IEEE Open Journal of
Intelligent Transportation
Systems

Received 15 June 2020; revised 6 October 2020 and 8 December 2020; accepted 30 December 2020. Date of publication 21 January 2021;
date of current version 12 February 2021.

Digital Object Identifier 10.1109/OJITS.2021.3053337

Semi-Automatic Framework for Traffic Landmark Annotation

WON HEE LEE, KYUNGBOO JUNG, CHULWOO KANG, AND HYUN SUNG CHANG (Member, IEEE)

Multimedia Processing Laboratory, Samsung Advanced Institute of Technology, Suwon 16678, South Korea

CORRESPONDING AUTHOR: H. S. CHANG (e-mail: hyun.s.chang@samsung.com)

Data is available on-line at https://ieee-dataport.org/documents/sait-traffic-landmark-dataset.

ABSTRACT We present a semi-automatic annotation method to build a large dataset for traffic landmark detection, where traffic landmarks include traffic signs, traffic lights as well as road markings. Labor-intensive bounding box tagging is a huge challenge to generate a large dataset for detection algorithms. To mitigate the labor, we adopt a high-definition (HD) map and a positioning system. We propose a process to align the HD map and images semi-automatically. Through the registration, the annotations of the HD map can be directly tagged onto traffic landmarks in the images. To make full use of the HD map for the dataset generation, we annotate the traffic landmarks with reference points, following the way that they are represented in the HD map, instead of the bounding boxes. The proposed semi-automatic method speeds up the annotation by a factor of 3.19, as compared to the manual annotation. Our dataset consists of about 150,000 images and includes about 470,000 annotated traffic landmarks. We train a deep neural network on our dataset to detect the traffic landmarks, and its performance is evaluated using a novel evaluation metric. Moreover, we show that the pretrained traffic landmark detection network is effective in detecting traffic landmarks in other countries using the bounding box by fine-tuning.

INDEX TERMS Detection algorithms, image databases, image registration, intelligent vehicles, sensor systems.

I. INTRODUCTION

TRAFFIC signs, traffic lights, and road markings, which we collectively call *traffic landmarks*, have been widely used in various traffic-related applications. For using such traffic landmarks in vision-based applications, it is necessary to know their locations and types in input images. Thus, various algorithms have been studied to detect traffic signs [1], [2], [3], traffic lights [4], [5], [6], and road markings [7]. To train and test detection algorithms, image datasets annotated with bounding boxes and class labels of objects are required.

Several traffic-related datasets for detection algorithms are publicly available. The German traffic-sign detection benchmark (GTSDB) dataset [2] has been widely used in traffic sign detection studies. The GTSDB dataset consists of 600 training images and 300 evaluation images annotated with bounding boxes and class labels of 43 types of German traffic signs. The LISA traffic sign dataset [8] is another widely used dataset. It is designed for U.S. traffic sign detection and consists of more images than GTSDB. The LISA dataset contains 6,610 images annotated with class labels of 49 types of traffic signs and their bounding boxes. The Bosch small traffic lights dataset [6] is a dataset widely used in traffic light detection research. This dataset consists of 5,093 training images and 8,334 test images. These numbers are similar to those of the LISA dataset. The four categories of the traffic light signal and the bounding box indicating the location of the traffic light are annotated. Recently, Lee *et al.* [9] have built a dataset for lane and road marking benchmark which consists of 20,000 images. The lanes and road markings are annotated with pixel masks because it is difficult to indicate their locations using the bounding boxes.

With recent developments in deep learning, vision-based object detection algorithms have been greatly improved by adopting neural networks with a large number of parameters [10], [11]. To prevent overfitting and improve the generalization of the neural network to new data, training data should be collected such that their distribution is close enough to the real data distribution. Unfortunately,

The review of this article was arranged by Associate Editor Jiwon Kim.

the datasets mentioned above contain an insufficient number of images to train a deep neural network, because time-consuming and labor-intensive bounding box tagging task makes it difficult to generate a large dataset for detection algorithms. With the need for a large traffic dataset for detection, Zhu *et al.* [12] created the Tsinghua-Tencent 100K dataset, which is a Chinese traffic sign dataset of 100,000 images annotated with class labels, bounding boxes, and pixel masks. Although they created a large dataset, the inevitable manual bounding box tagging remains a huge challenge for generating larger datasets.

