints. If no target is found in Step 3, we assume

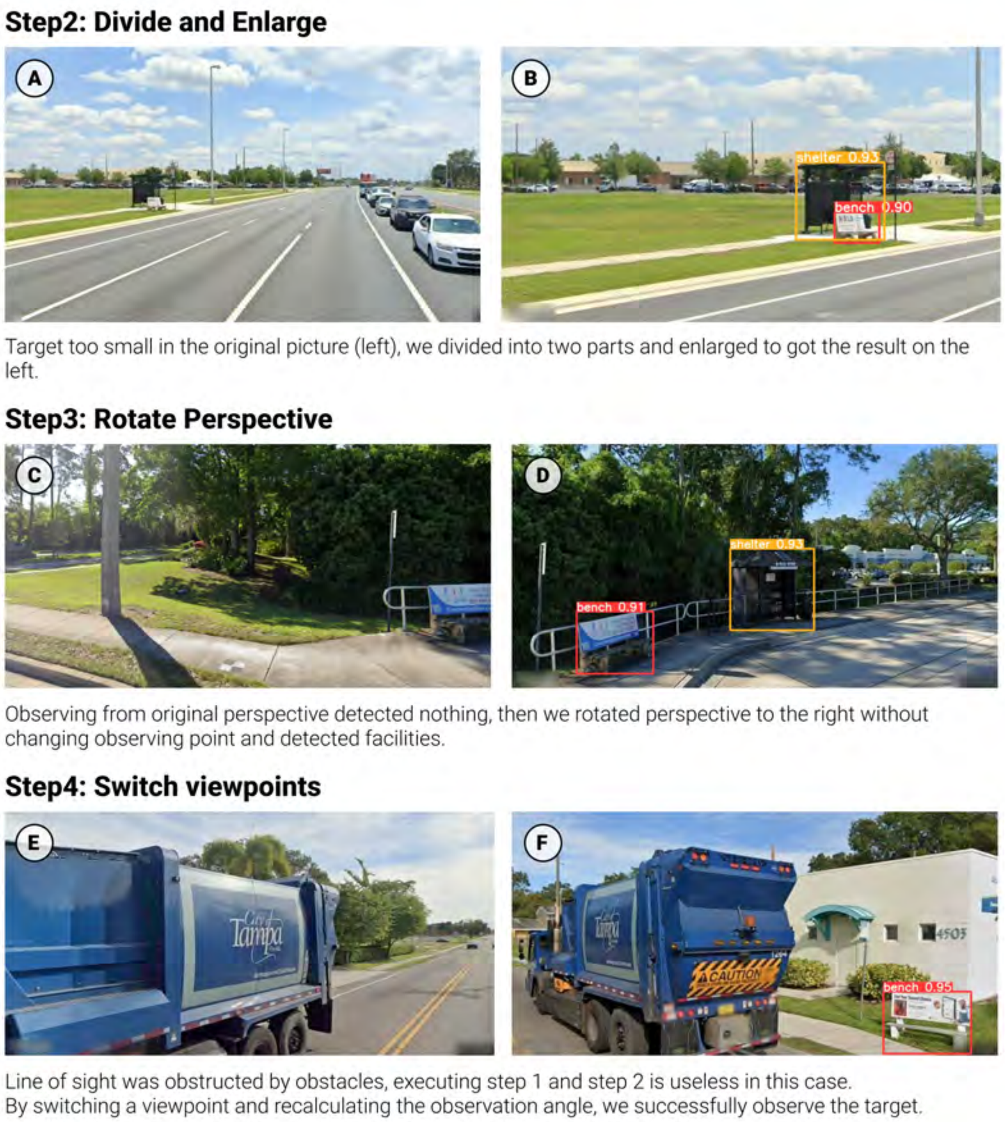

Figure 3: Illustration of technical approaches

the target coordinates are nearly correct but obstructed, with false-negative result caused by objective obstructions (Challenge 3). To tackle this issue, we move the observation point to four adjacent locations in the north, south, east, and west directions. We select locations ten meters away from panoID location, an empirical value that allows us to obtain a new panorama from GSV without moving too far in the cardinal directions of the target. This provides different observation points from GSV cars' panoramic shots. After changing the observation point, we calculate the new angle and distance to the target, download the image, and repeat the operations from Step 1 to Step 2. Notice that we also execute divide and zoom-in operation after rotating since the new images would still have the low-resolution issue. Additionally, this step serves an implicit purpose. While the previous operations focused on changing the observer's perspective to locate the shelter, this step gathers multiple angles of observation for the shelter. This approach compensates for the image recognition model's potential deficiencies in identifying shelters from certain angles, thereby further enhancing the model's recognition capability. If a shelter is still not identified after executing Step 4, we finally conclude that the target bus stop lacks a shelter and continue to the next bus stop in the loop.

3.3.3. Evaluation and Testing

Before applying the models developed from the previous two steps to detect features at all transit stops and construct a comprehensive bus stop amenities database, one should evaluate the model performances with measures such as prediction accuracy, precision, and mAP50 (mean average precision calculated at an intersection over union threshold of 0.50). If the project has budget constraints, the analyst is advised to also perform a scalability test as the TAAS deployment process involves the use of GoogleMaps API, which can occur some costs depending on the usage amount. Finally, if models trained from a TAAS deployment are intended for use in new study areas not included in the training set, we recommend conducting transferability tests before such use.

4. Application of the Transit Amenities Assessment System in Florida

To illustrate how TAAS can be deployed in practice, we apply it to assess transit stop amenities (focusing on shelter and bench) in five Floridian cities and their transit systems: Miami (Miami-Dade Transit), Orlando (LYNX), Tampa (Hillsborough Area Regional Transit Authority), Jacksonville (Jacksonville Transportation Authority), and Gainesville (Regional Transit Authority).

