IET Smart Cities

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Automating City Accessibility Constraints Mapping Through AI-Assisted Scanning of Street View Imagery

Rui S. Moreira[1,2] | Sérgio Moita[1,3] | José Manuel Torres[1,2] | Feliz Gouveia[1,2] | Maria Alzira P. Dinis[1] | Diogo Ferreira[1] | Madalena Araújo[1] | Maria João S. Guerreiro[1]

[1]ISUS LIACC Pole @ Faculty of Science and Technology (FCT), University Fernando Pessoa (UFP), Porto, Portugal | [2]Artificial Intelligence and Computer Science Laboratory (LIACC), University of Porto (UP), Porto, Portugal | [3]Instituto Superior de Engenharia do Porto (ISEP), Instituto Politécnico do Porto (IPP), Porto, Portugal

Correspondence: Rui S. Moreira (rmoreira@ufp.edu.pt)

Received: 21 April 2025 | **Revised:** 18 September 2025 | **Accepted:** 19 November 2025

Handling Editor: Guyue Zhou

Keywords: artificial intelligence | city design | data analytics and machine learning | data structures | governance | planning and policy | smart cities | smart cities applications

ABSTRACT

Urban environments often pose challenges for individuals with mobility impairments due to inadequate pedestrian infrastructure. In addition, the lack of accurate mapping of accessibility features limits the ability to monitor and address these constraints effectively. This paper introduces a framework for Automating City Accessibility Mapping using AI (ACAMAI), that is, provides an AI-assisted pipeline for the automated identification and geolocation of urban accessibility constraints using Google Street View (GSV) panoramas. The ACAMAI pipeline comprises two main stages: (i) training a YOLOv8 object detector to recognise accessibility-related features, such as curb ramps, missing ramps, obstacles and surface problems, in 2D sidewalk images; and (ii) scanning 360° GSV panoramas by extracting multiple perspective views to be analysed by the trained model. The model was trained on a combination of international (Project Sidewalk Dataset—PSD) and local (Porto Dataset—PTD) datasets, achieving high performance across classes, including 91% *recall* and 85% *precision* for curb ramps. In the panorama scanning stage, using a fine angular iterative step (2°) maximised the *recall*, reaching 90% for curb ramps and 93% for obstacles in a locally annotated dataset (GSV Panorama Porto Dataset—GSV-PPD). Although this improved detection coverage, it also led to a high number of redundant predictions, which contributed to a reduced overall *precision*. Finally, identified constraints are georeferenced and mapped onto OpenStreetMap (OSM), supporting scalable and inclusive urban planning.

1 | Introduction

The New Urban Agenda of the United Nations Human Settlements Programme [1] paves the way for the Sustainable Development Goals (SDG), which guide and track urbanisation around the globe. In particular SDG 11 focuses on making cities and human settlements inclusive, safe, resilient and sustainable. Especially for the 15% of the world's population with disabilities, there is a widespread lack of accessibility facilities in built environments (e.g., from private houses and infrastructures, to public buildings and urban spaces). Inaccessible public infrastructures impact the lives of millions of people, preventing them from actively participating in society, decreasing walking rates and increasing the use of even more vehicles. For example, pedestrian accessibility (cf. ramps on sidewalks and obstacle-free pathways) is a major concern for city planners/managers, having a significant impact on the mobility of citizens [2], especially those with motor disabilities, using wheelchairs or

Abbreviations: GSV, Google Street View; OSM, OpenStreetMap; YOLO, You Only Look Once.

pushing children strollers. Urban designers are tasked with planning and redesigning better inclusive places than those we have today [3], requiring more efficient, timely and cost-effective methods for mapping accessibility issues, particularly in urban spaces. In particular, crosswalks play a vital role in urban mobility safety. However, not all crosswalks provide the same facilities for all citizens to cross the streets safely and easily. Several mobility issues, ranging from simpler obstructive elements hindering pedestrian flow to more severe issues regarding the absence of curb ramps or even surface problems, like potholes, further exacerbate accessibility issues and become potential hazards. These situations or accessibility issues should be easily and automatically identified by authorities, through the use of AI-enhanced tools, for timely corrective intervention planning.

Although crowd-sourcing efforts have been used to map sidewalk accessibility in major cities, with interesting results [4, 5], scaling up is time intensive and costly. In addition, several deep learning classification models (e.g., ResNet) have been used to identify sidewalk accessibility, such as curb ramps, missing ramp, sidewalk obstacles and surface problems, although with more simple pipelines.

This paper proposes an alternative two-stage approach process for automating large-scale scanning of Google Street View (GSV) panoramas with the goal of pin-spotting accessibility constraints. The ACAMAI (Automating City Accessibility Mapping using AI) framework introduces an AI-assisted pipeline designed for the automated identification and geolocation of urban accessibility constraints using GSV panoramas. The ACAMAI pipeline, illustrated in Figure 1, comprises two main stages: (i) training a YOLOv8 object detector to recognise accessibility-related features in 2D sidewalk images; and (ii) automating the scanning of 360° GSV panoramas by extracting multiple perspective views for analysis.

The key findings from our comprehensive evaluation of this framework are:

- Optimised Object Detection of Accessibility Constraints: Training a YOLOv8 model on a combined international (Project Sidewalk Dataset—PSD) and local (Porto Dataset—PTD) dataset, allowed achieving high performance across accessibility classes, including 91% recall and 85% precision for curb ramps. This mixed training strategy proved crucial for the model's generalisation across diverse urban contexts and outperformed baseline classification models in realistic street scenes.
- Effective Panoramic Scanning: Using an iterative panoramic scanning process, demonstrated the ability to maximise the detection of accessibility constraints. The finest angular iterative step (2°), achieved 90% recall for curb ramps and 93% for obstacles in a locally annotated dataset (GSV Panorama Porto Dataset—GSV-PPD). This highlights the effectiveness of the framework in comprehensively identifying issues within GSV panoramas.
- Actionable Geospatial Integration: Georeferencing and seamless mapping of identified constraints onto OpenStreetMap (OSM) enriches existing map data with essential accessibility information. This process distinguishes between constraint types and aligns them with relevant OSM nodes. This directly supports scalable and inclusive urban planning, bridging the gap between raw detection and actionable data. It also provides a reusable and extensible tool for assessing urban mobility.

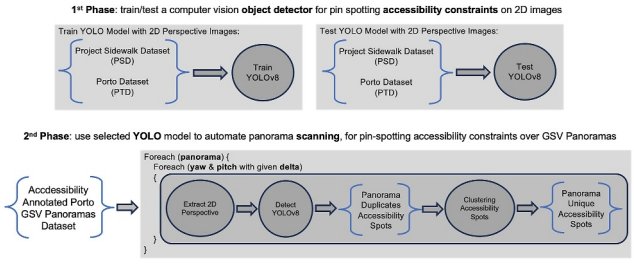

FIGURE 1 | Two-stage process for automating AI-assisted accessibility constraints pin-spotting.

More specifically, the first stage proposes the use of a YOLO object-detection model applied to 2D images of roads and sidewalks, to enable the identification of regions containing accessibility issues of interest (cf. curb ramps, missing ramps, obstacles and surface problems such as potholes) [6]. The YOLO model was: (i) firstly, trained and tested on Project Sidewalk Dataset (PSD); (ii) then tested on our own small Porto Dataset (PTD) and (iii) finally, trained on a combination of both PSD + PTD datasets and tested on PTD. When trained solely on PSD the model does not generalise well to PTD. When trained solely on PTD, mostly due to the small size of PTD, the model performs poorly. However, when trained on PSD + PTD the model performs well on PTD, achieving performances comparable to the first model on PSD. Therefore, introducing a small dataset from Porto significantly improved the performance of the model for this specific region. Afterwards, for a set of GSV annotated panoramas gathered for the Paranhos Parish in Porto, the second stage iteratively extracts 2D perspectives from each panorama and systematically explores all pertinent yaw and pitch values, enabling the model to detect accessibility constraints across the entire scenes. The diagram depicted in Figure 1 visually presents this two-stage workflow, emphasising the sequential nature of model training and deployment for large-scale accessibility mapping.

The remainder of the paper is divided into four sections. First, a background context and related work are provided to emphasise current approaches to automating the detection and mapping of city accessibility constraints. Then, the first stage of our approach is presented, based on the use of a YOLO object-detection model to automate the detection of sidewalk accessibility issues. A comparative evaluation of differently trained YOLO models is presented and compared with a traditional CNN classifier-based approach. Subsequently, the second stage of scanning GSV intra-panoramic images is described, emphasising the need for downsizing the search space in each panorama and also on using clustering to eliminate redundant pin-spots of accessibility constraints. An evaluation and a time profile of the pipeline are also presented. Finally, the paper

concludes with a brief summary of the findings and proposals for future work.

2 | State of the Art on City Accessibility Mapping

Automation of pedestrian accessibility features recognition in urban areas has spurred interest in machine learning techniques. Several deep CNN architectures have been used and their training and testing performance have been analysed to select the best-adapted architectures. These approaches also demonstrated that street view images can be an important source of data for the extraction of accessibility features and for predicting other relevant factors, for city planning and policy-making. In this context, image datasets are typically built from GSV and pre-processed to remove outliers (e.g., images with objects or people occluding parts of the sidewalks) or blurred images.

2.1 | Background on Computer Vision Machine Learning Models

Machine Learning (ML) models applied for image classification, object detection and segmentation tasks are currently the gold standard in Computer Vision (CV) due to the robust results typically reported in the literature. Classification, object detection and segmentation are inherently distinct tasks in the field of computer vision, each addressing different aspects of image understanding. Classification assigns one or more class labels to an entire image without spatial localisation. However, this may be limited when there are two or more objects of interest that can be identified. Therefore, in this paper, we follow an object detection approach and compare its results with the Duan et al. [7] project that uses classification for similar accessibility identification tasks. The latter solution is based on the important ResNet family of deep neural network architectures for image classification [8]. Object detection extends classification by identifying and localising objects using bounding boxes and associated class labels. Segmentation also offers finer granularity, allowing one to assign a class label to each pixel (cf. semantic segmentation) or even distinguishing individual object instances within the same class (cf. instance segmentation), providing both class and shape information. Segmentation is usually valuable for detailed structure analysis, such as in the delineation of skin lesions. More particularly, in urban infrastructure mapping, the works of Hosseini et al. [9] and Hamim et al. [10] also employ deep learning architecture-based semantic segmentation, which is based on a Hierarchical Multi-Scale Attention model by Tao et al. [11]. Both works refer to the integration of HRNet-W48 by Sun et al. [12] and Wang et al. [13] as the backbone and the Object-Contextual Representations (OCR) module by Yuan et al. [14].