To solve this problem, we propose a semi-automatic annotation framework to build traffic datasets. In this framework, we first acquire and store image sequences and real-time location data of the vehicle through real driving. The location data are obtained from a positioning sensor that integrates the Global Navigation Satellite System (GNSS) and the Inertial Navigation System (INS). Next, we manually specify guide points in the acquired image which are used to define a score function assessing the degree of alignment between the image and the HD map. Then, the vehicle position is corrected by optimizing the score function. Lastly, the annotations of traffic landmarks are transferred from the HD map to the aligned images. Our contributions are as follows:

- We design a semi-automatic annotation framework to facilitate the construction of a large traffic dataset for detection. The dataset generated with our framework includes about 150,000 images and annotations of about 470,000 traffic landmarks such as traffic signs, traffic lights, and road markings. Our dataset is primarily targeted for localization based on traffic landmark detection.
- We migrate a generic object detection network for detecting reference points of traffic landmarks because reference points are commonly used in HD maps as the traffic landmark representation. We train the detection network with our dataset and evaluate the results on the datasets.
- We show that the traffic landmark detection network trained on our dataset can be applied for detection of traffic landmarks of various countries with bounding boxes by finetuning on public traffic datasets.

This article is organized as follows. Section II gives a brief overview of HD maps and studies on traffic landmark detection. Our framework and procedure of dataset generation are described in Section III. Experimental results are presented and discussed in Section IV. Finally, Section V concludes our work.

II. RELATED WORK

In this section, we briefly introduce HD maps and their reference points. Subsequently, deep learning-based traffic landmark detection algorithms are reviewed.

A. HIGH-DEFINITION MAP AND REFERENCE POINTS

With the growing interest in autonomous driving, the need of high precision localization is increasing. HD maps are essential for high precision localization using vision sensors. Commercial map providers, such as HERE [13] and TomTom [14], produce HD maps annotated with 3D geographic and semantic information on traffic landmarks within an accuracy of 10 cm. As these map providers expand the range of the maps in various countries, HD maps are expected to be used in various studies on intelligent vehicles. Ride-sharing platform companies such as Lyft Level 5 [15] sometimes create their own HD maps for localization of autonomous vehicles. Crowd-souring is also widespread, utilizing in-vehicle diagnostic data or smartphone sensor data from many sources and generating HD maps at low cost [16], [17], [18], [19]. Along with the above individual efforts to create HD maps, there are also activities that emphasize the standardization of HD maps [20]. In South Korea, the National Geographic Information Institute (NGII) [21] builds HD maps with accuracy comparable to that of the HD maps provided by HERE and TomTom.

An HD map is created by processing data from a mobile mapping system equipped with LiDAR, GNSS, INS, and camera sensors [22], [23]. The mobile mapping system generates a geographic 3D point cloud by integrating local 3D point cloud data acquired from the LiDAR and geographic position and attitude of the system from the GNSS and INS. Referring to the camera images, the geographic 3D positions of traffic landmarks are stored in the HD map along with the semantic information. As it is difficult to store all 3D points of the HD map components because of the capacity problems, only a small number of representative points, referred to as *reference points* throughout this article, are stored. For polygonal components such as a crosswalk, the reference points are the points that constitute the polygon. Linear components such as lane markings and curbs use the center points of the line segments as reference points. Pointwise components, such as traffic lights and traffic signs, usually have their center points as reference points. Note that in the NGII HD map used in our experiments, a road marking is stored in the form of a point rather than a polygon, and its reference point is located at the lower right corner (for example, see Fig. 5(b)).

