2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

SideGuide: A Large-scale Sidewalk Dataset for Guiding Impaired People

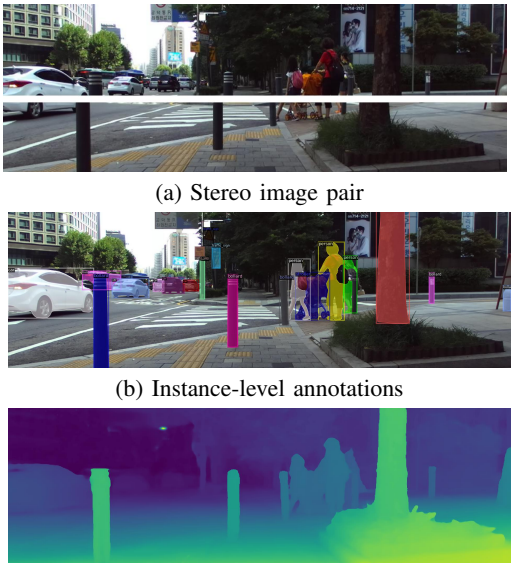
Kibaek Park[1†], Youngtaek Oh[1†], Soomin Ham[1†], Kyungdon Joo[2†],
Hyokyoung Kim[3], Hyoyoung Kum[3], and In So Kweon[1*]

***Abstract*— In this paper, we introduce a new large-scale sidewalk dataset called *SideGuide* that could potentially help impaired people. Unlike most previous datasets, which are focused on road environments, we paid attention to sidewalks, where understanding the environment could provide the potential for improved walking of humans, especially impaired people. Concretely, we interviewed impaired people and carefully selected target objects from the interviewees' feedback (objects they encounter on sidewalks). We then acquired two different types of data: *crowd-sourced data* and *stereo data*. We labeled target objects at instance-level (*i.e.*, bounding box and polygon mask) and generated a ground-truth disparity map for the stereo data. SideGuide consists of 350K images with bounding box annotation, 100K images with a polygon mask, and 180K stereo pairs with the ground-truth disparity. We analyzed our dataset by performing baseline analysis for object detection, instance segmentation, and stereo matching tasks. In addition, we developed a prototype that recognizes the target objects and measures distances, which could potentially assist people with disabilities. The prototype suggests the possibility of practical application of our dataset in real life.**

I. Introduction

Despite significant advances in deep learning, there still exists a gap in the knowledge needed to exploit this technology in real-world situations. One of the reasons for this is insufficient data. This is because it is hard to reflect real-world environments thoroughly using data with a limited distribution. To minimize the gap between the technology in research and that in practical use in real life, we need to explore the environments around us.

To date, autonomous driving has been one of the most researched topics. Previous studies have largely been focused on self-driving cars and their driving environment, which has led to ample datasets related to roads (*e.g.*, KITTI [1], Cityscapes [2]). In contrast, there has not been enough data investigating the perspectives of pedestrians and their environments, such as sidewalks. Traditionally for cars, fixed lanes separate cars from other vehicles, which makes it easier to detect moving objects. On the other hand, when it comes to a sidewalk, there are no fixed lanes, and there are lots of objects (*e.g.*, pedestrians, personal mobility vehicles, animals, and bollards) that lack directional consistency. Therefore, in the sidewalk environment, objects often occlude each other (*i.e.*, partially blocked by other objects).



(a) Stereo image pair

(b) Instance-level annotations

(c) Ground-truth dense disparity map

Fig. 1: **An example image-pair and annotation of a sidewalk**. (a) Stereo image pair captured by a ZED stereo camera (top and bottom images indicate left and right images, respectively), (b) Instance-level bounding box and polygon mask annotations in the left image, and (c) The ground-truth dense disparity map.

It is often thought that autonomous driving applications will make our ways of life much more convenient; however, not all people are going to be able to benefit from this technology. For instance, it is not as straightforward for the visually impaired to have access to personal vehicles because it is hard for them to react to potential dangers they may face while driving. Ideally, the technology should be able to accommodate all forms of disabilities, including visual impairments. Although it is important to have technical advances to improve everyday lifestyles for the convenience of the general public, it is also desirable to fulfill one's essential needs, such as safety rights.

Our work is the first large-scale dataset (termed the *SideGuide*) focused on sidewalks, and which can be deployed in the field of recognition to aid impaired or disabled people. We also used a stereo camera system to capture images of sidewalks, which provides ground-truth disparity as well as instance-level annotation (see Fig. 1).

The SideGuide has been generated through data acquired

Acknowledgment This work was supported by the National Information Society Agency for construction of training data for artificial intelligence (2100-2131-305-107-19). We thank *SelectStar, Inc.* and *Korea Spinal Cord Injury Association (KSCIA)* for their support.

[1]K. Park, Y. Oh, S. Ham, and I. S. Kweon are with the School of Electrical Engineering, KAIST, Daejeon, South Korea. {parkkibaek, youngtaek.oh, smham, iskweon77}@kaist.ac.kr

[2]K. Joo is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, USA. This work was done when K. Joo was at KAIST. kjoo@andrew.cmu.edu

[3]H. Kim and H. Kum are with TestWorks, Inc., Seoul, South Korea. {hk.kim, hy.kum}@testworks.co.kr

[†]The first four authors contributed equally to this work.