In ACAMAI we considered that the use of object-detection could be more effective in contexts with higher variability and occlusion present in street view environments. Our approach is based on You Only Look Once (YOLO) models (cf. Redmon et al. [15] and Jocher et al. [6]), which are state-of-the-art object detection solutions in computer vision. A YOLO object detection pipeline applies a single neural network to the entire image at once. The image is divided into regions and then the system predicts bounding boxes of visual objects and probabilities for each region. Consequently, YOLO enhances the efficiency and speed of object detection when automating the scanning of panorama GSV images. Moreover, it facilitates an approach that mimics human visual processing in scanning 2D rectangular images extracted from GSV panoramas to detect several objects of interest, enabling the identification of multiple accessibility issues on sidewalks.

Recent reviews such as Urban Visual Intelligence highlight the growing role of computer vision in understanding and interpreting urban environments, providing a foundation for AI-driven analysis of streetscapes and public infrastructure [16].

2.2 | Related Work on City Accessibility Mapping

The growing field of urban analysis is increasingly using advanced computer vision techniques to comprehensively understand and map city characteristics. These tools have been gaining popularity, especially for automating time-consuming tasks that can be prone to human error. In fact, manual audit of accessibility and other general urban features is costly, subjective, infrequent and time-consuming tasks. Therefore, the exploration of novel technologies became essential to accelerate the characterisation of multiple urban features.

Several recent initiatives have explored the use of computer vision, crowd-sourcing and street-level imagery to map urban accessibility constraints. The most relevant projects are using GSV images together with machine learning to perform city analytics and assess walkability [17], to automatically validate crowd-sourced labels and tag sidewalk accessibility issues [5, 18, 19], to perform sidewalk extraction [20], to build a crowd-sourcing-based disabled pedestrian service [21], to measure street-level walkability [22], to build street-level sidewalk GIS data [23] and to compare automatic and manual curb ramp labelling [4]. Other urban analysis projects include, for example, Hosseini et al. [9] that proposes the Tile2Net framework to extract sidewalk, crosswalk and footpath polygons from orthorectified aerial imagery in developed urban contexts; also Hamim et al. [10] adapt a similar HRNet + OCR based approach using GSV images to identify roads and sidewalks. ACAMAI also addresses generalisation by training a YOLOv8 object detection model on a combined international dataset (PSD) and a smaller, locally annotated dataset (PTD). We improved the performance of the model for this specific region, aligned with Hamim et al. [10] emphasis on the importance of retraining models with local training images for better performance in different geographical settings. In contrast to pixel-level infrastructure mapping, other research projects use street-level imagery for broader urban insights. For example, Gebru et al. [24] applied Convolutional Neural Networks (CNNs) and a Deformable Part Model (DPM) to analyse 50 million GSV images for fine-grained vehicle classification to infer socioeconomic characteristics and voting patterns of neighbourhoods. Similarly, Rangel et al. [25] automate the understanding and mapping of city regions from GSV images. They employ pre-trained CNNs (e.g., Places-GoogLeNet) to generate image-level semantic descriptors, which are then clustered and smoothed

using K-nearest neighbours (KNN) to categorise urban zones. These projects demonstrate the versatility of computer vision in extracting valuable insights from various sources of urban imagery, ranging from detailed infrastructure mapping to broader social and land use analysis. The primary objective of ACAMAI is distinct from these projects, which focus on a broader demographic or land-use mapping. ACAMAI focuses on automating the detection and geolocation of specific urban accessibility restrictions that hinder pedestrian movement. Therefore, providing an effective tool for the timely planning of corrective intervention by authorities.

More particularly related to urban accessibility mapping, Project Sidewalk [26] allows citizens to label curb ramps and obstacles directly in GSV, while OASIS [27] proposes an automated system for mapping sidewalks using embedded sensing and neural networks. Other efforts such as the Mapillary platform [28] and Crowd + AI-based models [29] have used imagery and public participation to expand sidewalk coverage at scale. In contrast to these approaches, our proposal aims to fully automate accessibility constraints detection from GSV panoramas and directly integrate results into OpenStreetMap (OSM), with a focus on panoramic geometry and geospatial precision. Incorporating accessibility information into mapping services is also an important goal and the OpenSidewalks project [30] is moving forward with standards that can be used in OSM. The AccessMap project [31] is also using crowd-sourcing to feed accessibility features into maps. In general, street view imagery is now a part of urban analytics and GIScience [17].

Most of these projects are using Computer Vision technologies, such as deep CNNs which are state-of-the-art methods for classification and feature extraction in images [32], with consistently good results. Networks are trained in labelled elements from the datasets or trained in other datasets and then their learning is applied to other sources of data. Transfer learning can also be used during preprocessing to identify images with occluding objects, thus reducing outliers. The use of transfer learning could be particularly useful since labelling street datasets on a regional or national scale is a complex and time-consuming task, even when assisted by semi-automatically crowd-sourcing processes [5].

Trained deep convolution networks have also been used with other sources of information to improve the classification of accessibility characteristics and thus typically offer a reliable data source for city planning and route generation [19]. The work presented by Weld et al. [19], supported by the Project Sidewalk dataset of 300,000+ image-based sidewalk accessibility labels [26], present an innovative use of deep learning to automatically assess sidewalks in GSV panoramas. The same work investigates two application areas: (i) automatically validating crowd-sourced labels and (ii) automatically labelling sidewalk accessibility issues. For both tasks, Residual neural Networks (ResNet) were used. In the work presented by Weld et al. [19] some open questions are related to the possibility of experimentation on different training sizes to analyse the impact on model performance, for instance, on whether the model generalised well enough to be applied to different cities. The Sidewalk project, hosted at the University of Washington, made available an accessibility labelling tool, used in several cities in the United States and also in Europe [33]. This tool allowed for collecting a large dataset of geolocated sidewalk accessibility labels, which can potentially be used in new transfer learning approaches. A more recent project combines the use of YOLO together with a CNN, the former to detect the presence of pedestrian path problems and the latter to evaluate the degree of damage [34]. The addressed city path problems are strictly focused on obstacles and damaged pedestrian paths (cf. surface problems), hence not covering specific accessibility issues, such as the presence or absence of curb ramps in crosswalks, which are major concerns regarding city accessibility mapping.

The experimental work presented in this paper explores the use of YOLO object-detection models [6] to automate the identification and mapping of city accessibility constraints. Our results are compared with those of Duan et al. [7], which trained per-city models, using crowd-sourced datasets filtered by label approval ratings. To the best of our knowledge, previous approaches have typically relied on the training of separate CNNs tailored to individual cities and per-image classification of single accessibility issues. In contrast, our work introduces a two-stage pipeline designed to generalise across different urban contexts: the first stage focuses on training a YOLOv8 model to detect several accessibility features in 2D sidewalk views, while the second stage addresses the automated scanning of 360° GSV panoramas to extract and process multiple perspective images for large-scale constraint detection and mapping.

3 | Phase 1: Detection of City Accessibility Objects in 2D Images

The first phase of the Automating City Accessibility Mapping using AI (ACAMAI) project addresses the training and testing of several YOLO object detection models, using combined city accessibility datasets from several cities of Mexico, United States of America, Netherlands and Portugal.

3.1 | 2D-Image Datasets Preparation

For the purpose of training and comparing YOLO models, two datasets were collected. The first dataset, which we named Project Sidewalk Dataset (PSD), was collected using the Project Sidewalk API [26], which serves image labels with location and accessibility attributes. Currently, this service provides annotated images from 11 cities, of which three are in Mexico (cf. CDMX, La-Piedad, SPGG), one in Europe (Amsterdam) and the remaining seven cities in the United States. In total, the combined PSD comprises approximately 390,000 labels. The second dataset, named Porto Dataset (PTD), was manually gathered by our team with 2044 annotations of accessibility attributes from photos taken on the streets of two central areas of Porto city, in the north of Portugal.

The usable annotations in the PSD were found to be significantly less than those provided in the original Project Sidewalk dataset, due to several issues detailed below. Therefore, the initial total number of labels was filtered into a usable sub-set, as shown in Figure 2.

Since the images' label annotations were collected through crowd sourcing, the Project Sidewalk website offers a functionality to validate or challenge a specific annotation. Employing basic filters, such as excluding labels with more than two disagreements and retaining labels with a clear consensus (e.g., agree count more than the double of disagree count), it was possible for us to refine the dataset into approximately 178,000 labels. This approach is aligned with the experimentation results of Duan et al. [7], which stated that *precision* and *accuracy* for curb ramps assessment increased, though with minimal performance differences, for other types of labels.

Out of the nine possible annotations attributes, five exhibited low frequencies (cf. 'Other', 'Signal', 'Occlusion', 'Crosswalk' and 'No Sidewalk') and therefore were excluded, resulting in the retention of around 164,000 annotations with the following attributes: Curb Ramp (CR), No Curb Ramp (NC), Obstacle (O) and Surface Problem (SP).

Since Project Sidewalk only provides annotations for panoramic images, but not the images themselves, we needed to download those images from GSV's API, using the panorama identification string for each annotation. However, GSV imagery is periodically refreshed, which can result in updated image content or camera positions. As such, some of the panoramas originally referenced in Project Sidewalk are no longer directly accessible via existing links. Although the GSV API provides access to historical imagery in certain locations, the availability of historical versions may be inconsistent across the geographic scope of the dataset and many of the older panoramas linked to specific annotations cannot be accessed through the API in practice. Therefore, our filtering process retained only those annotations linked to the panoramas that were available at the time. Consequently, 67,518 labels within this subset were discarded, resulting in a more reduced dataset with approximately 96,441 labels.