The proposed semi-automatic annotation framework inevitably relies on the HD maps containing the 3D location and semantic information of traffic landmarks as well as of lane lines. Accordingly, the applicability of this method is limited, but it is expected to grow rapidly as HD maps are widely built and deployed around the world.

B. TRAFFIC LANDMARK DETECTION

Detection algorithms of traffic signs, traffic lights, road markings, or lanes, which are installed along the road, are essential elements of advanced driver assistance systems and autonomous driving systems. Recently, various algorithms relying on deep learning have been developed. Zhu *et al.* [12] applied a convolutional neural network (CNN) to traffic sign detection. For a robust deep learning-based algorithm, they created a Chinese traffic sign dataset (Tsinghua-Tencent 100K) containing 100,000 images and confirmed its effectiveness. Another CNN-based method [24] was proposed to identify traffic sign locations using a fully convolutional network and to classify traffic sign types using a CNN. Their experiments were performed on the Swedish Traffic Signs dataset. Arcos-García *et al.* [25] applied various object detection algorithms to traffic sign detection with various deep networks for feature extraction, and they compared the results using the GTSDB. Qian *et al.* [26] proposed a fast-RCNN-based road marking detection system, which was trained and tested on their own road marking dataset of the U.K. Kim *et al.* [27] proposed a two-step method to find traffic light candidates based on a deep neural network and to detect false traffic lights using a point-based reward system. The U.S. traffic light dataset was collected for training and evaluation of the proposed method.

Because traffic environments are different in different countries, the datasets are also different. Thus, the learning-based methods introduced above are bound to depend on the datasets used for their training. Direct application of a method trained on one country's dataset to another is difficult. To alleviate this difficulty, we investigate whether the knowledge of large traffic datasets collected in one country is also effective in detecting traffic landmarks in other countries.

III. DATASET GENERATION

To generate a dataset, data acquisition and annotation should be performed. In the data acquisition process, data from several sensors are acquired simultaneously. In the annotation process, the HD map and images are semi-automatically aligned. Our dataset consists of 155,197 images and 468,651 instances. The dataset is published and publicly available at IEEE Dataport [28].

A. SENSOR DESCRIPTION AND DATA COLLECTION

A dataset for traffic landmark detection contains localization data and image data. For localization data acquisition, a high-accuracy GNSS/INS integrated system is utilized. For image data acquisition, commercial-level image sensors are used. Detailed specifications of the sensors are provided below.

- *Novatel SPAN-CPT RTK GNSS/INS Positioning System:* 6 axis, sampling rate: 100 Hz, location accuracy: 0.01 m, velocity accuracy: 0.02 m/s, attitude accuracy: 0.05° (pitch/roll), 0.1° (azimuth angle).
- *OnSemi AR0134 CMOS Sensor:* 1/3″, 1.2 Megapixels (1280 × 960), global shutter, 54 fps, focal length: 6 mm. *OnSemi MT9M024 CMOS sensor.* 1/3″, 0.9 Megapixels (1280 × 720), electronoic rolling shutter, 60 fps, focal length: 6mm.