[*]Corresponding author.

TABLE I: **Comprehensive analysis of related datasets**. Note that all quantities except for *classes* denote the number of images. i.e., *100K in the mask field means 100K images are annotated with mask annotations throughout all the images in the dataset.*

Datasets	Total images	Object annotation			Semantic annotation		Depths		Scenes
		boxes	mask	classes	imgs	classes	num	type	
KITTI [1]	15K	15K	–	8	400	12	400[1]	LiDAR	Driving
Cityscapes [2]	25K	25K	25K	30	25K	30	25K	Stereo	Driving
ApolloScape [3]	147K	–	–	–	147K	25	51K	LiDAR	Driving
Mapillary Vistas [4]	25K	–	25K	37	25K	29	–	–	Street[2]
SOID [5]	50K	–	–	5	–	–	–	–	Sidewalk
Ours	312K+180K[3]	350K	100K	29	–	–	180K	Stereo	Sidewalk

[1] KITTI Stereo 2015 Benchmark.
[2] *Street* scene includes both driving scenes and sidewalk environments.
[3] Our dataset consists of 312K crowd-sourced images and 180K stereo image pairs with a disparity map captured by impaired people. Details (including statistics and the split) are explained in Sec. IV.

from both ZED and smartphone cameras. The former collected a pair of stereo images at a time to estimate the depth of objects. The latter obtained a single image for object recognition and collected a considerable number of images from a crowd-sourcing platform. From the raw data, we annotated 492K images in total for ground-truth (see Table I). Specifically, as instance-level annotation, we generated ground-truth bounding boxes (BB) for 350K images and polygon segmentations for 100K images as a subset of BB. We also generated 180K dense disparity maps from pairs of stereo images. The instance-level annotations and ground-truth disparity provided inference data to the detection model and stereo matching algorithm, respectively, to validate our dataset. The SideGuide was further validated by implementing a prototype that returns the output of object detection and distance of an object from the camera in real-time.

Our first large-scale sidewalk dataset, the SideGuide will contribute to reducing the gap between technology and real-world deployment for recognition. This dataset could be useful not only for impaired people, but also for other applications like mobile robotics and assistance for vulnerable road users who are not necessarily visually or mobility impaired.

II. Related Work

The use of large-scale labeled datasets is one of the key components contributing to the success of deep learning. A variety of large-scale datasets [6], [7], [8], [9], [10], [11], [12], [13] have enabled us to research fundamental problems and to improve their performance in most computer vision tasks. For example, ImageNet [6] provides 1M labeled images based on 1K categories, and a deep neural network model [14] trained on ImageNet, achieved human-level classification performance of object recognition tasks. In addition, thanks to a variety of public datasets, many computer vision applications are now available within the framework of deep learning. This is because its use has advanced to include a number of huge industrial activities such as autonomous driving, healthcare, and chatbots.

In previous work, the datasets needed to understand the environment surrounding us drew a great deal of interest. In particular, various types of real-world datasets about road environments have been released and now contribute to autonomous driving-related research (such as CamVid [15], Daimler Urban Segmentation [16], KITTI [1], Cityscapes [2], Mapillary [4], and ApolloScape [3]). As a representative dataset, the KITTI is known for its various benchmarks such as stereo, flow, and 2D/3D object detection. However, the number of object and segmentation annotations is relatively small compared to the raw data, where 7K training images are annotated by bounding boxes in the object detection benchmark, and 200 training images are annotated at pixel-level in the semantic segmentation benchmark. The Cityscapes follows up on the KITTI by accumulating more segmentation annotations. It consists of 20K and 5K frames with coarse and fine-detailed class annotation, respectively, and SGM stereo disparity [17] (see Table I).

Unlike for road environments, there has been a surprisingly small number of datasets considering sidewalk environments [4], [5], [18]. Mapillary Vistas [4] is focused on segmentation tasks in street scenes, which includes both driving scenes and sidewalk environments. It contains 25K images with fine annotations in 66 categories. Ahmed *et al*. [5] released a pilot dataset about sidewalks (called the Sidewalk Obstacle Image Dataset: SOID), which contains 50K images for image classification with 5 categories. Although SOID was captured in a sidewalk environment, its only feasible task is limited to simple classification. It is not appropriate for object detection, segmentation, and depth perception tasks, all of which are necessary for autonomous or assistance systems. On the other hand, the proposed dataset contains various types of annotations for the sidewalk environment. To the best of our knowledge, we introduce here the first large-scale sidewalk dataset that provides instance-level object annotations (bounding box and polygon segmentation), as well as ground-truth depth. We believe that this dataset will contribute to making it easier for everybody to walk, including impaired people, on sidewalks.

III. Data Generation

In this section, we provide a detailed process for the generation of sidewalk data. When we designed this dataset, we considered two essential factors that would make our dataset valuable: practicality and diversity. First, the dataset should reflect the perspective of the target population for practical usage (*i.e*., in this case, people who have impaired (limited) mobility or sight). Second, the dataset has to include diverse objects on the sidewalk, and the target objects labeled should

TABLE II: **Object categories of the sidewalk dataset**.