Considering the potential over-representation of certain label types (e.g., curb ramp), a deliberate under-sampling strategy was implemented to achieve a more balanced dataset, since the least represented type (cf. surface problem) has 9873 labels. After reducing the occurrences of the other three types, this brings the new total of all labels to 39,492.

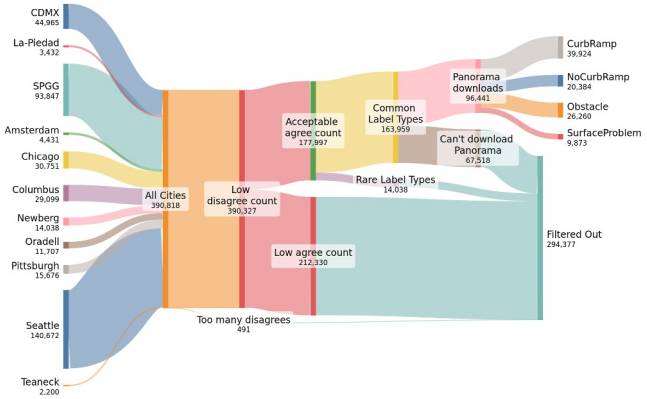

FIGURE 2 | Sankey' diagram illustrates aggregated data from 11 cities, filtered by usable labels.

Besides the labels, the Project Sidewalk API service uses several fields to characterise each panorama of the dataset: (i) labels of specific types of accessibility issues (cf. curb ramp, obstacle, etc.); (ii) an identification string, used for downloading the associated GSV panorama; and finally (iii) several other fields locating the annotations in the panorama; due to its complexity, these fields are explained in more detail with a step-by-step description in the next sub-sections.

3.2 | Processing 360° Panoramic Images

An equirectangular panorama, as seen in Figure 3, is a specific type of panoramic image that is formatted to be displayed in a 2:1 aspect ratio. This representation is commonly used in virtual reality applications or in 360° photos. The equirectangular projection maps the spherical panorama onto a flat surface, with the x-axis representing the full 360° of horizontal rotation, which we refer to as yaw (θ) and the y-axis covering 180° of vertical rotation, referred to as pitch (φ).

In Figure 3, the yaw is measured in degrees (°), in the signed range [−180°, 180°]. Likewise, pitch is also measured in degrees (°), in the signed range [−90°, 90°]. Direct visualisation of these images is not recommended due to inherent distortion. For optimal viewing, a specific section of the panorama must be selected and transformed into a 2D plane. This process is divided into two steps: define (i) how much of the panorama we want to project and (ii) where in the panorama we want to look at.

To determine the section size to extract from the panorama, we specify the horizontal and vertical angles that define the *Field Of View* (FOV). Figure 4 illustrates (in a 2D view for easier visualisation) how these angles widen or narrow the view of the panorama. The blue arc in the figure can be seen as the spherical panorama image and the blue line as the projection of that spherical image onto a 2D plane or 1D line, in this case strictly for visualisation purposes.

However, an issue arises because the Project Sidewalk service does not provide the vertical FOV (VFOV) in which the images were captured. Consider a scenario involving two cropped images: one where the horizontal and vertical FOV are the same, thus resulting in a square image; and another where the

FIGURE 3 | Panoramic image with visible distortion areas away from the centre.

horizontal FOV (HFOV) is larger than the vertical, hence resulting in an image that is wider than taller. Therefore, changing the two FOV angles alters the ratio between the width and height of the resulting image, commonly called the aspect ratio. Likewise, modifying the aspect ratio alters one of the FOV angles while keeping the other angle fixed.

Due to the nature of streetscape imagery that has more information displayed horizontally than vertically, it is often more useful to change the horizontal FOV (HFOV) instead of VFOV. Moreover, since for image processing we need to extract a squared or rectangular shape, the VFOV remains dependent on the required aspect ratio, through the Formula (1).

$$VFOV = 2 \times \arctan\left(\frac{\tan\left(\frac{HFOV}{2}\right)}{ASPECT\_RATIO}\right) \quad (1)$$

The aspect ratio of an image is calculated by the ratio of width to height. Therefore, we effectively set the controlling variables of the size of the section designated for cropping as: the horizontal FOV, the *width* and the *height* of the cropped image. Once these parameters are set, the crop section is rotated to the correct position using two additional parameters: the *yaw* and *pitch* angles. The result is the final properly oriented cropped section represented in a 2D plane.

Hence, a cropped section from a panorama may be converted into a 2D plane by controlling five parameters: *yaw*, *pitch*, horizontal FOV, *width* and *height* of required 2D image. The Project Sidewalk API provides all these parameters, offering perspectives with a resolution of 720 × 480 pixels (corresponding to an aspect ratio of 3:2) and a horizontal FOV of 89.75°, 53.0° or 27.682°.

Moreover, two further issues warrant attention. First, the provided *yaw* and *pitch* are relative to the image *heading*, in relation to the north, according to GSV standards. To compensate, metadata scraping from GSV panoramas is necessary to obtain this information, which changes for each panorama. Second, the API does not align the angles directly with the label annotations; instead, they are used to generate a crop containing the annotation. The (*x*, *y*) coordinates provided for each label tell us the position of the annotations in the cropped image. To simplify the workflow, these coordinates and photo *heading* offsets are integrated into the *yaw* and *pitch* angles. The 2D image extraction process was based on a library from Wang et al. [35], which facilitates the use and integration of the necessary parameters, along with the panorama image, to obtain the desired crop.

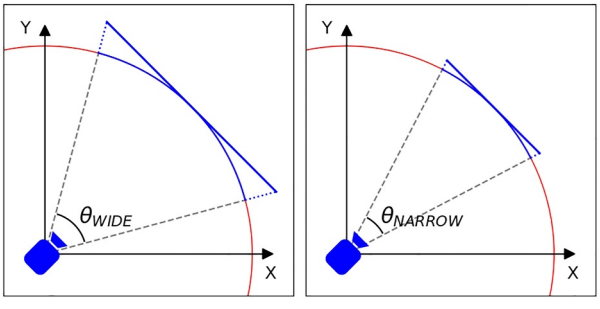

FIGURE 4 | Illustration of how a section (blue) of a panorama (red) can change sizes by altering the FOV (θ_{WIDE} is 60° and θ_{NARROW} is 35°).

3.3 | Object Detection Bounding-Boxes

The annotations on the Project Sidewalk Dataset (PSD) are made as point annotations overlaid on the panoramas, that is, each accessibility issue spot is placed over the panorama. Therefore, for running the object identification process, we first need to extract 2D perspective images from the panorama. Duan et al. [7] extracted 2D images 'zooming' on the area surrounding the annotation spot, so that the ResNet model can focus on each accessibility problem at a time. In our case, we expand the extracted 2D image by a factor of five and set the bounding boxes to 20% of the 2D image extracted from the panorama. This approach was chosen as a practical approximation, given that the PSD provides only point annotations without bounding-box sizes.

Object detection enables the identification of one or more objects per image, as well as their spatial location in a scene, by calculating a boundary box, as exemplified by the one depicted in Figure 5. For example, the green bounding box highlights where in the image a curb ramp is, as well as how much of the image it occupies. The red bounding box identifies a no curb ramp associated with the crosswalk present in the same figure. However, these bounding boxes are of different sizes, presenting a challenge in terms of control within the PSD dataset.

Since Project Sidewalk Dataset (PSD) only provides annotations in the form of points marked over the GSV panoramas, it is necessary to extrapolate the appropriate dimensions of the bounding box for each annotated point. However, choosing the size of the bounding boxes presents two related challenges: first, objects further away may have smaller boxes; and second, objects with different vertical and horizontal proportions may also have different size bounding boxes, such as the pole and the plants in the top image of Figure 6.

Although variable-size bounding boxes could be explored, that was a parallel issue not in the main scope of our approach. For practical reasons, all experiments reported in this paper adjusted the width and height of the bounding boxes to 20% of the width and height of the respective 2D image extracted from a panorama. This rule of thumb enables the size of the bounding

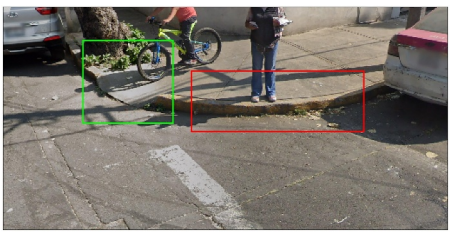

FIGURE 5 | Image with two objects of interest: curb ramp (green) and no curb ramp (red).

box to be adjusted relative to the width and height of each extracted 2D perspective image (and associated accessibility objects) rather than the full panoramic image.

3.4 | Evaluating the Object-Detection Approach

Several experiments were carried out with the YOLO architecture, using its version 8 model, YOLOv8 [6], known for its efficiency in real-time object detection. More specifically, the YOLOv8x variant, with 68.2 million parameters, was used in all experiments. On all experiments with the YOLOv8x model, the datasets were split into 3 parts, 70% of the data for training, 15% for validation and the remaining 15% for testing (see Figure 7).

A base model with provided pre-trained weights was used, a standard practice in deep learning, which allows the final model to benefit from knowledge transfer, gained during the base training, often resulting in faster convergence and improved performance. The weights of the base model were obtained by training the model on the Microsoft COCO dataset [36].

All training, validation and testing cycles were conducted on a machine running Microsoft's Windows Server 2022 operating system, equipped with 128 GB of RAM and a NVIDIA RTX 3090 GPU graphic card with 24 GBs of VRAM.

Each of the experiments conducted followed a specific strategy for collecting the datasets and then trained, validated and tested the YOLOv8x model:

- Strategy 1 (S1): Train, validation and test performed with Project Sidewalk Dataset (PSD) only;
- Strategy 2 (S2): Train, validation and test performed with Porto Dataset (PTD) only;
- Strategy 3 (S3): Train, validation and test performed with PSD + PTD, considered jointly;

In all experiments, the datasets were built with images centred on the centroid of each annotation and containing a bounding box with 20% of the 2D image dimensions. Table 1 summarises the *precision* and *recall* results obtained, which are comparable and in some cases better than the baseline ResNet approach published by Duan et al. [7].

Table 1 depicts the overall *precision* (p), *recall* (r) and *F*1 values of the YOLOv8x models trained, validated and tested according to the selected strategies. Figure 7 illustrates the split percentages used for each dataset and considered in the specific evaluation strategies.