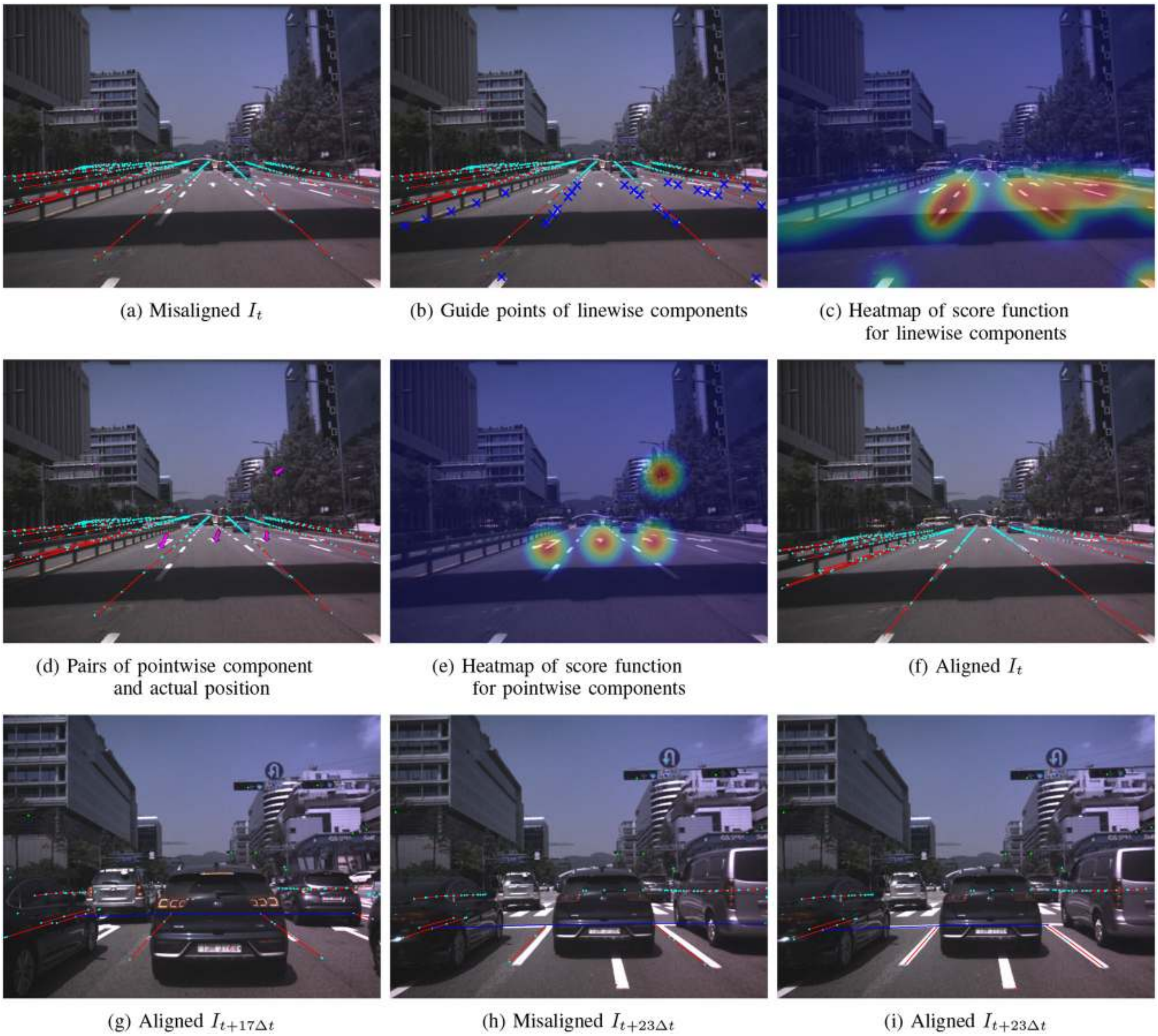
The camera is mounted on the windshield in the front direction, and the localization sensor, GNSS/INS integrated system, is located on the floor under the driver's seat. Camera calibration for intrinsic parameters and calibration for extrinsic parameters between the camera and the positioning sensor are performed prior to the data acquisition. We assume that the camera and the positioning sensor are fixed on the rigid frame.

Our dataset was collected in an urban area of Seoul and suburban areas of Suwon, Hwaseong, Yongin, and Seongnam in South Korea at different times of the day. Images taken in the morning or evening included a large number of saturated areas due to exposure to direct sunlight. Most images taken under the low light condition of the late evening were low-contrast. The images taken at noon included the reflection of the windshield due to strong sunlight. These various elements contribute to the development of a robust detection algorithm because they can be encountered in a real test environment.