Dynamic (13)	bicycle, bus, car, carrier, cat, dog, motorcycle, movable signage, person, scooter, stroller, truck, wheelchair
Static (16)	barricade, bench, bollard, chair, fire hydrant, kiosk, parking meter, pole, potted plant, power controller, stop, table, traffic light, traffic light controller, traffic sign, tree trunk

Fig. 2: **Example of distinct objects**. Such objects are often encountered on sidewalks and could be obstacles for impaired people.

be well distributed across all the data. Considering these two factors, we used two strategies for collection of data about sidewalk environments. Moreover, the acquired images were labeled in several ways: instance-level annotation (bounding box and polygon mask) and disparity.

A. Object Categories

Depending on the existence of mobility, object categories were divided into two super-categories: dynamic objects and static objects. We determined the detailed object categories from several interviews conducted with impaired people.

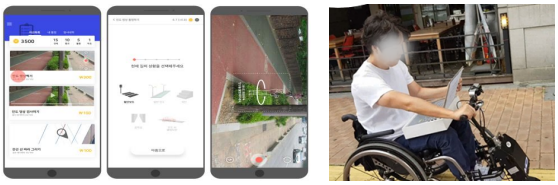
The list of object classes is shown in Table II. Note that unlike other datasets focusing on autonomous driving applications, our dataset contains diverse object categories that can easily be spotted while walking on a sidewalk. Visual examples of some of the classes are shown in Fig. 2.

B. Image Acquisition

To collect data considering the above-mentioned two factors (*i.e.*, practicality and diversity), we captured sidewalk scene images by two distinct methods. First, we collected a set of single images using a crowd-sourcing platform in mobile phones. We denoted these images as *crowd-sourced data*. Second, we collaborated with impaired people for data collection. Impaired people directly captured stereo image pairs using a stereo camera system installed on wheelchairs (denoted as *stereo data*). This acquisition approach enabled us to capture stereo data from their viewpoints.

1) Crowd-sourced Data: To collect a set of diverse images on the sidewalk, we used a mission-based crowd-sourcing platform developed by SelectStar[1] (see Fig. 3a). This platform automatically sets the ratio of the captured image to 16 : 9 and resizes its resolution to 1920×1080 regardless of the smartphone models (Android only). Using this platform, we provided guidelines that explain how to capture a scene adequately (*e.g.*, appropriate viewpoint and the required number of objects). We also supported a UI/UX

[1] https://www.selectstar.ai/



(a) (b)

Fig. 3: **Two different data acquisition methods**: (a) Mobile collection using a crowd-sourcing platform, and (b) Direct capture with stereo camera by impaired people.

platform so that the users could easily capture images satisfying our guidelines. In addition, only verified users who passed a simple capture test could participate in the data collection to assure image quality. Using this systematic platform, we collected diverse image data captured from various locations, in various environments, and at different times of the day.

2) Stereo Data: To acquire stereo data composed of rectified left and right images (a stereo pair), we used a commercial stereo camera system, the ZED camera. We installed the ZED camera on a wheelchair using a 3-axis gimbal for stabilization. We located the camera to provide the viewpoint of impaired people. Before capturing the data, we performed a calibration process for image rectification using the ZED calibration toolbox[2] and set its resolution to 1920×1080. Using this capture system, people with spinal cord injuries[3] went around the sidewalk area at different times of the day and in multiple cities (Seoul, Daejeon, *etc.*) of South Korea to acquire stereo data. We captured a stereo pair at 1 frame-per-sec using the macro (see Fig. 3b).

C. Ground-truth Annotation

We acquired both stereo and crowd-sourced data on public sidewalks. This raw data contains noisy images, such as blurry and oversaturated images, and includes personal information such as human faces and license plate. As preprocessing, we first used reviewers to filter noisy data out manually. We then anonymized identifiable information to protect the privacy of the personal information. Specifically, we detected faces and license plates using fine-tuned Faster-RCNN [19] and then blurred the detected regions using the Gaussian blur function in OpenCV. The de-identified images were double-checked by reviewers manually.

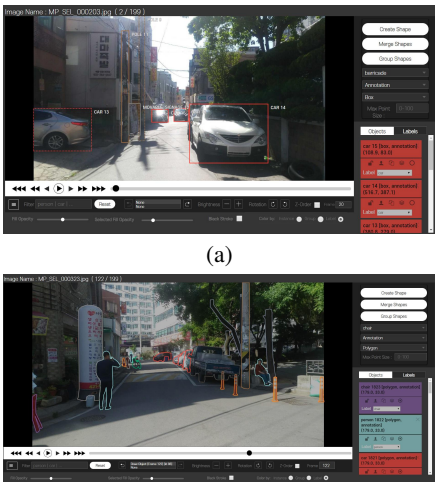
1) Instance-level Annotation: To annotate the acquired dataset, we developed an annotation tool called Aiworks Annotation Tool (AAT), as shown in Fig. 4. AAT builds upon a free annotation tool for computer vision (the Computer Vision Annotation Tool[4]) and was slightly modified for our task (*e.g.*, text was translated into Korean for the workers).