Table 2 shows additional *precision* (p) and *recall* (r) results on a per-object basis; however, with the YOLOv8x S3 model applied to 15% of the PTD.

Strategy 1 (S1) demonstrates that using an object-detection model on the Project Sidewalk Dataset (PSD), is not only possible but also allows obtaining comparable metrics against previous ResNet-based projects reported in the literature [7].

However, when using the YOLOv8x model, obtained from S1, for testing with the Porto Dataset (PTD), we observed that the model exhibited poor generalisation. This outcome can be attributed to particular urban characteristics of Porto city, such as the abundant use of cobbled streets and characteristic Portuguese sidewalk paving, which may confuse the model, which was trained primarily on North American data.

The YOLOv8x model obtained from Strategy 2 (S2), that is, trained solely with PTD, was expected to offer better results. However, we observed only marginal improvements in one type of object, though with the same or worse results in all other types of objects/labels. This is potentially attributable to the limited manually collected data.

The model obtained by Strategy 3 (S3) achieved better results as shown in Table 2. The S3 model, which combines training on the large-scale Project Sidewalk Dataset (PSD) with the smaller, locally annotated Porto Dataset (PTD), is a key finding with implications for model generalisation and deployment. The initial poor generalisation of the S1 model (trained solely on PSD) to PTD highlighted the sensitivity of deep learning models to regional urban characteristics, such as Porto's abundant cobblestone streets and distinctive sidewalk paving. S3 effectively mitigated this issue by leveraging transfer learning and fine-tuning with local data, demonstrating that even a relatively small local dataset (PTD, with 2044 annotations) can significantly adapt a globally trained model to specific urban contexts. This suggests a cost-effective and scalable strategy for rapidly deploying ACAMAI in diverse cities: starting with a broad, international dataset and augmenting it with targeted local annotations for fine-tuning.

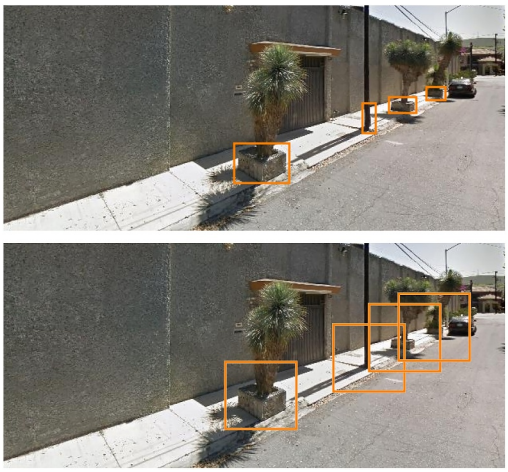

FIGURE 6 | Illustrating how the size of a bounding box can result in improper labelling.

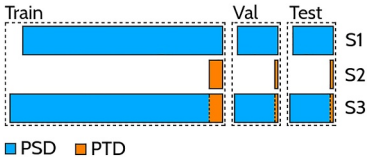

FIGURE 7 | Datasets' splitting percentages considered in the evaluation strategies.

TABLE 1 | Test metrics for YOLOv8x with different evaluation strategies—Baseline Duan et al. [7] refers to results with ResNet.

Model	p (%)	r (%)	*F*1 (%)
Baseline ResNet [7]	73.9	72.1	73.0
S1 with YOLOv8	84.5	84.2	84.3
S2 with YOLOv8	65.1	30.3	41.4
S3 with YOLOv8	84.7	83.5	84.1

TABLE 2 | Test metrics obtained by S3 YOLOv8x when applied to 15% of PTD.

	p (%)				r (%)			
	CR	NC	O	SP	CR	NC	O	SP
S1	74.6	35.5	59.1	1.9	30.1	39.3	12.4	5.6
S2	77.6	0.0	75.0	0.0	28.8	0.0	45.7	0.0
S3	85.0	82.1	92.9	70.0	91.0	82.1	75.2	38.9

Note: Results on per-object basis (cf. Curb Ramp—CR; No Curb Ramp—NC; Obstacle—O; Surface Problem—SP).

The confusion matrix in Figure 8 illustrates the performance of the S3 YOLOv8x model in all four classes of objects when applied to 15% of PTD. There are high true positive rates for curb ramps (91%) and no curb ramps (82%), good rates for obstacles (75%) despite some confusion mainly with the curb ramp class, although limited performance for surface problems (39% recall), which exhibit the highest misclassification rate, likely due to the scarce dataset and visual ambiguity of such problems.

Figure 9 illustrates nine characteristic YOLOv8 labelling scenarios in the PTD dataset. For example: (i) Tile [0][0] shows a cobblestone road with no curb ramp (NC), which is a common feature in historic areas of Porto; (ii) Tile [0][1] identifies a surface problem (SP) even at a relatively distant position in the image; (iii) Tile [0][3] presents a no curb ramp (NC) between two cars, clearly associated with a well marked crosswalk; (iv) Tile [1][0] depicts an obstacle (O) that appears to be associated with the parked car, which is not a permanent accessibility

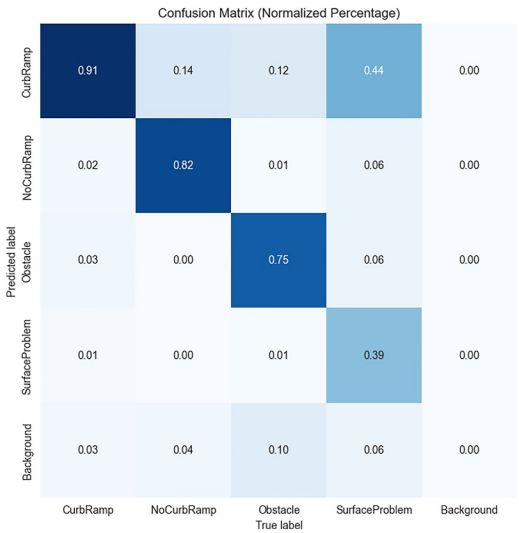

FIGURE 8 | Confusion matrix for S3 YOLOv8x model when applied to 15% of PTD.

FIGURE 9 | Tiled scenarios illustrating YOLOv8 accessibility labelling.

issue; (v) Tile [1][1] identifies a curb ramp (CR), which is associated with two crosswalks, one well-marked in the street and another old-faded crosswalk above it; (vi) Tile [1][2] shows two objects (cf. obstacle and curb ramp) despite the excessive brightness in that area of the image; (vii) Tile [2][0] also shows two obstacles (O) that have been placed to prevent vehicles from passing through the identified area; (viii) Tile [2][1] again shows two objects (cf. obstacle and curb ramp), which are associated to the same crosswalk; (ix) Tile [2][2], finally presents two curb ramp (CR) that are also associated with the same crosswalk. These examples show various urban scenarios in Porto and illustrate the ability of YOLOv8 to detect multiple accessibility features under varying visual and contextual conditions.

The competitive performance of our S3 YOLOv8x model compared to the baseline ResNet approach indicates the robustness and efficiency of YOLO architectures for real-time object detection in complex street view environments. This capability is critical to enable the timely identification of accessibility issues, moving beyond time-intensive and costly crowd-sourcing efforts. In particular, the high true positive rates for curb ramps (91%) and no curb ramps (82%) underscore the potential of ACAMAI to provide accurate data for urban planners focused on pedestrian infrastructure.

Despite these strengths, the current model exhibits limited performance for surface problems (39% *recall*), which can be associated with the scarce representation of surface problems in PSD + PTD datasets and the visual ambiguity of such issues in street-level imagery. More data collection will be necessary to address this specific challenge.

4 | Phase 2: Scanning GSV Panoramas for Accessibility Objects

The second stage of our project involves the iterative analysis of GSV panoramas to identify accessibility issues on sidewalks.

Given a GSV 360° panoramic image, it is necessary to thoroughly scan the image with the previously tuned YOLOv8 model to pinpoint accessibility objects. The scanning process traverses the panoramic image both vertically and horizontally so that the selected YOLOv8 model can fully screen it at all points around the 360°. As illustrated in Figure 10, the scanning process of each panoramic image involves the use of vertical delta-yaw ($\Delta\theta$) and horizontal delta-pitch ($\Delta\phi$) displacements to extract perspective 2D images. Each of these 2D images are then fed to YOLOv8 to identify the predefined accessibility object-classes. In this way, each panorama is fully covered for extracting all possible 2D perspectives, providing the YOLOv8 model with multiple opportunities to scan and detect every existing accessibility spot.

4.1 | GSV Panoramas Porto Dataset Preparation

For processing and evaluating the scanning process, a new GSV-Panorama Porto Dataset (GSV-PPD) was manually collected and annotated for several streets of the Paranhos parish in Porto city. The annotations were collected by architects, who identified in each GSV panorama all accessibility issues. This GSV-PPD contains 99 panoramas, with 438 annotated accessibility issues. The distribution of accessibility classes is shown in Figure 11. The GSV-PPD dataset was built as a ground truth, to be used during the execution and evaluation of the second stage pipeline, described in the following subsections.

4.2 | Downsizing Search Space Within Panoramic Images

The scanning pipeline of each GSV panorama, depicted in Figure 12, undergoes three stages, starting by loading the GSV panorama, then scanning the GSV 360° panoramic image (by varying yaw and pitch by delta increments) and finally eliminating redundant pinspots by clustering them into a single set. This process, although automatic, is computationally intensive and considerably resource-demanding, especially the second stage (cf. extracting all 2D perspective images and passing them through the YOLOv8 model) as it involves iterative repetition to cover the entire 360° panorama.

Figure 13 illustrates the ACAMAI scanning process when applied to different urban scenarios. It uses selected frames from two animated GIFs depicting anticlockwise rotations over one panorama from the PTD dataset (a) and another panorama from the PSD dataset (b). Each picture in the figure summarises three moments of the scanning sequence, beginning with an initial frame, followed by two more frames revealing a similar detection pattern. First, an obstacle is identified, followed by a curb ramp as the scan rotates from right to left across each panorama.