4.1. Model Development

Specifically, we collect the most recent version of GTFS data from these cities and train a deep learning model for detecting amenities at each bus stop. We employ the YOLOv8 model for transfer learning on bus stop sample images that we manually picked and labeled from the five Florida cities. Our training set has 1,140 bus stops in total, including 380 bus stops in Gainesville and 190 bus stops in the other four Floridian cities, respectively. Our validation and test sets have 125 and 510 bus stops, respectively. In addition, we diversify the samples based on different surrounding environment and land use types. These measures can strategically prevent the inflation of test results due to excessive similarity between the training and testing samples in terms of traffic and urban infrastructure.

We have also obtained a dataset from the City of Gainesville that contains a comprehensive list of the bus stop amenities, which can serve as the ground-truth data for model validation. In addition, to test the transferability of the system, we used the transit amenities inventory database developed by Marcel Moran for the city of San Francisco from to test the performance of the trained YOLO model when applied to a different study area [5]. The results will be discussed in a separate section below.

4.2. Evaluation of Model Performance

We validate the YOLO model with a validation set of 125 bus stops and 452 images. For bus shelters, the precision, recall, and mAP50 of the trained YOLO transfer learning is 0.965, 0.878, and 0.918, respectively; for benches, the three measures are 0.945, 0.862, and 0.905 for bench, respectively. This shows our deep learning is well trained for the purpose of detecting shelters and benches. However, it is noteworthy that the performance of the YOLO model is different from the performance of the system due to the technical challenges discussed above.

To test the inference outcomes of the whole system, we calculate the accuracy and precision of the system on the test set as well as the false-positive rate and API usage. We create the ground-truth data from the official amenities inventory and our manually annotated stop statistics. To test the generazability of our system to diverse environments with different urban forms and distinctive architectural styles of bus stop facilities, we undertook detailed validations for the five Florida cities, i.e., Miami, Jacksonville, Orlando, Gainesville and Tampa, as shown in Table 1.