B. SEMI-AUTOMATIC ANNOTATION

Annotation is the main difficulty in constructing a traffic dataset for detection because annotating images with bounding boxes and class types is labor-intensive. To reduce the manual labeling effort, we suggest using an HD map for annotation, as HD maps contain geo-locations and class types of traffic landmarks that can be used for localization. Specifically, once an image has been obtained from the region where the HD map exists, class types and their image coordinate position can be tagged by projecting the 3D geolocation of traffic landmarks in the HD map onto the images using the precise location. As precise localization is difficult, we obtain the location and attitude of the vehicle from the positioning system synchronized with the camera. Although the accuracy of the positioning system is centimeter-level, the projected traffic landmarks and lanes are misaligned due to the dynamic positioning error and the calibration error of the coordinate systems of the positioning system and camera. For example, when the HD map is projected onto the image using the position and attitude directly obtained by the positioning system, the projected positions of the traffic landmarks and lanes are misaligned with those in the image, as shown in Fig. 1(a). Therefore, registration between the HD map and the image is important when creating the dataset.

Since it is difficult to automatically align HD maps and images with high accuracy, human intervention is involved in the registration process illustrated in Fig. 2. There are two manual interventions involved in the registration process. One intervention is to specify where the linewise components of the map are to be located in the image by selecting multiple guide points. The other intervention is to pair the misaligned pointwise components with their actual positions. Manual interventions need not be thorough, so it is allowed to skip some landmarks especially if they are occluded and thus are not visible in the image. Examples of selected points and pairs are illustrated by blue crosses and magenta arrows in Figs. 1(b) and 1(d). Note that the different process for two types of components is due to the ambiguity in specifying



(a) Misaligned I_t

(b) Guide points of linewise components

(c) Heatmap of score function for linewise components

(d) Pairs of pointwise component and actual position

(e) Heatmap of score function for pointwise components

(f) Aligned I_t

(g) Aligned $I_{t+17\Delta t}$

(h) Misaligned $I_{t+23\Delta t}$

(i) Aligned $I_{t+23\Delta t}$

FIGURE 1. Visualization of the proposed annotation procedure. The HD map is initially misaligned with the image, I_t, in (a). For the semi-automatic registration, guide points for linewise components are designated as (b) and misaligned pointwise components are paired with their actual positions as (d). The score functions of the linewise and the pointwise components are visualized using heatmaps in (c) and (e), respectively. By applying the optimized position correction parameter, the HD map is transformed to be aligned with the image as shown in (f). The same parameter can be applied to the following frames, e.g., $I_{t+17\Delta t}$ in (g). When the frame is out of alignment, e.g., $I_{t+23\Delta t}$, a new registration procedure is performed, as shown in (h) and (i).

the correspondence points of the linewise components. In order to align the HD map with the image using the manual selections, we define a score function to maximize in terms of 6-DOF pose correction parameters.

To formulate the score function, we start by defining mathematical notations. Let $\mathbf{w}$ be a four-dimensional vector comprising the 3D coordinate (x, y, z) of an HD map component, represented in the camera coordinate system, and 1, i.e., $\mathbf{w} = \left[x; y; z; 1\right]$. We use superscript L for linewise component ($\mathbf{w}^{\mathrm{L}}$) and superscript P for pointwise component ($\mathbf{w}^{\mathrm{P}}$) to distinguish between the two types, and use subscripts whenever we specifically denote individual samples ($\mathbf{w}_i^{\mathrm{L}}$ or $\mathbf{w}_i^{\mathrm{P}}$). Note that the camera geolocation is required to obtain $\mathbf{w}^{\mathrm{L}}$ and $\mathbf{w}^{\mathrm{P}}$ in camera coordinates.

A pose correcting transformation is typically represented by $\mathbf{T} = [\mathbf{R} \mid \mathbf{t}]$, where $\mathbf{R}$ and $\mathbf{t}$ denote a 3D rotation matrix and a translation vector. Although $\mathbf{R}$ is a 3×3 matrix, it must belong to the special orthogonal group, only having three degrees of freedom. Generally, it should be decomposable into

$$\mathbf{R} = \underbrace{\begin{bmatrix} c_z & -s_z & 0 \\ s_z & c_z & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{=\mathbf{R}_z} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_x & -s_x \\ 0 & s_x & c_x \end{bmatrix}}_{=\mathbf{R}_x} \underbrace{\begin{bmatrix} c_y & 0 & s_y \\ 0 & 1 & 0 \\ -s_y & 0 & c_y \end{bmatrix}}_{=\mathbf{R}_y}, \tag{1}$$