Using AAT, we annotated bounding box (BB) and polygon segmentation (PS) of target objects at an instance level (see Fig. 5). Note that, to assure the quality of our instance-level annotation, the annotated data were carefully reviewed

[2] https://www.stereolabs.com/developers/release/

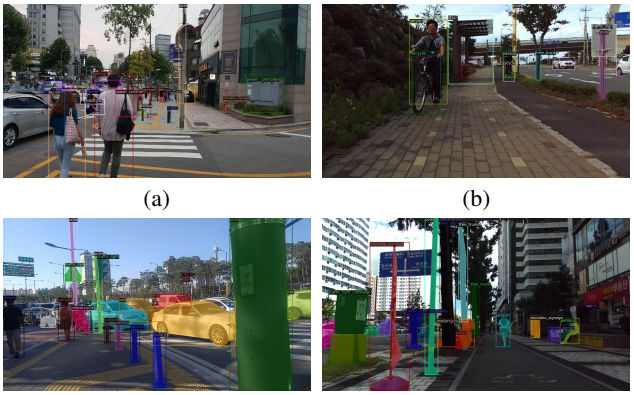
[3] People belonging to the Korea Spinal Cord Injury Association (KSCIA) participated in the stereo data capture.

[4] https://github.com/opencv/cvat



(a) (b)

Fig. 4: **Illustration of our annotation toolbox**: (a) Bounding box and (b) Polygon mask toolboxes.



(a) (b) (c) (d)

Fig. 5: **Sampled instance-level annotated images in our SideGuide dataset**. (a,b) Bounding box annotations on crowd-sourced and stereo data, respectively. (c,d) Polygon mask annotations on crowd-sourced and stereo data, respectively. Additional annotation examples are available in the supplementary material.

by three different people in a hierarchical manner: 1) by a worker after annotation, 2) by a reviewer (checking a set of data), and 3) by a project manager (cross-checking).

Bounding Box Annotation. We annotated the bounding box of object instances for 350K images. We put 22 workers, 8 reviewers, and 2 project managers (PM) on the BB annotation task. The PMs provided a guideline for the BB annotation and handled the overall schedule (please refer to the detailed annotation guideline in the supplementary material). Following the guideline, each worker annotated the target object as a bounding box by clicking two points (left-top and right-bottom points) that tightly circumscribe the object. We saved the annotated objects in files following the XML format (<meta>+<image>). On average, each worker annotated 150 images (4.5K object instances) per day.

Polygon Segmentation Annotation. Unlike with BB annotation, labeling the pixel-level instance mask is a time-consuming task in that it should tightly circumscribe the object without overlapping others. We put 20 workers, 5 reviewers, and 2 PMs on the PS annotation task. We annotated PS mask for 100K images, a subset of the 350K BB annotated images. Similar to the BB annotation, the PMs provided a guideline for the PS annotation and handled the overall schedule (please refer to the detailed annotation guideline in the supplementary material). Following the guideline, each worker annotated an average of 70 images (490 object instances on average) per day. We saved the PS annotation in XML file format and reviewed the quality.

2) Ground-truth Disparity: We computed the ground-truth disparity of stereo images using a state-of-the-art stereo algorithm, GANet [20]. After computing disparity by GANet, we manually validated the estimated ground-truth from several perspectives.

Concretely, we first removed the inappropriate pair of images, *e.g.*, images saturated under direct sunlight or flashing car headlights, or images shaken by bumps. Second, we eliminated improper disparity using the photometric-consistency. Using intensity difference between stereo images as a confidence measurement, we generated confidence images and then, reliable disparity images was chosen if the number of confident pixels was higher than a specific threshold (empirically defined). Last, we visually checked reliable disparity map using the confidence image. We made a tool to blend disparity and confidence images, inspecting sparse disparity errors that have low confidence values to compensate for the possibility of adopting a wrong inlier in the previous steps. Most disparity errors occurred in the bleeding of thin objects (*e.g.*, wires) and the light reflection from surfaces (*e.g.*, building windows). For this process, we used 4 reviewers and 1 project manager, and each reviewer checked an average of 1.6K disparity images per day.

Consequently, we generated 180K reliable ground-truth stereo images composed of left, right, disparity and disparity confidence pairs with cropped resolution of 1920×592 to adjust the resolution ratio of KITTI due to the pre-trained model. For disparity and disparity confidence data, a user can adjust the confidence level to get more reliable ground-truth. This can be done by applying a certain disparity confidence threshold to the generated disparity, which is stored in 16-bit data (see sampled ground-truth disparity images in Fig. 6).

IV. Dataset Analysis

A. Dataset Summary

Our sidewalk dataset consists of 492K images (see Table I) that are split into four subsets according to the level of ground-truth annotations (*c.f.*, Sec. III-C). For 350K images—most of them from crowd-sourced data and some from stereo data—we performed box-level annotations for our object categories and called these 350K images with BB annotation the *bounding box set*. With 100K images (a subset of bounding box set), we performed PS annotation for our object categories and called these the *polygon set*. From stereo data, we generated a dense disparity map for 180K stereo image pairs. We denoted these stereo pairs as the *disparity set*. In particular, 38K stereo image pairs contained all the annotation modalities (*i.e.*, bounding box,