Table 1: Result Evaluation

City	Samples	Shelter		Bench		API Usage
		Accuracy	Precision	Accuracy	Precision	
Total	650	0.963	0.941	0.845	0.901	10243
Orlando	140	1.000	1.000	0.836	0.906	2278
Gainesville	140	0.950	1.000	0.814	0.941	2263
Jacksonville	140	0.979	0.923	0.85	0.885	2246
Miami	140	0.921	0.885	0.886	0.867	1991
Tampa	90	0.967	0.869	0.834	0.912	1465

The system achieves state-of-the-art predictive performance compared with prior study [6, 15] on a large testing sample. Our model exhibits robust performance in shelter predictions across all the four listed cities with consistently high accuracy (0.963) and precision (0.941) as well as a low False Positive Rate (FPR, 1.73%). Among the five cities, Orlando shows the best results and Miami has the worst performance. The differences in model performance can be partially attributable to their built environment characteristics, with Miami having a lower performance for having diverse land use patterns, sophisticated infrastructure systems, and rich environmental factors.Bench detection achieved satisfactory results with an overall accuracy of 0.845 and an overall precision of 0.901, although its performance was slightly lower than that of the shelter. This exhibits that the system can achieve high reliability and fidelity for multiple types of bus amenities.

Through Naive Prediction (Divide and Zoom in if needed), only 25 shelters were detected from the 650 sampled transit stops in the five cities. Through Rotate (Divide and Zoom in if needed), 44 cases were detected. Through Change Viewpoints (Divide and Zoom in if needed), 113 cases were detected. There were 468 cases where shelters were concluded to be absent. This result directly justifies the necessity of our system. Only 13.73% of shelters could be identified using naive detection and zooming in. The remaining 86.27% required other methods provided by the system, indicating that the majority of the bus stop location data contains inaccuracies or faces the challenges we mentioned earlier.

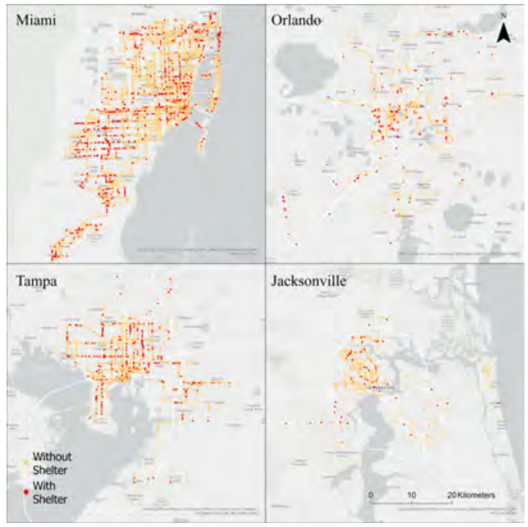

Figure 4: Spatial Patterns of Bus Shelters with and without Shelters

4.3. Deploying the Trained Model to Assess All Bus Stops in Florida Cities

We further run our system to perform an assessment of all bus stops in Miami, Orlando, Tampa, and Jacksonville. We did not run it in Gainesville because, as discussed above, a bus stop amenity inventory dataset already exits in Gainesville.

Figures 4 and 6 visualize the location of bus stops with and without shelter and bench for Miami, Orlando, Tampa, and Jacksonville. The city centers generally witness more bus stops with shelters, which is consistent with findings from prior studies [5, 24]. However, the spatial patterns of the rate of bus stops with shelters is quite different from prior conclusions: Figure 5 shows that urban centers of Orlando, Tampa, and Jacksonville have lower rate of shelters, which can be due to higher number of bus stops in those areas. For Miami especially, some urban outskirts like Kendall and Sunset (southwestern Miami) have much higher rate of shelters. Compared with shelters, benches are more available in all four cities, possibly due to lower installation cost. The spatial distribution of benches is also less clustered than bus shelters.