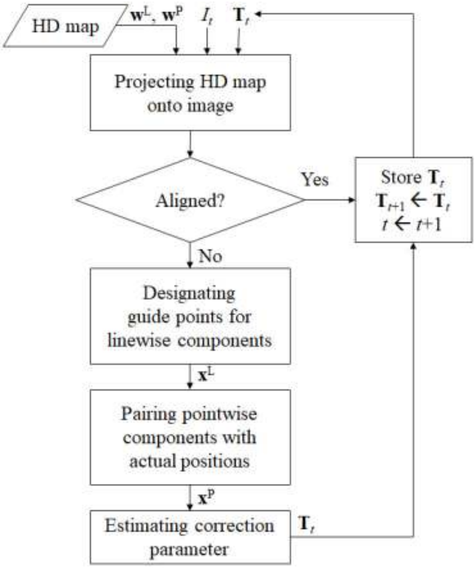

HD map
$\mathbf{w}^{\mathrm{L}}, \mathbf{w}^{\mathrm{P}}$
I_t
$\mathbf{T}_t$
Projecting HD map onto image
Aligned?
Yes
Store $\mathbf{T}_t$
$\mathbf{T}_{t+1} \leftarrow \mathbf{T}_t$
$t \leftarrow t+1$
No
Designating guide points for linewise components
$\mathbf{x}^{\mathrm{L}}$
Pairing pointwise components with actual positions
$\mathbf{x}^{\mathrm{P}}$
Estimating correction parameter
$\mathbf{T}_t$

FIGURE 2. Procedure of the proposed semi-automatic registration. The HD map components, w^{L} and w^{P}, are projected onto the image, I_t, using the current pose correcting transformation, T_t. If aligned, the parameter is stored for the next frame. Otherwise, manual intervention to specify the actual positions of the HD map components is conducted. Through the optimization, a new correcting transformation is estimated.

where $c_v = \cos(\theta_v)$, $s_v = \sin(\theta_v)$, for $v = x, y, z$. We call θ_z, θ_x, θ_y Tait-Bryan angles [29] or *roll*, *pitch*, *yaw* in the respective order.

Then, given the camera intrinsic matrix $\mathbf{K}$, the HD map component $\mathbf{w}$ is projected onto the pixel $\mathbf{u} = \Pi(\mathbf{KTw})$ in the image, where Π is the perspective projection function, i.e., $\Pi : (x, y, z) \mapsto (x/z, y/z)$. The intrinsic matrix $\mathbf{K}$ is usually represented by

$$\mathbf{K} = \begin{bmatrix} f_x & k & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

Here, f_x and f_y are focal lengths, k is the skew parameter, and p_x and p_y represent the principal point. These parameters are independent of $\mathbf{T}$ and can be calibrated in advance.

We are now ready to formulate the score function. Let us denote the manually designated guide points by $\mathbf{x}_i^{\mathrm{L}}$ and $\mathbf{x}_i^{\mathrm{P}}$. For pointwise components, $\mathbf{u}_i^{\mathrm{P}}$ and $\mathbf{x}_i^{\mathrm{P}}$ have an explicit one-to-one correspondence between the two, and the score function should increase as $\mathbf{u}_i^{\mathrm{P}}$ becomes close to $\mathbf{x}_i^{\mathrm{P}}$. This is basically the same as what the conventional 3D-2D point registration does. However, this strategy does not hold for linewise components with which such correspondence is unavailable. The score function should be designed to increase if $\mathbf{u}_i^{\mathrm{L}}$ matches any of $\mathbf{x}_j^{\mathrm{L}}$ and not to be penalized excessively even if it founds no match. Keeping this in mind, we write the score function as