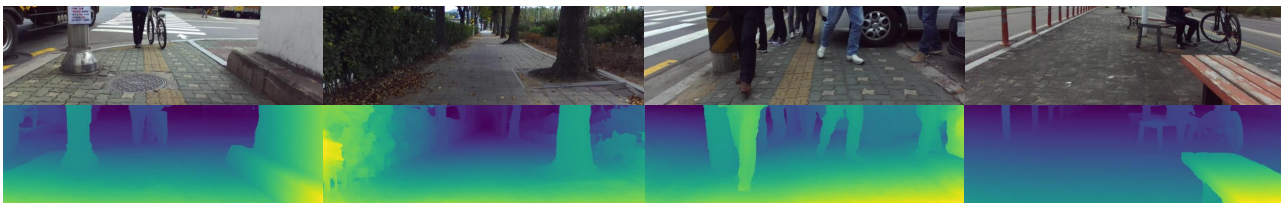


Fig. 6: **Sampled ground-truth disparity images in the SideGuide dataset**: The first and second rows indicate left images and the corresponding ground-truth depth maps, respectively. Additional samples are available in the supplementary material.

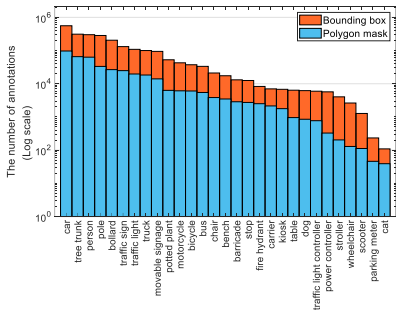
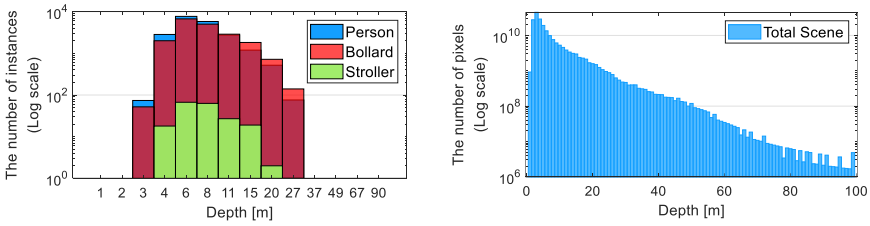


Fig. 7: **Histogram of class-wise annotation count in log scale**.



(a) Distance statistics for each object

(b) Distance statistics for total scene

Fig. 8: **Distance statistics in the SideGuide dataset**: (a) The number of object instances, containing *Person*, *Bollard*, and *Stroller*, are computed according to the depth in meters. (b) The number-of-pixels distribution on the disparity set with regard to distances.

polygon mask, and disparity). We denoted this set of images as the *intersection set*, which can be used to release a domain gap between crowd-sourced data and stereo data. Using the intersection set, we validated the practical usage of our dataset in Sec. VI.

Table III shows summaries of the annotated images and instances with respect to dataset splits. For data release, we split each subset into training, validation, and test sets. Specifically, we randomly split the bounding box set and polygon set into training, validation, and test images with a ratio of $\frac{7}{10}$ (training), $\frac{2}{10}$ (validation), and $\frac{1}{10}$ (test), respectively. In case of disparity set, we split into training and test sets with a ratio of $\frac{17}{20}$ (training) and $\frac{3}{20}$ (test). We also labeled a total of 3.3M bounding boxes and 588K polygon segmentations of our target objects.

B. Statistics

To understand better our dataset from various perspectives, we analyzed our dataset using several statistics. Figure 7 shows the per class number of instances in the bounding box set and polygon set. Because our dataset targets sidewalk environments, uncommon classes in the road scene such as *bollard*, *table*, and *stroller* are labeled and well distributed in our dataset (see Fig. 2).

In addition, we inspected the relationship between the class categories and the depth to see how the distance in each class was distributed in our dataset. Distance statistics of *person*, *bollard*, and *stroller* in the *intersection set* are illustrated in Fig. 8a. We generated masked disparity with corresponding polygon annotation and average to mean depth. This shows that the *person* class is frequently located at 4–15 meters in our dataset and that *stroller* class rarely appeared at more than 15 meters. This is because we mostly annotated the close obstacles as safety issues. We also inspected the number-of-pixels distribution with regard to depth variation in the disparity set, as shown in Fig. 8b.

TABLE III: **Summary of dataset splits.** For bounding box set and polygon set, we also provide number of annotated instances.

		Training	Validation	Test
Bounding box set	#images	249K	37K	67K
	#boxes	2,379K	351K	652K
	#avg box per image		9.6	
Polygon set	#images	70K	10K	20K
	#masks	409K	61K	118K
	#avg mask per image		5.9	
Disparity set	#images	147K	-	33K

V. Algorithmic Analysis

In this section, we provide experimental results for object detection, instance segmentation, and stereo matching tasks to validate our SideGuide dataset. First, we briefly explain background knowledge for each task and the experimental setup. Then, we report and analyze the baseline results.

A. Task and Baseline Networks

Object Detection. Object detection is a task that localizes an object within a bounding box belonging to a specific class.