Meanwhile, we witness major disparities among different cities in terms of stop amenities availability. Table 2 shows the number and percentage of bus stops with shelters and bench. As the city with most extensive and frequent transit service, Miami have the most shelters, followed by Tampa. On the

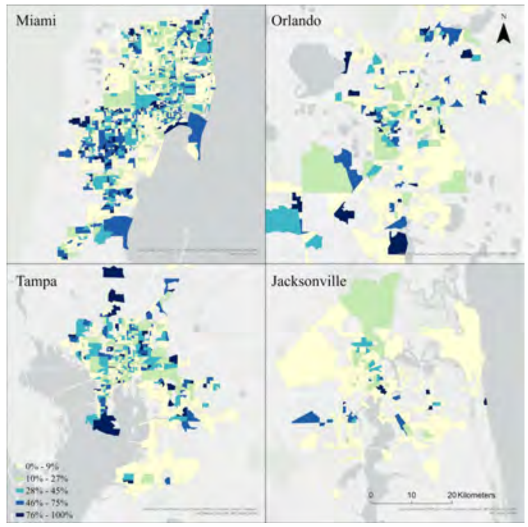

Figure 5: Spatial Patterns of Rate of Bus Stops with Shelters in Census Block Groups for Four Cities

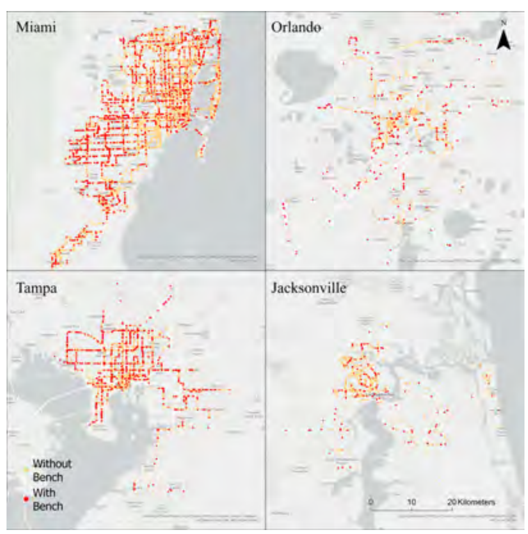

Figure 6: Spatial Patterns of Bus Shelters with and without Bench

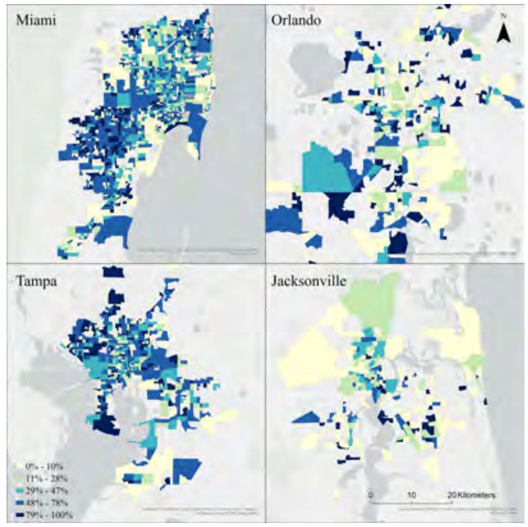

Figure 7: Spatial Patterns of Rate of Bus Stops with Shelters in Census Block Groups for Four Cities

other hand, Orlando and Jacksonville, two highly car-dependent cities, only have 14.86% and 9.10% of bus stops with shelters, respectively. Meanwhile, the city with highest benches is Tampa with almost half of the bus stops with benches. However, highly car-dependent cities like Orlando and Jacksonville still have lower percentage of bus stops with benches.

Table 2: Number and Percentage of Stops with Shelters and Bench in Four Floridian Cities

City	Stops Count			Shelter Rate	Bench Rate
	Total	with Shelter	with Bench		
Miami	8097	1797	3303	22.19%	40.79%
Tampa	2327	492	1118	21.14%	48.04%
Orlando	1460	217	422	14.86%	28.90%
Jacksonville	1033	94	234	9.10%	22.65%

In conclusion, the majority of bus stops in all four cities lack infrastructure to shield public transit users from extreme heat and provide places to rest when waiting, which could incur major negative implications on users' experience and health [25, 24]. This effect may be even more significant considering the cities with less bus shelters also have lower service frequency and longer waiting time [26].