$$s(\mathbf{T}) = \sum_i \sum_j \frac{e^{-\left\|\mathbf{u}_i^{\mathrm{L}} - \mathbf{x}_j^{\mathrm{L}}\right\|^2 / 2\sigma^2}}{2\pi\sigma^2 n} - \alpha \sum_i \left\|\mathbf{u}_i^{\mathrm{P}} - \mathbf{x}_i^{\mathrm{P}}\right\|,$$
$$\text{subject to } \mathbf{u}_i^{\mathrm{L}} = \Pi\big(\mathbf{KTw}_i^{\mathrm{L}}\big),\ \mathbf{u}_i^{\mathrm{P}} = \Pi\big(\mathbf{KTw}_i^{\mathrm{P}}\big). \tag{3}$$

Here, n is the number of the user-designated guide points for the linewise components ($\mathbf{x}_j^{\mathrm{L}}$). The values of σ and α are set to 51 and 1, respectively.

The score function in (3) has the form of Gaussian mixture with respect to $\mathbf{u}_i^{\mathrm{L}}$ and the negative Euclidean distance with respect to $\mathbf{u}_i^{\mathrm{P}}$, as illustrated by heatmaps in Figs. 1(c) and 1(e). Unlike the conventional 3D-2D point registration, the reason for including the linewise components in registration is to maintain the correct pose for every frame and to ultimately reduce our manual intervention. Driving scenes often have fewer pointwise components than required by the conventional 3D-2D point registration algorithm (e.g., perspective-n-point solution [30]). In contrast, linewise components are almost everywhere and help to successfully perform the registration even in situations where the number of pointwise components is scarce.

We convert (3) into an unconstrained problem by replacing $\mathbf{u}_i^{\mathrm{L}}$, $\mathbf{u}_i^{\mathrm{P}}$ in $s(\mathbf{T})$ with $\Pi(\mathbf{KTw}_i^{\mathrm{L}})$ and $\Pi(\mathbf{KTw}_i^{\mathrm{P}})$, respectively, i.e.,

$$\begin{aligned} s(\mathbf{T}) = \sum_i \sum_j & \frac{e^{-\left\|\Pi\left(\mathbf{KTw}_i^{\mathrm{L}} - \mathbf{x}_j^{\mathrm{L}}\right\|^2 / 2\sigma^2}}{2\pi\sigma^2 n} \\ & - \alpha \sum_i \left\|\Pi\big(\mathbf{KTw}_i^{\mathrm{P}}\big) - \mathbf{x}_i^{\mathrm{P}}\right\|. \end{aligned} \tag{4}$$

Recall that matrix $\mathbf{T}$ is parameterized by θ_x, θ_y, θ_z and $\mathbf{t} = (t_x, t_y, t_z)$. To maximize (4) with respect to these six scalar variables, we use `fminsearch`, a built-in MATLAB$^{\mathrm{TM}}$ function which is based on the Nelder-Mead search algorithm [31]. Also known as downhill simplex method, the Nelder-Mead method is a heuristic but very popular method for multi-dimensional unconstrained optimization. Because it is based on direct comparison between function values, it is gradient-free and often applied to nonlinear optimization problems for which derivatives may not be known. In our case, the Nelder-Mead algorithm maintains a 6D polytope with seven vertices, with each vertex corresponding to a candidate solution, and iteratively updates the polytope by replacing the worst vertex with a new one. The new candidate solution is usually obtained by reflecting the worst vertex through the centroid of the others and occasionally by contracting the worst vertex toward the centroid (in case that reflection turns out to make no improvement). If neither of these replacement schemes works well, the algorithm shrinks the polytope in all directions, pulling it around the best point. The Nelder-Mead method is very efficient, while it possibly converges to a non-stationary point. Interested readers can refer to [32] for more details and analysis on the algorithm.

As a result of the optimization, we obtain $\mathbf{T}$ with which the HD map becomes aligned with the entire image as shown in Fig. 1(f). The alignment is maintained even when the correction parameter is applied to the following frames. Therefore, we apply the same correction parameters to the subsequent frames until the alignment is maintained, as shown in Fig. 1(g), and then a new human intervention and optimization process are performed at the frame that is out of alignment like in Fig. 1(h).