We selected widely used detection models from both one-stage and two-stage models. YOLO [21] is a representative one-stage model devised to obtain improved accuracy under real-time computational complexity. After this real-time model, many one-stage models were published. For example, RetinaNet [22] aims to resolve imbalance between foreground and background anchors by paying more attention to less confident samples, and showed promising results. As a pioneering two-stage network, Faster R-CNN [19] predicts initial region proposals and refines them to achieve exact object bounding boxes.

Instance Polygon Segmentation. Instance segmentation aims to recognize objects in a pixel-level mask rather than in a coarse bounding box. Mask R-CNN [23] is one of the most successful *detect then segment* methods. The network

TABLE IV: **Baseline model evaluation results**. Note that R-50-FPN is the abbreviation of the ResNet-50 based FPN [28] backbone.

(a) Evaluation results for object detection

Model	Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
Yolo v3 [21]	DarkNet	31.9	56.3	32.4	2.1	20.3	39.6
RetinaNet [22]	R-50-FPN	47.4	70.7	52.6	6.5	36.8	53.6
RetinaNet [22]	R-101-FPN	49.1	72.3	54.8	7.0	38.1	55.6
Faster RCNN [19]	R-50-FPN	50.6	73.4	57.5	10.3	40.5	56.0
Faster RCNN [19]	R-101-FPN	52.3	74.8	59.5	10.5	41.6	58.1

(b) Evaluation results for instance segmentation

Model	Backbone	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
TensorMask [24]	R-50-FPN	37.3	58.6	39.8	9.0	31.5	47.4
TensorMask [24]	R-101-FPN	38.6	59.4	41.3	8.9	32.8	49.2
Mask RCNN [23]	R-50-FPN	41.1	62.5	44.8	11.8	36.1	49.0
Mask RCNN [23]	R-101-FPN	42.1	63.4	45.9	11.5	36.9	50.9

leverages Faster R-CNN to generate bounding boxes, then an extra convolutional branch predicts pixel-level masks within the bounding boxes. Recently, a dense sliding window based instance segmentation method called TensorMask [24] was proposed. Chen *et al*. [24] exploited structured 4D tensor representations for masks in a set of densely sliding windows. With those representations, TensorMask predicts dense masks over a spatial domain.

Stereo Matching. The final task for validation of our dataset is stereo–matching disparity estimation. By inferring disparity (*i.e*., the inverse of depth), we could understand a depth map of a given sidewalk scene. That is, we could determine the distances of observed objects from the current view, which could help people walking on the sidewalk. For this purpose, we used our disparity set to train several state-of-the-art stereo matching algorithms. Among the various approaches to stereo matching, we evaluated three representative algorithms: HSMNet [25], PSMNet [26], and GwcNet [27]. HSMNet achieves high-resolution disparity in a coarse-to-fine manner in real-time. PSMNet is an end-to-end network embedding global context with dilated convolution in the framework of pyramid pooling network. GwcNet constructs a group-wise correlation cost volume to achieve effective feature representation.

B. Dataset and Evaluation Metrics

For object detection, we performed experiments on the bounding box set. As mentioned in Table III, we utilized 249K images from *train split* for training, and reported the results using 67K images from *test split*. When evaluating the instance segmentation models, 70K of *train split* images from the polygon set were used. In the stereo matching task, we trained using 147K images in the training set and 33K in the test set of the disparity set.

We adopted the standard COCO-style evaluation metric Average Precision (AP) [12]. With this metric, The mean AP is averaged APs across IoU thresholds from 0.5 to 0.95 with an interval of 0.05. In addition, the AP at different IoU thresholds (AP_{50}, AP_{75}) and the AP at different scales (AP_S, AP_M, AP_L) are also reported. We reported the box AP for object detection and mask AP for instance segmentation.

To assess ground-truth reliability in the stereo matching task, we utilized some standard stereo matching metrics [25]: bad-2.0, bad-1.0, and RMSE, where the lower the metric the better the performance.

C. Implementation Details

For RetinaNet, Faster R-CNN, Mask R-CNN, and TensorMask, we followed the standard COCO training schedule implemented in the detectron2 framework [29]. The models were initialized from ImageNet pre-trained weights. We trained the models for a total of 90K iterations with 8 Titan Xp GPUs. The initial learning rate started from 0.2 with linear warm-up of 1K iterations, and was decreased by a factor of 10 after 60K and 80K iterations, respectively. SGD with momentum 0.9 was used as an optimizer. The input images were resized to have 800 and 1333 pixels for short edge and long edge, respectively, without changing the aspect ratio.

For training YOLO [21], the input images were resized to the resolution of 416×416. The mini-batch size was set to 64 with 8 GPUs. We trained using 90K iterations with an initial learning rate of 0.002, and we downscaled after 60K and 80K iterations. Unless mentioned specifically, other options were kept the same as the default COCO training configurations.

For a fair comparison of the stereo matching, we used the code released by the authors and followed the default settings and training details of each method [25], [26], [27]. We conducted the evaluation with and without fine-tuning, utilizing the pre-trained weight. We trained the models with 8 Titan Xp GPUs, 192 maximum disparity, and a SGD optimizer with momentum 0.9, applying each learning rate as defined by the authors.