5. Scalability and Transferability of the Transit Amenities Assessment System

The analyses above exemplify the effectiveness of our methods to assess the bus stop amenities with high fidelity and precision across four Florida cities. As the system is intended to be applicable to a wide spectrum of scenarios and urban contexts, we further conduct a series of scalability and transferability tests to evaluate the applicability of the our proposed methods in various deployment contexts and study areas.

5.1. Balancing Prediction Accuracy and API Usage for Improved Scalability

Unlike many prior studies that relied on existing training images captured by humans [6, 9], our system uses a real-time API to dynamically capture the location and details of the features. As of July 2024, the cost of requesting one static GSV image is $0.007 per image. An extensive bus system can have several thousands of bus stops, and the inference of one bus stop can incur multiple API requests since the system would depend on iteration of inference process. Therefore, API usage becomes a major factor as it determines the monetary cost of the system deployment, which can have critical implications for the implementation of the framework.

Therefore, we first conduct a detailed API usage analysis of the system presented above. Considering all five cities, Step 1 and Step 2 cost up to 3 API usages, Step 3 costs up to 6 API usages, and Step 4 costs up to 12 API usages. This shows that the high performance of the system comes at the expense of increased API usage. On average, one bus stop requires about 15.8 static images to achieve the best performance, which is equivalent to $0.11.

To reduce API expenses, we present a cost-lite version of TAAS designed to balance prediction accuracy and API usage, thereby enhancing the system's scalability. As noted in the model performance section, Step 4 identified shelters in 113 cases, while no shelter was detected in 468 cases. This indicates that 89.3% of the shelter detection results are produced from Step 4, with 80.5% of these results concluding that the bus stop is not equipped with shelters. Since most results conclude that there is no shelter, reducing the precision of Step 4 would not significantly impact the overall accuracy of the system. Given that reaching a "no shelter detected" conclusion requires executing all proposed solutions and completing all steps, Step 4 accounts for the highest API usage and is therefore the most expensive phase of the process. Consequently, we aim to reduce costs specifically in Step 4, where we performed zoom operations from four observation angles to mitigate the impact of our assumptions on system performance. In the cost-lite version of TAAS deployment, we adopt a more aggressive strategy regarding the accuracy of the coordinates, allowing us to reduce the cost during the divide and zoom-in operations. Specifically, we assume that the target coordinates are accurate enough to calculate the distance from the observation point to the target and to set the GSV API parameters. While this approach reduces fault tolerance for limited FoV and potential coordinate deviations, it significantly cuts API usage in Step 4 by two-thirds.

Table 3 demonstrates the potential for a widespread application of TAAS's cost-lite version. We find that the cost-lite version can achieve 33.2% lower API usage with the cost of accuracy drops of 0.017 and precision drops of 0.012. One bus stop costs about 10.5 images for the cost-lite version, which is equivalent to $0.07. This trade-off significantly improves the cost efficiency of the system with a small price in performance as a much more practical and economic approach to apply our assessment system to a new context. However, it is also noteworthy that despite the relatively higher cost of the full version system, our system is still much cheaper than previous approaches in that it saves the costs for manual annotation and high-resolution image storage.

Table 3: Trade-off Lite Version

City	Samples	Accuracy (Shelter)	Precision (Shelter)	API Usage
Total	650	0.946	0.929	6837
Orlando	140	0.972	0.956	1530
Gainesville	140	0.936	0.960	1512
Jacksonville	140	0.971	0.957	1490
Miami	140	0.9	0.863	1345
Tampa	90	0.956	0.9	960

5.2. Transferability Tests

While the fine-tuned system introduced above achieves both high accuracy and high precision in Floridian cities, another crucial question for future deployment and application is its transferability to other cities not included in model training. This is especially important for high-density urban areas with very diverse and heterogeneous urban settings and facilities.