The scanning middle phase is more costly the smaller the delta increment in the yaw and pitch values. Therefore, processing efficiency could be enhanced by reducing the search space within each panoramic image, that is, constrain the range of yaw and pith values that should be covered in the search for accessibility spots. Therefore, we conducted an analysis of the existing annotation space within the Project Sidewalk Dataset (PSD). Figure 14 plots all annotations in the crowd-collected PSD, revealing a trend towards a narrower range of pitch values, approximately below 20° and above −62°. This observation is logical, as although yaw values (horizontal scanning range) can cover the entire 360° range, pith values (vertical scanning range) tend to be limited to a central interval because

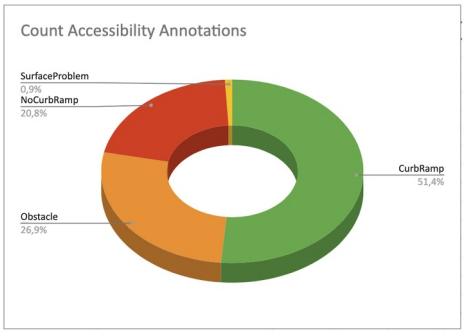

FIGURE 11 | Distribution of annotated accessibility issues per object-class in GSV-PPD dataset.

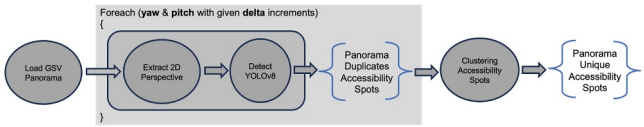

FIGURE 12 | Scanning pipeline for pin-spotting accessibility constraints over each GSV panorama.

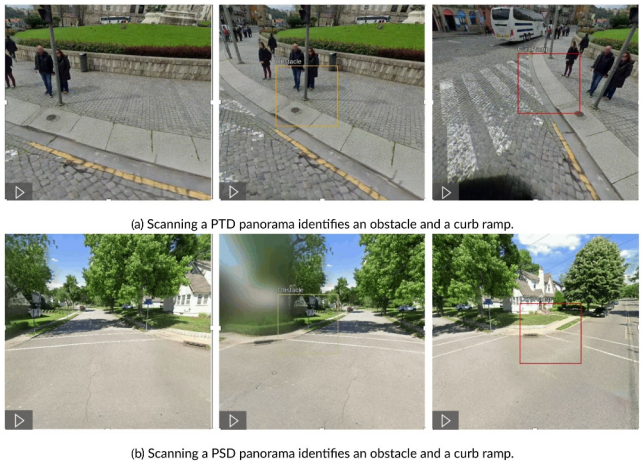

FIGURE 13 | Illustrative anticlockwise rotation scanning examples of two different panoramas: (a) PTD sample; (b) PSD sample.

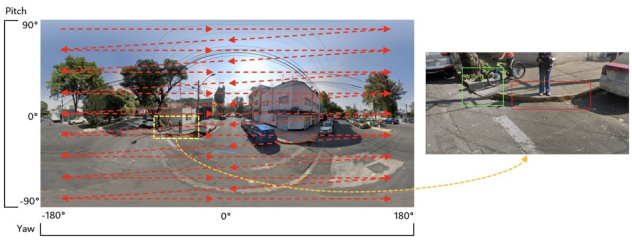

FIGURE 10 | Scanning process of a 360° panoramic image.

accessibility spots will not exist or cannot be identified higher towards the sky and lower in close ground.

The lower range of pitch values reflects the position of the camera on top of the vehicle. This configuration inherently results in a downward pitch to capture sidewalks that lie below the camera's level. The range of yaw values is generally uniform, except that it tends towards two values 0°/360° (front of the vehicle) and 180° (rear of the vehicle). At these yaw angles, the pitch values become more constrained. This behaviour is attributable to the fact that sidewalks, in typical urban environments, are placed on the sides of the vehicle rather than directly in front or behind it, as seen in Figure 15.

The panoramic scanning space can consequently be reduced by 44.32%, while retaining 100% of accessibility spots. This substantially limits the pitch search range without loss of data annotations. Figure 16 shows the distribution of 100% of PSD data annotations between the range 20° < pitch < −62°. The red dots represent the centre points of each 2D perspective extracted from the panorama to be analysed by the YOLOv8 model to classify accessibility objects.

In practice, due to the sum in the vertical boundaries of the vertical FOV in each scan window, the total panorama area covered by the scanning addresses pitch values ranging from 62° to −90°, as illustrated in Figure 17. This is due to the fact that the extracted 2D perspective images extend, in VFOV/2, beyond the sampling centre points, thereby encompassing a broader range of pitch values. The adjacent 2D perspective images extracted from each panorama will exhibit overlapping areas. This overlap is necessary, as the YOLOv8 model may fail to identify certain accessibility spots if non-overlapping perspectives are utilised.

4.3 | Clustering Groups of Individual Accessibility Objects Within Panoramic Images

The scanning process iteratively extracts successive 2D perspectives with overlapping areas to avoid missing accessibility spots. However, this results in repeated identifications of the same spots, leading to redundant intra-panorama accessibility spot detections that require elimination. Unsupervised learning solutions based on clustering algorithms were explored to group sets of the same accessibility spots detected by YOLOv8 on the same panorama.

Given the nature of the problem, we selected the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. Unlike K-Means, which assumes spherical cluster shapes and requires a predefined setup number of clusters, DBSCAN is more well-suited to our use case as it identifies clusters of arbi-

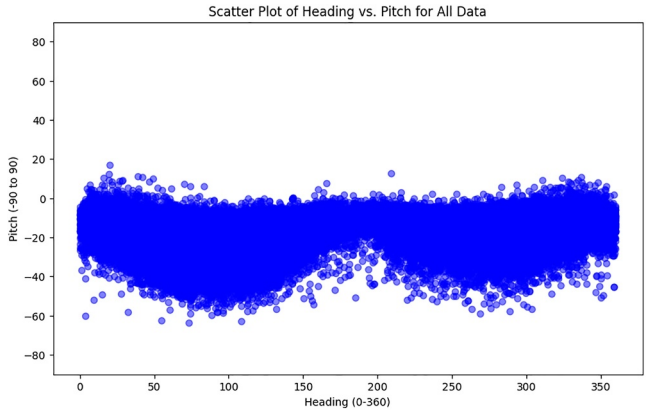

FIGURE 14 | Distribution of yaw and pitch values for crowd-based PSD annotations.

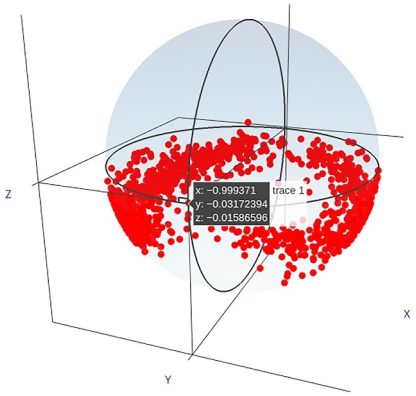

FIGURE 15 | 3D scatter plot distribution of yaw and pitch values for crowd-based PSD annotations.

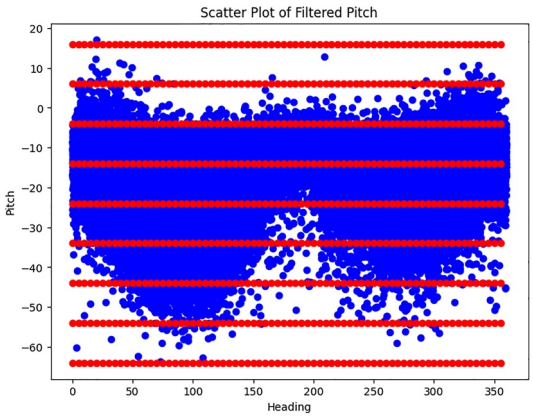

FIGURE 16 | Filtered distribution of accessibility annotations in the PSD (20° < pitch < −62°).

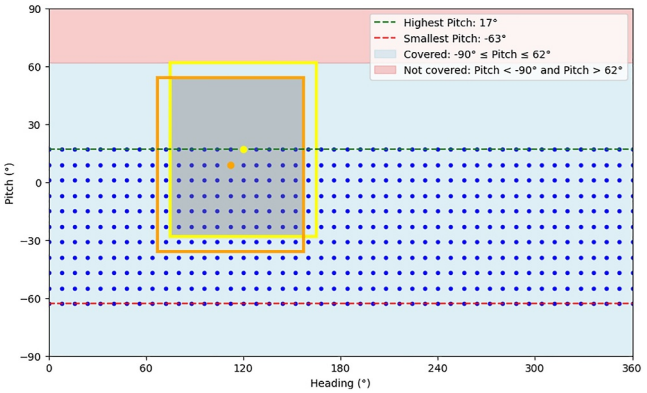

FIGURE 17 | Real range of pitch values scanned on every GSV-PPD.

trary shapes and automatically determines the number of clusters based on density. This is particularly advantageous given the spatial nature of our accessibility detections, which may vary in distribution of the same real-world location across multiple overlapping projections. Additionally, DBSCAN inherently handles noise by distinguishing sparse outlier detections from dense clusters, improving the robustness of the final accessibility constraint localisation. These characteristics make DBSCAN a more appropriate choice than K-Means for consolidating repeated detections into distinct accessibility issues.

The DBSCAN algorithm is parameterised with two key parameters: (i) `eps` - radius within which detections are considered close enough to form one cluster; (ii) `min_samples` - number of detections required to form a valid cluster. The `eps` used was 3, reflecting an expected spatial tolerance between detections of the same object, empirically set based on maximum distances between related predictions. The value of `min_samples` was 2, reflecting the minimum number of overlap detections expected for a true positive to be considered reliable. This setup enables the algorithm to effectively merge redundant detections while treating isolated false positives as noise. Figure 18 shows the clustering results of the DBSCAN algorithm for one panorama of the GSV-PPD dataset. As observed, accessibility spot detections typically exhibit vertical distribution shapes (non-spherical) and may vary in number, thus rendering it challenging to pre-estimate the number of clusters that could exist in each panorama.