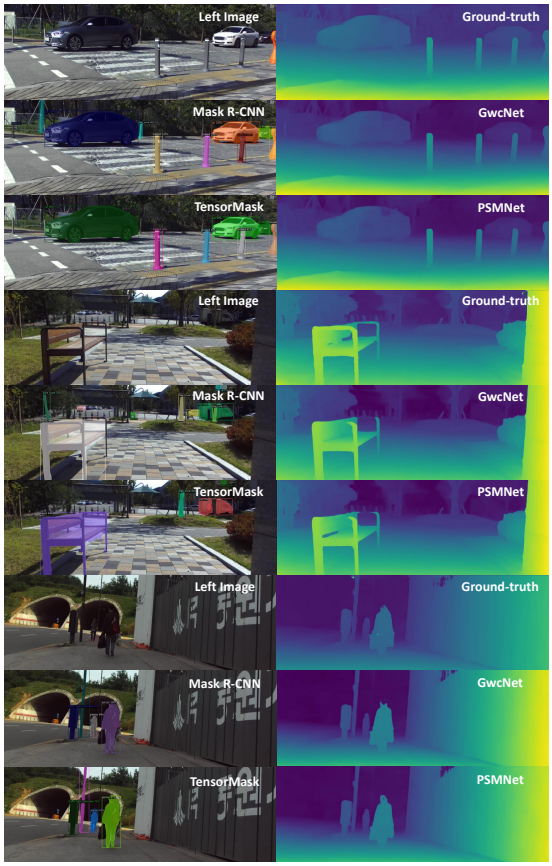
D. Experimental Results

We report our results on different backbones and different frameworks for both object detection and instance segmentation task, in Table IV. The results show that the AP performance consistently improves with deeper backbones. In Table V, we report class-specific AP as evaluated using Faster R-CNN and RetinaNet for object detection, and Mask R-CNN and TensorMask for instance segmentation. Note that all models were trained with R-50-FPN backbones. Figure 9a visualizes the predicted bounding boxes of objects from Faster R-CNN *(left)* and RetinaNet *(right)*. The instance segmentation results correspond to Fig. 9b. The left part and right part shows the segmentation results from Mask R-CNN and TensorMask, respectively. The backbone networks were fixed as R-50-FPN for all detection and segmentation models.

In addition, the results from all the stereo matching methods showed improved performance after the fine-tuning process in Table VI. Concretely, large margins of error were reduced in the metrics bad-2.0 and bad-1.0, indicating that outliers were removed through the learning process. In particular, HSMNet shows that the outliers have been substantially reduced by more than 50% through training on our dataset. In Fig. 9c, qualitative stereo matching results of test images in the *intersection set* are represented. For example, the thin part of a bench, which is error prone for disparity estimation, is well predicted in ground-truth, but

(a) Object detection visual results

(b) Instance segmentation visual results



(c) Predictions on *intersection set*

Fig. 9: **Qualitative results:** (a) Detection results for Faster R-CNN (left) and RetinaNet (right). (b) Instance segmentation results for Mask R-CNN (left) and TensorMask (right). (c) Instance segmentation and stereo matching results from the test set of the *intersection set*.

TABLE V: **Class-specific evaluation results in object detection and instance segmentation.** Class-wise AP was computed using the baseline models. We used common backbone networks as ResNet-50 based FPN [28].

Model	car	tree trunk	person	pole	bollard	traffic sign	traffic light	truck	*ms**	potted plant	motorcycle	bicycle	bus	chair	bench
RetinaNet [22]	74.7	46.8	61.0	49.2	51.7	61.0	51.2	67.2	57.1	39.7	58.7	55.8	66.2	43.0	38.1
Faster RCNN [19]	78.2	53.8	64.8	63.2	57.7	65.1	58.2	69.3	60.8	42.1	60.0	57.7	68.4	46.3	41.4
TensorMask [24]	80.9	49.6	56.1	54.1	53.0	61.1	48.6	66.1	54.2	31.0	49.8	35.6	66.8	20.4	26.8
Mask RCNN [23]	82.3	56.3	61.4	65.0	61.3	65.1	57.2	67.7	58.3	35.2	52.4	38.9	70.3	21.9	28.0
	barricade	**stop**	**fire hydrant**	**carrier**	**kiosk**	**table**	**dog**	***tlc****	***pc****	**stroller**	**wheelchair**	**scooter**	**parking meter**	**cat**	
RetinaNet [22]	42.8	52.8	51.3	28.7	41.9	32.5	79.4	26.7	30.9	64.6	73.9	27.4	0.1	0.4	
Faster RCNN [19]	45.8	52.1	53.0	30.4	42.0	37.0	79.2	30.4	34.3	68.2	77.1	30.5	0.0	0.0	
TensorMask [24]	31.1	49.4	43.6	12.1	32.6	12.4	1.0	44.5	51.0	17.2	9.9	2.6	1.1	11.1	
Mask RCNN [23]	33.8	52.4	46.9	14.7	38.7	13.6	20.2	49.1	53.7	30.0	16.0	0.1	0.0	0.0	

* Note that movable signage (*ms*), traffic light controller (*tlc*) and power controller (*pc*).

TABLE VI: **Comparison of stereo matching accuracy.** The meaning of 'bad-1.0' and 'bad-2.0' is percentage of bad pixels whose error is greater than 1.0 pixel and 2.0 pixel respectively. Lower metric is better.