To answer this question, we choose San Francisco, one of most transit-oriented cities in the US with intensive transit services and diverse land use patterns, as the test site. We conduct a sensitivity analysis on the size of the local training set, i.e., how many local street view images we add to the training set, and calculate the accuracy and precision of each model. For example, with 0 local images, we essentially apply the system trained with Floridian cities to San Francisco; with 200 local images, we download 200 images of randomly selected San Francisco stops via GSV API and train the system with 200 stops from San Francisco and 510 stops in Floridian cities. For each test, we train the YOLO model from the default weight until convergence. For the testing set, we randomly choose 300 bus stops from San Francisco Bus Stop Census Open Data and collect generate ground truth and calculate the accuracy and precision.

Figure 8 shows how accuracy and precision vary with the size of local training set. Both accuracy and precision would peak at the training set size of 150, while more images do not help increase the performance. This test provides firsthand evidence for future application of the system: The above results suggest that if analysts want to transfer TAAS to a new study area, they can achieve satisfactory prediction outcomes by adding approximately 150 images from another city to the existing training set. Compared with the prior model [6], which primarily focused on small- and medium-sized cities, our approach demonstrates that we can achieve equally high object detection capabilities in large and complex urban environments. This highlights the robustness and transferability of our system.

6. Discussion and Conclusion

Bus stops serve as the initial gateways to transit services and opportunities [1]. However, public data on the amenities such as shelters and benches available at a given bus stop are largely unavailable. To address this data gap, this study presents an automated, low-cost, and generalizable system

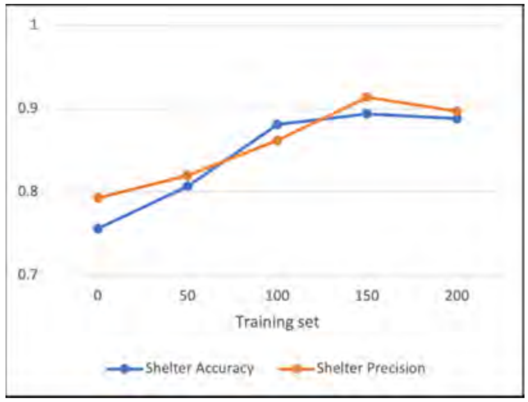

Figure 8: Result Evaluation

for stop amenities assessment. By leveraging the YOLO model for object detection and transfer learning, our system introduce an automated, dynamic prediction algorithm to adjust the API parameters and predict the amenities in an adaptive, real-time manner. The system not only achieves state-of-the-art accuracy but also reduces the difficulty and time required for model training.

The system is highly scalable and transferable. Considering the potential implications of cost, we present a cost-lite version of the system to minimize the API usage without significantly compromising the performance, which enhances the scalability of the system and its transferability to other cities. We also present a sensitivity analysis to assess the size of local training set needed when transferring the system to a different city. Our results show that the system would achieve satisfactory performance at a cost of collecting 150 local GSV images and adding them to the training set. These analyses provide empirical evidence that supports large-scale applications of the system in different scenarios with low manual labor and monetary costs.

The application of the system has significant practical implications for future transit planning and administration. First, the results reveal very heterogeneous spatial patterns both inside each city and across different cities. With more frequent transit services and higher demand, Miami and Tampa's stop amenities are significantly better than Orlando and Jacksonville, which are highly car-dependent. Meanwhile, with lower cost of installation and maintenance, benches are also much more available than shelters. Second, our system can dramatically reduce the time and cost associated with manual assessments, providing reliable data to improve bus stop facilities and enhancing the overall understanding of public transportation infrastructure. This, in turn, enhances the quality of public transportation services and increases user satisfaction.

In sum, the system represents a significant advancement in the field of transportation infrastructure assessment, offering a scalable and efficient solution to a traditionally labor-intensive process. With continued development, it has the potential to become an indispensable tool for urban transportation planning.