4.4 | Evaluating the Scanning Process of Panoramic Images

The evaluation of the scanning pipeline was performed on the same Server machine used for the previous YOLO models train/test experiments. The scanning process was conducted on each panorama, with the aim of allowing YOLOv8 to iteratively go through the entire panorama to identify all accessibility issues. To this end, the algorithm successively extracts 2D projections, spaced by delta-step over the ranges of pitch ($\Delta\phi$) and yaw ($\Delta\theta$), which are then analysed by YOLOv8 to predict accessibility spots. The goal was to ascertain the most effective scanning

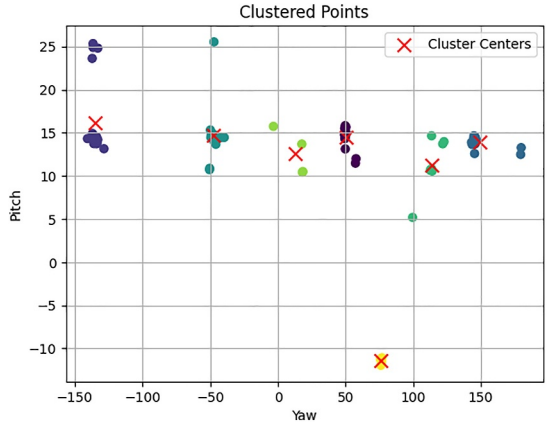

FIGURE 18 | Example clustering results using DBSCAN for a GSV panorama of GSV-PPD with seven accessibility spots.

delta-variation that allows maximising the pinpointing of accessibility issues in all 99 annotated panoramas of the GSV-PPD. The evaluation was guided by a central principle: prioritising the automated identification of all accessibility spots annotated in GSV-PPD, even if this may lead to the detection of additional, non-annotated instances.

The time profile of the scanning pipeline is crucial for evaluating the overall process. The scanning pipeline consists of four different stages. For profiling purposes only, an additional stage was introduced to estimate the distance of each accessibility spot estimate relative to the camera viewpoint in the panorama. Hence, these are the five stages of our evaluation pipeline:

- Task A: Reading/Loading one panorama from disk;
- Task B: Generating a single 2D perspective from a panorama;
- Task C: Executing YOLOv8 model inference [6] of accessibility objects from a 2D panorama;
- Task E: Clustering per-panorama predictions to group repetitive/redundant accessibility spots;
- Task D: Performing Depth model inference [37] to estimate the distance of each predicted accessibility spot centroid in the panorama.

The depth values for each panorama pixel can be estimated using the Depth Anything V2 model, which provides dense monocular depth predictions from 2D perspective images [37]. For each of these 2D perspectives, the model generates a depth map in which the value of each pixel represents the Euclidean distance from the camera centre (i.e., the original GSV viewpoint) to the corresponding point in the scene. These depth values enable the assignment of real-world distances to both predicted accessibility spot centroids and annotated ground-truth points. Given these values for each GSV-PPD panorama, we can compute the spatial distance between each predicted accessibility spot and its closest annotation. To calculate the Euclidean distance `D` between a particular annotation and a predicted centroid in the panorama, we consider their respective depth values, d_a and d_p, along with the angular distance α between them (derived from yaw and pitch). The distance is then computed as:

$$D = \sqrt{d_a^2 + d_p^2 - 2d_a d_p \cos(\alpha)} \tag{2}$$

These `D` spacing measures can then be used to assess proximity and therefore the association between predictions and existing annotations. Such distance measures are used for matching/pruning predictions and accounting scores (true and false positives and negatives with respect to accessibility spot estimates and annotations). For this matter, it is important to emphasise that each panorama has potentially more non-annotated accessibility spots than the ones collected by our team, since the annotations were only made closer to the camera field of view in the panorama but not far away in the panorama.

Figure 19 illustrates the time allocated to each scanning step of a GSV panorama. Stage A requires the most time as it involves I/O

operation with Disk, which operates at a significantly slower rate than CPU/GPU tasks. However, Stage A is executed only once for each panorama and is independent of the delta-step used for scanning. Consequently, the weight of Stage A in the overall scanning process is diluted when compared against the sum of all the other stages. Stage D is the second most time-consuming task; however, it is not part of the scanning pipeline because it was only used to calculate distances between accessibility spots estimates and annotations and therefore accounts for hit rates of those accessibility spot predictions.

Moreover, as illustrated in Figure 20, the time spent on the comprehensive scanning process of a single GSV panorama is dependent on the delta-step. It is observed that as the delta-step decreases, the time allocated to specific tasks increases considerably. Specifically, Stages B and C exhibit a substantial increase in time consumption, owing to the fact that as the delta-step diminishes, the number of 2D perspectives that must be extracted and the frequency of YOLO predictions that must be executed for each perspective also increases. Stage D increases considerably too for lower delta-steps; however, this is not problematic for the scanning pipeline as Stage D is not part of it (since it was needed/included just for evaluation purposes).

The time profiling results reveal that the pipeline tasks involving generation of 2D perspectives (Task B) and YOLOv8 inference (Task C) are the most time-consuming. The significant increase in processing time as the delta-step decreases directly reflects the computational burden of higher-resolution scanning. This presents a critical design consideration for practical deployment: a tunable delta-step provides flexibility for different operational needs. For comprehensive initial audits, prioritising high *recall* with a smaller delta-step might be acceptable, even with longer processing times. Conversely, for continuous monitoring or verifying known issues, a larger delta-step could be chosen to optimise for efficiency and lower computational cost, accepting a potential trade-off in detection coverage.

Furthermore, the reduction of 44.32% in panoramic scanning space while retaining 100% accessibility spots through constrained pitch range has major implications for improving computational efficiency and scalability. Focussing scanning where accessibility features are most likely to occur allows ACAMAI to significantly reduce unnecessary processing, making large-scale deployments more feasible.

The *recall* and *precision* metrics are typically used together to evaluate the system's effectiveness in identifying true accessibility constraints and its reliability in avoiding false positives, respectively. More precisely, the *recall* metric quantifies the ability to identify all pertinent accessibility locations within each panorama dataset, while *precision* metric evaluates the ability to identify only the relevant annotated spots. However, during the scanning process, the YOLOv8 model detects a significantly higher number of accessibility locations than those that were annotated. This discrepancy arises because, during the annotation process, the team responsible for annotating the panoramas exclusively marked accessibility spots in close proximity to the camera's viewpoint. Consequently, any accessibility spots far away from the camera's viewpoint were not annotated. This fact results in these distant YOLOv8-detected accessibility spots being classified as false positives. However, these spots may, in fact, represent true positive accessibility spots that were not annotated in the GSV-PPD. This partly accounts for the lower *precision* metric results.

Figure 21 shows that as the delta-step increases, *recall* decreases while *precision* increases. This indicates that coarser panoramic scanning reduces the system's ability to detect all annotated accessibility constraints, leading to lower *recall*, but also reduces the number of redundant detections, resulting in higher *precision*. It is important to emphasise that surface problems are considerably less common in our datasets, thus constituting the least represented class. Consequently, the trained YOLOv8 model shows worse object identification metrics for surface problems as presented in Section 3.4.

Macro-average calculates metrics individually for each class to average the results. Hence, each class contributes equally, regardless of how many samples it contributes with. Thus, even if surface problems are rare and curb ramps are common, macro-average assigns them equal weight when computing average *recall* and *precision*. These measures help to understand how well the model performs across all classes, including those underrepresented. Micro-average aggregates true positives (TP), false positives (FP) and false negatives (FN) across all classes for computing metrics. Therefore, each detection contributes equally and frequent classes dominate the score. So, if curb ramps and

FIGURE 19 | Mean time per task for scanning a panorama.

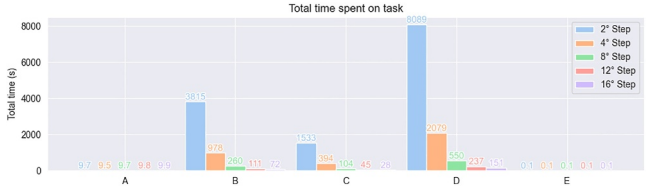

FIGURE 20 | Global mean time per task of an entire panorama scanning process.

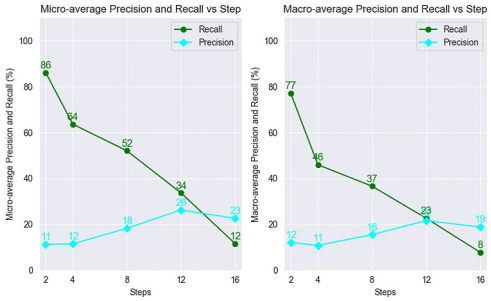

FIGURE 21 | Overall classes micro and macro-average *recall* and *precision* versus delta-step used.

obstacles are more dominant, the micro-average will mostly reflect how well the model detects these classes, possibly overlooking weaker performance on rare classes. Micro-average gives a better sense of overall detection performance, weighted by class frequency.

In summary, Macro-average computes metrics equally across all classes, highlighting performance on rare categories, while micro-average aggregates detections across classes, emphasising overall performance driven by more frequent classes.

Table 3 summarises the best *precision* and *recall* values achieved for each accessibility constraint class during the evaluation of the GSV-PPD dataset. These values were observed with the smallest delta-step tested (2°), which provides the densest coverage during panoramic scanning. The curb ramp class achieved the highest *recall*, indicating that the pipeline is the most effective at consistently detecting these elements. The obstacle class, while slightly lower in *recall*, reached the highest *precision*, suggesting fewer false positives in its predictions. Overall, the results confirm that finer scanning granularity leads to enhanced detection performance across all classes.

Figure 22 illustrates the variation in *recall* and *precision* for each accessibility constraint class, namely curb ramp, no curb ramp and obstacle, across different delta-step values used during the GSV panorama scanning process. As the delta-step increases, *recall* decreases consistently for all classes, reflecting the reduced coverage caused by fewer overlapping 2D projections. This decline is particularly evident for the *obstacle* class, which is typically more challenging to detect. In contrast, *precision* tends to increase with larger delta-steps, due to fewer detections. However, overall *precision* values remain relatively low, which can be attributed to the nature of the GSV-PPD ground truth, that is, only accessibility constraints close to the camera viewpoint were annotated, meaning that all far away detections in more distant areas of the panoramas are treated as false positives (FP). This limitation in annotation coverage leads to an underestimation of true *precision* (TP), despite the model's ability to identify plausible constraint instances throughout the full 360° scene. Among all classes, the curb ramp consistently achieves higher *recall* and *precision*, suggesting that it is the class most reliably detected at different scanning resolutions. These trends underscore the trade-off between detection coverage and result compactness controlled by the scanning resolution.