Method	RMSE		bad-2.0		bad-1.0	
	w/o FT	w/ FT	w/o FT	w/ FT	w/o FT	w/ FT
HSM [25]	5.55 px	3.60 px	15.16 %	5.16 %	38.83 %	14.23 %
PSMNet [26]	5.85 px	4.32 px	13.17 %	5.88 %	41.18 %	12.90 %
GwcNet [30]	4.33 px	3.34 px	9.23 %	4.81 %	22.74 %	12.58 %

has relatively low quality in PSMNet and GwcNet. In rear the window part of the first car, for which it is also hard to predict disparity, PSMNet estimated a low-quality result rather than ground-truth.

VI. Application

As an additional way to validate SideGuide, we developed a prototype trained on SideGuide which could be used as an assistance system for impaired people by informing obstacles with distances. The prototype recognizes target objects on the sidewalk, and predicts the distance for each object and the average distance within the object region. More precisely, the prototype consists of two neural networks that perform object detection and stereo matching at the same time. We trained YOLO (object detection) and PSMNet (stereo matching) on our bounding box and disparity sets, respectively, and then fine-tuned both networks on the *intersection set*.

To assess the predictability of unseen data, we conducted

Fig. 10: **Demonstration of proposed application.** The image taken from impaired person on a wheelchair, where many safety-related obstacles are scattered on the way. Each bounding box is marked with class label and distance.

experiments on real-world scenarios. We directly used image sequences taken by a ZED camera installed on a wheelchair (*i.e.*, these does not exist in our training dataset). As shown in Fig. 10, impaired people often encounter difficulties due to the many safety-related obstacles along the pathways people travel. However, our network detects multiple obstacles and predicts the distance to each obstacle in real-time ($\approx 25fps$). Therefore, by using our dataset, we could resolve challenges that occur often while walking on sidewalks.

We provide a supplementary video that includes an application demo as well as some visualization results, along with annotation processing and guidelines.[5]

VII. Conclusion

In this paper, we introduced a new large-scale sidewalk dataset for guiding impaired people. By utilizing a crowdsourcing platform and collaborating with impaired people, we collected diverse images on sidewalks. These included many distinct objects (*e.g.*, strollers and bollards) and reflected their perspective. Our sidewalk dataset consists of 492K images with different types of ground-truth annotations: bounding box, polygon mask, and disparity. We analyzed our dataset in relation to several tasks via statistics and training using state-of-the-art methods. We believe that this meaningful dataset on sidewalks will encourage various computer vision applications, as we demonstrated via the prototype, and further contribute to making it easier to walk on sidewalks, especially for impaired people.

References

[1] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision Meets Robotics: The KITTI Dataset," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[3] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The ApolloScape Dataset for Autonomous Driving," *arXiv: 1803.06184*, 2018.

[4] G. Neuhold, T. Ollmann, S. R. Bulò, and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," in *IEEE International Conference on Computer Vision*, 2017.

[5] F. Ahmed and M. Yeasin, "Optimization and Evaluation of Deep Architectures for Ambient Awareness on a Sidewalk," in *IEEE International Joint Conference on Neural Networks*, 2017.

[5] https://youtu.be/qhoMECrumUw

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A Large-scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[7] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[8] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic Understanding of Scenes through the ADE20K Dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2019.

[9] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," in *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

[10] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The Role of Context for Object Detection and Semantic Segmentation in the Wild," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[11] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8M: A Large-scale Video Classification Benchmark," *arXiv:1609.08675*, 2016.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision*, 2014.

[13] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ Questions for Machine Comprehension of Text," *arXiv:1606.05250*, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification," in *IEEE International Conference on Computer Vision*, 2015.

[15] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and Recognition using Structure from Motion Point Clouds," in *European Conference on Computer Vision*, 2008.

[16] T. Scharwächter, M. Enzweiler, U. Franke, and S. Roth, "Efficient multi-cue scene segmentation," in *German Conference on Pattern Recognition*, 2013.

[17] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, p. 328–341, 2008.

[18] K. Yang, L. M. Bergasa, E. Romera, and K. Wang, "Robustifying semantic cognition of traversability across wearable rgb-depth cameras," *Applied optics*, vol. 58, no. 12, pp. 3141–3155, 2019.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, 2015.

[20] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "GA-Net: Guided Aggregation Net for End-to-end Stereo Matching," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[21] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv*, 2018.

[22] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *IEEE International Conference on Computer Vision*, 2017.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *IEEE International Conference on Computer Vision*, 2017.

[24] X. Chen, R. Girshick, K. He, and P. Dollar, "TensorMask: A Foundation for Dense Object Segmentation," in *IEEE International Conference on Computer Vision*, October 2019.

[25] G. Yang, J. Manela, M. Happold, and D. Ramanan, "Hierarchical Deep Stereo Matching on High-resolution Images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[26] J.-R. Chang and Y.-S. Chen, "Pyramid Stereo Matching Network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[27] X. Cheng, P. Wang, and R. Yang, "Learning Depth with Convolutional Spatial Propagation Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[28] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature Pyramid Networks for Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[29] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[30] X. Guo, K. Yang, W. Yang, X. Wang, and H. Li, "Group-wise Correlation Stereo Network," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.