Our study has several limitations. First, the dataset, while comprehensive, may not cover all possible variations of bus stop environments. In areas with dense bus stops, the system may misidentify a different bus stop due to close proximity. Additionally, regions like internal community roads or school internal roads often lack street view images, making them inaccessible for our system. The performance of the YOLO model might also degrade in complex or cluttered scenes. Second, despite infrequent upgrading rate of bus amenities, the construction and styles of shelters can be continuously evolving, while our training set is based on existing data and is not fully up-to-date. This may result in lag behind urban development and construction.

Future research can address these limitations by expanding the dataset to include a wider range of bus stop types and conditions. Exploring other advanced object detection models could further improve accuracy. Introducing more rigorous mathematical calculations to replace current empirical values could enhance the precision of the system. Moreover, integrating real-time monitoring data could enable continuous and dynamic assessment of bus stop amenities.

7. Acknowledgments

We are grateful for the funding support from the Center for Equitable Transit-Oriented Communities Tier-1 University Transportation Center (Grant No. 69A3552348337).

References

[1] R. Ewing, Asking transit users about transit-oriented design, Transportation Research Record 1735 (1) (2000) 19–24. arXiv:https://doi.org/10.3141/1735-03, doi:10.3141/1735-03.
URL `https://doi.org/10.3141/1735-03`

[2] J. Y. Kim, K. Bartholomew, R. Ewing, Another one rides the bus? the connections between bus stop amenities, bus ridership, and ada paratransit demand, Transportation Research Part A: Policy and Practice 135 (2020) 280–288. doi:https://doi.org/10.1016/j.tra.2020.03.019.
URL https://www.sciencedirect.com/science/article/pii/S0965856418314198

[3] M. R. Talbott, Bus stop amenities and their relationship with ridership: a transportation equity approach, The University of North Carolina at Greensboro, 2011.

[4] J. N. Balog, Guidebook for attracting paratransit patrons to fixed-route services, Vol. 24, Transportation Research Board, 1997.

[5] M. E. Moran, Are shelters in place? mapping the distribution of transit amenities via a bus-stop census of san francisco, Journal of Public Transportation 24 (2022) 100023. doi:https://doi.org/10.1016/j.jpubtr.2022.100023.
URL https://www.sciencedirect.com/science/article/pii/S1077291X22000236

[6] J. Kim, J. Park, J. Lee, K. M. Jang, Examining the socio-spatial patterns of bus shelters with deep learning analysis of street-view images: A case study of 20 cities in the u.s., Cities 148 (2024) 104852. doi:https://doi.org/10.1016/j.cities.2024.104852.
URL https://www.sciencedirect.com/science/article/pii/S0264275124000660

[7] MARTA Army,

[8] A. Campbell, A. Both, Q. C. Sun, Detecting and mapping traffic signs from google street view images using deep learning and gis, Computers, Environment and Urban Systems 77 (2019) 101350. doi:https://doi.org/10.1016/j.compenvurbsys.2019.101350.
URL https://www.sciencedirect.com/science/article/pii/S0198971519300870

[9] X. Li, H. Ning, X. Huang, B. Dadashova, Y. Kang, A. Ma, Urban infrastructure audit: an effective protocol to digitize signalized intersections by mining street view images, Cartography and Geographic Information Science 49 (1) (2022) 32–49, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15230406.2021.1992299. doi:10.1080/15230406.2021.1992299.
URL https://doi.org/10.1080/15230406.2021.1992299

[10] X. Ding, H. Fan, J. Gong, Towards generating network of bikeways from mapillary data, Computers, Environment and Urban Systems 88 (2021) 101632. doi:https://doi.org/10.1016/j.compenvurbsys.2021.101632.
URL https://www.sciencedirect.com/science/article/pii/S0198971521000399