The inverse relationship between *recall* and *precision* as the delta-step increases highlights a fundamental trade-off in panoramic scanning resolution. A finer angular iterative step maximises the chances of detecting all existing accessibility spots by extracting more overlapping 2D perspectives, thus achieving high *recall* (cf. 90% for curb ramps, 93% for obstacles). However, this strategy inherently leads to multiple detections of the same physical object, resulting in a higher number of redundant predictions and, consequently, a reduced overall precision after clustering.

The relatively low overall *precision* values observed are significantly influenced by the nature of the GSV-PPD ground-truth annotations. Our manual annotation process primarily focused on accessibility spots in close proximity to the camera's viewpoint. This means that numerous legitimate accessibility constraints located further away in the panorama, were correctly identified by the YOLOv8 model but inadvertently classified as false positives due to the absence of corresponding ground-truth labels. This leads to an underestimation of the true precision of the ACAMAI pipeline in real-world scenarios. This limitation underscores the challenges of creating exhaustive ground-truth datasets for 360° panoramic imagery and suggests that the model's actual field performance for precision might be better than the reported metrics indicate.

Finally, consistent higher metrics for curb ramps across different scanning resolutions mean that this is the most reliably detectable class, likely due to their distinct visual characteristics and importance in accessibility contexts. The more challenging detection of obstacles and surface problems, which are typically more diverse in appearance and context, highlights the need for continuous model refinement and potentially more robust training data for such varied classes.

4.5 | Mapping Accessibilities Geolocations

Following the identification and clustering of accessibility constraints within GSV panoramas, the final step of the ACAMAI pipeline consists of integrating these observations into the OSM collaborative mapping platform. Each detected accessibility constraint is associated with geographic coordinates (latitude and longitude) derived from the original GSV panorama, for subsequent upload to OSM. Rather than introducing new map elements, the priority is aligning each detection with an existing *OSM node*, selected according to the type of accessibility issue identified. More specifically, detections of curb ramps or missing ramps are matched to nodes located at the edges of pedestrian crossings—typically individual *OSM nodes* explicitly tagged as `highway=crossing` or *OSM nodes* that form the endpoints of *OSM ways* tagged with `footway=crossing`.

TABLE 3 | Best per-class *precision* (p) and *recall* (r) values obtained with the scanning pipeline applied to GSV-PPD.

Object class	p (%)	r (%)	Delta-step (°)
Curb ramp (CR)	23	90	2
No curb ramp (NC)	18	75	2
Obstacle (O)	5	93	2
Surface problem (SP)	2	50	2

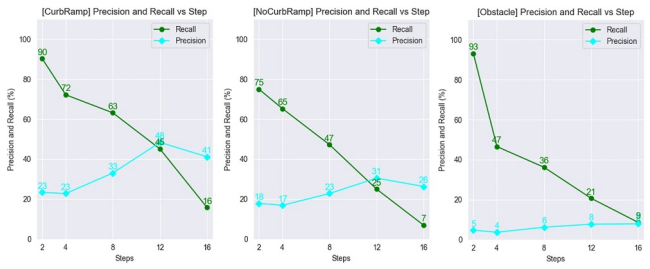

FIGURE 22 | Per-class *recall* and *precision* versus delta-step used.

Detections of obstacles or surface problems are matched to existing *OSM nodes* located along sidewalks or pedestrian paths, typically nodes that are part of *OSM ways* tagged with `footway=sidewalk`, `highway=footway` or `highway=path`.

The identified *OSM nodes* are enriched with *OSM tags* corresponding to the detected constraint:

- *Curb/Missing Ramp*: `curb=lowered` (if present), `curb=raised` or `curb=absent` (if missing), applied at crossing nodes;
- *Obstacle*: `obstacle=yes` (indicating presence of obstacle on a sidewalk segment);
- *Surface Problem*: `surface=damaged` or `smoothness=bad` (indicating degradation in walkable surface quality).

The detected accessibility constraints are first matched to the nearest appropriate *OSM node* using the Overpass API [38] and then the corresponding tags are added or updated through the authenticated OSM API OpenStreetMap Contributors [39] using the `osmapi` Python library [40]. The mapping process can be executed via the `osmapi` interface, either to submit direct edits—if contributor credentials and permissions are available—or in a *dry-run mode*, producing `GeoJSON` or `.osm` files for manual validation and upload by experienced mappers.

This final integration stage ensures that accessibility issues detected through the pipeline are translated into actionable geographic data. Using the detection of appropriate structural elements of the OSM network, the system improves data quality and facilitates its use in inclusive navigation, urban planning and accessibility auditing.

The integration of detected accessibility constraints into OSM represents a critical step in translating raw AI detections into actionable geospatial intelligence. By automating the association of identified constraints with existing OSM nodes (e.g., curb ramps with crossing nodes, obstacles/surface problems with sidewalk nodes), directly contributes to enriching open geographic data with vital accessibility information. This semantic mapping makes the generated data not only geographically located, but also contextually relevant and usable by existing GIS platforms and navigation services.

This georeferenced accessibility data in a widely used open format empowers urban planners and researchers to: (i) prioritise interventions based on a clearer understanding of problem distribution and type; (ii) develop inclusive navigation applications that can guide individuals with mobility impairments along accessible routes; and (iii) conduct urban audits more efficiently, identifying areas needing urgent attention. Nevertheless, a notable limitation in the OSM integration stage is the assumption of well-structured pedestrian networks. For example, in less-mapped regions, where OSM data for sidewalks, crossings and paths may be incomplete or absent, the system's ability to accurately match detected constraints to appropriate nodes would be compromised. This means that the automation of OSM integration relies on the existing quality and completeness of OSM data.

5 | Conclusion

Accessibility in urban environments remains a pressing equity issue, particularly for individuals with mobility impairments. This work presents a practical and scalable approach for automating the detection and mapping of city accessibility constraints through AI-assisted analysis of Google Street View (GSV) panoramas. By integrating object detection techniques within geospatial data pipelines, the ACAMAI framework aims to contribute to more inclusive and data-driven urban planning.

The proposed two-stage methodology combines (i) the training of an object detector to identify accessibility issues in 2D images and (ii) the automated scanning of 360° GSV panoramas using tunable yaw and pitch parameters. The first stage experimental results demonstrate that augmenting a large-scale international dataset (PSD) with a smaller, locally annotated dataset (PTD) leads to an improved performance in a target city (Porto). The mixed training strategy produced a YOLOv8 model that generalised well across urban contexts, outperforming baseline classification models, particularly in detecting curb ramps and obstacles in realistic street scenes. The second stage of the pipeline introduces a panoramic scanning strategy that balances coverage with computational efficiency by limiting the pitch range, leveraging FOV geometry and applying clustering to eliminate repetitive detections. Evaluation on a manually annotated panorama dataset (GSV-PPD) demonstrated that accessibility spot detection *recall* improves as the angular resolution of the scanning process increases, albeit at higher processing costs. These results validate the effectiveness of combining geometric sampling and deep learning in panoramic environments. Crucially, this study bridges the gap between detection and actionable data by mapping the identified accessibility constraints to the relevant nodes in OSM. The system distinguishes between constraint types by associating curb ramps with pedestrian crossing nodes and obstacles or surface issues with nodes along sidewalks and footways.

The ACAMAI framework represents a significant advancement towards truly data-driven and inclusive urban planning. The mixed training strategy for the YOLOv8 model demonstrated high performance and provides a scalable and adaptable approach to deploy AI-assisted mapping in diverse urban environments by combining global and localised datasets. This is a crucial implication to overcome the challenges of data scarcity and regional variations in infrastructure. Furthermore, the panoramic scanning pipeline, with tunable resolution and efficient reduction in search space, provides a flexible tool for comprehensive accessibility assessment at varying scales and computational budgets. The trade-offs between detection coverage and processing costs allow a practical guide for pipeline application. The seamless integration of georeferenced accessibility data directly into OSM is an impactful result that transforms raw detections into actionable information that directly supports the development of inclusive navigation systems, targeted urban interventions and participatory planning efforts. This approach goes significantly beyond traditional manual audits that require a lot of resources, offering a pathway to continuously updated and accessible urban data for all stakeholders.

Despite these advances, we acknowledge limitations and challenges concerning the model's current difficulty in distinguishing surface problems. This stems from the relatively small number of instances in our training datasets and the visual ambiguity of these issues, which impacts ACAMAI's ability to provide a completely comprehensive overview of all accessibility constraint types. Furthermore, the sparsity and near-camera focus of our ground-truth annotations in the GSV-PPD dataset significantly influence precision estimates. Although the model detects plausible constraints throughout the panorama, those far from the camera are often misclassified as false positives because of a lack of corresponding labels, potentially underestimating the true in-field precision. Lastly, while the OSM mapping is designed for semantic accuracy, its effectiveness is based on the presence of well-structured pedestrian networks in OSM. This implies that in less-mapped regions, the full automated utility of ACAMAI for integration might be constrained, requiring preliminary mapping efforts or greater manual oversight.

Future work should directly address these limitations and expand the framework's capabilities to further enhance its real-world impact. Improving object detection precision, especially for challenging classes like surface problems, will involve expanding and diversifying datasets and potentially incorporating human-in-the-loop feedback mechanisms for efficient validation. In addition, exploring and comparing other models, such as transformer-based object detection architectures (e.g., DETR or variants) will provide and expand valid alternatives to the process. It is also important to move beyond binary identification to include the estimation of severity levels for all constraint types (e.g., slope and width of curb ramps, degree of damage for surface problems, etc.). This will allow ACAMAI to provide richer and more nuanced data, enabling urban authorities to prioritise interventions based on the real-world impact of each accessibility barrier. Furthermore, exploring ways to ensure robust OSM integration even in less-mapped regions, perhaps through initial automated network generation, will be critical for broader applicability. Ultimately, the continuous development of ACAMAI aims to create a more precise, comprehensive and widely applicable tool to support the creation of truly inclusive, safe, resilient and sustainable cities around the world.