[11] M. Li, H. Sheng, J. Irvin, H. Chung, A. Ying, T. Sun, A. Y. Ng, D. A. Rodriguez, Marked crosswalks in US transit-oriented station areas, 2007–2020: A computer vision approach using street view imagery, Environment and Planning B: Urban Analytics and City SciencePublisher: SAGE PublicationsSage UK: London, England (Jul. 2022). doi:10.1177/23998083221112157.
URL https://journals.sagepub.com/doi/full/10.1177/23998083221112157

[12] H. Ning, Z. Li, C. Wang, M. E. Hodgson, X. Huang, X. Li, Converting street view images to land cover maps for metric mapping: A case study on sidewalk network extraction for the wheelchair users, Computers, Environment and Urban Systems 95 (2022) 101808. doi:https://doi.org/10.1016/j.compenvurbsys.2022.101808.
URL https://www.sciencedirect.com/science/article/pii/S0198971522000527

[13] B. W. Koo, S. Guhathakurta, N. Botchwey, Development and validation of automated microscale walkability audit method, Health Place 73 (2022) 102733. doi:https://doi.org/10.1016/j.healthplace.2021.102733.
URL https://www.sciencedirect.com/science/article/pii/S135382922100229X

[14] K. Ito, F. Biljecki, Assessing bikeability with street view imagery and computer vision, Transportation Research Part C: Emerging Technologies 132 (2021) 103371. doi:https://doi.org/10.1016/j.trc.2021.103371.
URL https://www.sciencedirect.com/science/article/pii/S0968090X21003739

[15] S. M. Shameli, E. Rezazadeh Azar, Computer vision-based generating and updating of the public transit bus stop inventories, Journal of Infrastructure Intelligence and Resilience 1 (2) (2022) 100016. doi:https://doi.org/10.1016/j.iintel.2022.100016.
URL https://www.sciencedirect.com/science/article/pii/S2772991522000160

[16] M. Kulkarni, C. Li, J. J. Ahn, K. O. Y. Ma, Z. Zhang, M. Saugstad, K. Wu, Y. Eisenberg, V. Novack, B. Chamberlain, J. E. Froehlich, BusStopCV: A Real-time AI Assistant for Labeling Bus Stop Accessibility Features in Streetscape Imagery, in: The 25th International ACM

SIGACCESS Conference on Computers and Accessibility, ACM, New York NY USA, 2023, pp. 1–6. doi:10.1145/3597638.3614481.
URL https://dl.acm.org/doi/10.1145/3597638.3614481

[17] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[18] S. Rahman, J. H. Rony, J. Uddin, M. A. Samad, Real-time obstacle detection with yolov8 in a wsn using uav aerial photography, Journal of Imaging 9 (10) (2023) 216.

[19] D. Shokri, C. Larouche, S. Homayouni, A comparative analysis of multi-label deep learning classifiers for real-time vehicle detection to support intelligent transportation systems, Smart Cities 6 (5) (2023) 2982–3004.

[20] Adco, Utility pole detection using yolov8, GitHub Repository (2023).
URL https://github.com/Adco30/Utility-Pole-Detection

[21] R. Team, 5 best image annotation tools in 2023, Roboflow Blog (2023).
URL https://blog.roboflow.com/5-best-image-annotation-tools-in-2023

[22] V. L. Team, 13 best image annotation tools of 2023, V7 Labs Blog (2023).
URL https://www.v7labs.com/blog/best-image-annotation-tools

[23] Ultralytics, Ultralytics yolo docs (2024).
URL http://https://docs.ultralytics.com/models/yolov8/

[24] Q. Miao, E. W. Welch, P. Sriraj, Extreme weather, public transport ridership and moderating effect of bus stop shelters, Journal of Transport Geography 74 (2019) 125–133.

[25] K. Lanza, C. P. Durand, Heat-moderating effects of bus stop shelters and tree shade on public transport ridership, International journal of environmental research and public health 18 (2) (2021) 463.

[26] L. Liu, X. Li, X. Yan, R. H. Pereira, Measuring exposure to extreme heat in public transit systems, Available at SSRN 4869682 (2024).