In general, the ACAMAI pipeline provides a reusable, open and extensible tool to support inclusive mobility assessments at an urban scale. It combines computer vision, geospatial reasoning and open data collaboration. The objective is to provide a scalable mapping of urban accessibility constraints that can be integrated into real-world tools used by planners, researchers and citizens worldwide.

Author Contributions

Rui S. Moreira: conceptualization, investigation, methodology, project administration, resources, supervision, validation, writing – original draft, writing – review and editing. **Sérgio Moita:** data curation, software, validation. **José Manuel Torres:** funding acquisition, supervision, validation, writing – review and editing. **Feliz Gouveia:** conceptualization, funding acquisition, methodology. **Maria Alzira P. Dinis:** funding acquisition, project administration. **Diogo Ferreira:** data curation. **Madalena Araújo:** data curation. **Maria João S. Guerreiro:** funding acquisition, investigation, project administration, supervision, writing – review and editing.

Acknowledgements

This work was financed by Portuguese Science and Technology Foundation (FCT), through Grant no. 2022.09218.PTDC, Project Automating City Accessibility Mapping using AI (ACAMAI) - DOI: https://doi.org/10.54499/2022.09218.PTDC. This publication was also financially supported by: UID/00027 - Artificial Intelligence and Computer Science Laboratory – LIACC - funded by national funds through the FCT/MCTES (PIDDAC).

Disclosure

The authors have nothing to report.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data available on request from the authors.

References

1. UN-Habitat, *United Nations Human Settlements Programme - The New Urban Agenda*. Technical report (United Nations, 2020).

2. WHO, *World Report on Disability*. Technical report (World Health Organization, 2011), https://www.who.int/teams/noncommunicable-diseases/sensory-functions-disability-and-rehabilitation/world-report-on-disability.

3. M. Carmona, T. Heath, T. Oc and S. Tiesdell, *Public Places Urban Spaces: The Dimensions of Urban Design* (Routledge, 2021).

4. K. Hara, J. Sun, R. Moore, D. Jacobs and J. Froehlich, "Tohme: Detecting Curb Ramps in Google Street View Using Crowdsourcing, Computer Vision and Machine Learning," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)* (Association for Computing Machinery, 2014), 189–204, https://doi.org/10.1145/2642918.2647403.

5. J. E. Froehlich, *Combining Crowdsourcing and Machine Learning to Collect Sidewalk Accessibility Data at Scale*. Technical report (University of Washington, 2021), www.pactrans.org.

6. G. Jocher, A. Chaurasia and J. Qiu, Ultralytics YOLOv8 (2023), https://github.com/ultralytics/ultralytics.

7. M. Duan, S. Kiami, L. Milandin, et al., "Scaling Crowd+AI Sidewalk Accessibility Assessments: Initial Experiments Examining Label Quality and Cross-City Training on Performance," in *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '22)* (Association for Computing Machinery, 2022), 1–5, https://doi.org/10.1145/3517428.3550381.

8. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 770–778, https://doi.org/10.1109/CVPR.2016.90.

9. M. Hosseini, A. Sevtsuk, F. Miranda, R. M. Cesar and C. T. Silva, "Mapping the Walk: A Scalable Computer Vision Approach for Generating Sidewalk Network Datasets From Aerial Imagery," *Computers, Environment and Urban Systems* 101 (2023): 101950, https://doi.org/10.1016/j.compenvurbsys.2023.101950.

10. O. F. Hamim, S. R. Kancharla and S. V. Ukkusuri, "Mapping Sidewalks on a Neighborhood Scale From Street View Images,"

Environment and Planning B: Urban Analytics and City Science 51, no. 4 (2024): 823–838, https://doi.org/10.1177/23998083231200445.

11. A. Tao, K. Sapra and B. Catanzaro, Hierarchical Multi-Scale Attention for Semantic Segmentation (2020), https://arxiv.org/abs/2005.10821.

12. K. Sun, Y. Zhao, B. Jiang, et al., High-Resolution Representations for Labeling Pixels and Regions (2019), https://arxiv.org/abs/1904.04514.

13. J. Wang, K. Sun, T. Cheng, et al., "Deep High-Resolution Representation Learning for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, no. 10 (2021): 3349–3364, https://doi.org/10.1109/TPAMI.2020.2983686.

14. Y. Yuan, X. Chen, X. Chen and J. Wang, Segmentation Transformer: Object-Contextual Representations for Semantic Segmentation (2021), https://arxiv.org/abs/1909.11065.

15. J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), 779–788, https://doi.org/10.1109/CVPR.2016.91.

16. F. Zhang, A. Salazar-Miranda, F. Duarte, et al., "Urban Visual Intelligence: Studying Cities With Artificial Intelligence and Street-Level Imagery," *Annals of the American Association of Geographers* 114, no. 5 (2024): 876–897, https://doi.org/10.1080/24694452.2024.2313515.

17. F. Biljecki and K. Ito, "Street View Imagery in Urban Analytics and GIS: A Review," *Landscape and Urban Planning* 215 (2021): 104217, https://doi.org/10.1016/j.landurbplan.2021.104217.

18. A. Abbott, A. Deshowitz, D. Murray, et al., "WalkNet: A Deep Learning Approach to Improving Sidewalk Quality and Accessibility," *SMU Data Science Review* 1 (2018), https://scholar.smu.edu/datasciencereview/vol1/iss1/7.

19. G. Weld, E. Jang, A. Li, A. Zeng, K. Heimerl and J. E. Froehlich, "Deep Learning for Automatically Detecting Sidewalk Accessibility Problems Using Streetscape Imagery," in *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19)* (Association for Computing Machinery, 2019), 196–209, https://doi.org/10.1145/3308561.3353798.

20. H. Ning, X. Ye, Z. Chen, T. Liu and T. Cao, "Sidewalk Extraction Using Aerial and Street View Images," *Environment and Planning B: Urban Analytics and City Science* 49, no. 1 (2022): 7–22, https://doi.org/10.1177/2399808321995817.

21. N. Blanc, Z. Liu, O. Ertz, et al., "Building a Crowdsourcing Based Disabled Pedestrian Level of Service Routing Application Using Computer Vision and Machine Learning," in *2019 16th IEEE Annual Consumer Communications and Networking Conference (CCNC)* (IEEE, 2019), 1–5.

22. B. W. Koo, S. Guhathakurta and N. Botchwey, "How Are Neighborhood and Street-Level Walkability Factors Associated With Walking Behaviors? A Big Data Approach Using Street View Images," *Environment and Behavior* 54, no. 1 (2022): 211–241, https://doi.org/10.1177/00139165211014609.

23. B. Kang, S. Lee and S. Zou, "Developing Sidewalk Inventory Data Using Street View Images," *Sensors* 21, no. 9 (2021): 3300, https://doi.org/10.3390/s21093300.

24. T. Gebru, J. Krause, Y. Wang, et al., "Using Deep Learning and Google Street View to Estimate the Demographic Makeup of Neighborhoods Across the United States," *Proceedings of the National Academy of Sciences* 114, no. 50 (2017): 13108–13113, https://doi.org/10.1073/pnas.1700035114.

25. J. C. Rangel, E. Cruz and M. Cazorla, "Automatic Understanding and Mapping of Regions in Cities Using Google Street View Images," *Applied Sciences* 12, no. 6 (2022): 2971, https://doi.org/10.3390/app12062971.

26. M. Saha, M. Saugstad, H. T. Maddali, et al., "Project Sidewalk: A Web-Based Crowdsourcing Tool for Collecting Sidewalk Accessibility Data at Scale," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)* (Association for Computing Machinery, 2019), 1–14, https://doi.org/10.1145/3290605.3300292.

27. Y. Zhang, S. Devalapalli, S. Mehta and A. Caspi, OASIS: Automated Assessment of Urban Pedestrian Paths at Scale (2023), https://arxiv.org/abs/2303.02287.

28. Mapillary, Mapillary: A Collaborative street-level Imagery Platform for Improving Maps (2024), https://www.mapillary.com.

29. M. Hosseini, M. Saugstad, F. Miranda, A. Sevtsuk, C. T. Silva and J. E. Froehlich, Towards Global-Scale Crowd+AI Techniques to Map and Assess Sidewalks for People With Disabilities (2022), https://arxiv.org/abs/2206.13677.

30. T. C. F. A. T. TCFAT, OpenSidewalks Project (2016), https://uwescience.github.io/DSSG2016-Sidewalks/.

31. T. C. F. A. T. TCFAT, AccessMap Project (2019), https://accessmap.net/.

32. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification With Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, Vol. 25, (2012), 1–9.

33. Amsterdam Intelligence AI, Project Sidewalk Amsterdam (2021), https://sidewalk-amsterdam-test.cs.washington.edu/.

34. M. J. Choi, D. G. Ku and S. J. Lee, "Integrated YOLO and CNN Algorithms for Evaluating Degree of Walkway Breakage," *KSCE Journal of Civil Engineering* 26, no. 8 (2022): 3570–3577, https://doi.org/10.1007/s12205-022-1017-1.

35. F.-E. Wang, H.-N. Hu, H.-T. Cheng, et al., "Self-supervised Learning of Depth and Camera Motion from 360° Videos," in *Computer Vision – ACCV 2018. ACCV 2018*, ed. C. Jawahar, H. Li, G. Mori, and K. Schindler, Vol. 11365 (Springer, 2019), https://doi.org/10.1007/978-3-030-20873-8_4.

36. T.-Y. Lin, M. Maire, S. Belongie, et al., "Microsoft COCO: Common Objects in Context," In *Computer Vision – ECCV 2014. ECCV 2014*, ed. D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Vol 8693 (Springer, 2014), https://doi.org/10.1007/978-3-319-10602-1_48.

37. L. Yang, B. Kang, Z. Huang, et al., Depth Anything V2 (2024), https://arxiv.org/abs/2406.09414.

38. OpenStreetMap Contributors, Overpass API (2024), https://wiki.openstreetmap.org/wiki/Overpass_API.

39. OpenStreetMap Contributors, OpenStreetMap API (2024), https://wiki.openstreetmap.org/wiki/API.

40. M. T. Metten and Contributors, osmapi: Python Wrapper for the OSM API (2024), https://github.com/metaodi/osmapi.