Although the HD map and the images are exactly aligned, the traffic landmark dataset contains outliers because a discrepancy may occur between the map information and the images. For example, the image and the map information may be different owing to the occlusion caused by the front vehicle or the roadside trees when acquiring the image. Back-facing traffic signs are also outliers because their categories cannot be predicted by the image. In addition, the traffic structure at the time of the map creation can differ from that during the image acquisition. To deal with these outliers, we simply select and tag them as such. A similar procedure used for aligning the map and the image is also utilized here. Outliers are tagged only in one frame with human intervention, and then the tagged outliers are propagated to subsequent frames until there is a change.

Note that using the optimized position correction parameters, all information contained in the HD maps can be converted into datasets. For example, not only lanes, curbs, stop lines, etc., but also pedestrian crossings and median strips can be converted into datasets. In addition to the information contained in the HD map, data generated by processing the HD map information, such as vanishing points, can also be created in the dataset. Furthermore, by aligning the overall instances in the image with the HD map, the variance of annotation position errors can be reduced, as compared to the manual annotation, and reliable annotation results can be consistently obtained.

C. DATASET STATISTICS

Our traffic landmark dataset consists of 155,197 images taken in various places in South Korea. The video frame rate is about 8 frames per second. As we collected images using two different image sensors, there are two types of image sizes: 1280 × 720 and 1280 × 960. Only the traffic landmarks located within 80 m from where the image is captured are used for annotation. Annotated classes are six traffic landmark types: warning sign, prohibition sign, mandatory sign, supplementary sign, traffic light, and road marking. Although the classes in our dataset are divided into major categories according to the shape of the traffic landmark, a class can be further subdivided using the subclasses stored in the HD map. For example, traffic light subclass can be horizontal light, vertical light, pedestrian light, etc. The subcategorization costs nothing if and only if it is confined to the subclasses defined in the HD map. Any other subclasses, e.g., dynamic status of traffic light such as "red-on traffic light," would require additional annotation. The number of instances for each class is listed in Table 1. The images are taken in various regions, but a data imbalance between the traffic light class and the other classes is observed. Although there is the data imbalance, the number of instances of the minority class is more than 10,000. Therefore, it seems to be sufficient for the application to machine learning-based algorithms.

TABLE 1. Number of instances for each class in the traffic landmark dataset.

Number of images	Train/Val	129,201	(83.2%)
	Test	25,996	(16.8%)
	Total	155,197	
Number of instances	Warning sign	11,259	(2.4%)
	Prohibition sign	47,872	(10.2%)
	Mandatory sign	45,128	(9.6%)
	Supplementary sign	27,217	(5.8%)
	Traffic light	258,125	(55.1%)
	Road marking	79,050	(16.9%)
	Total	468,651	

IV. EXPERIMENTAL RESULTS

In this section, we compare the effectiveness of our semi-automatic annotation method and the manual annotation. We also attempt to learn a traffic landmark detector with our dataset. Because our dataset uses reference points as the landmark representation, we modify the single shot multibox detector (SSD) [10] to regress the location of the reference point as well as the class of each landmark. Subsequently, we perform various experiments to evaluate the detector trained on our dataset. Details for training the detection network are provided, and the metric for evaluating the performance of the reference point detection is discussed. Then, the detection results are analyzed on our dataset. In addition, we finetune the network pretrained on our dataset to various public traffic datasets and compare the results with those obtained using the network pretrained on common object detection datasets.

A. EFFECTIVENESS OF SEMI-AUTOMATIC ANNOTATION

To examine the effectiveness of the proposed semi-automatic annotation, over the manual annotation of the reference points, the number of clicks per frame and the cumulative time required to annotate 1,000 frames are compared. The manual annotation requires clicks to locate reference points and assign their class labels. Meanwhile, the semi-automatic annotation requires clicks for the registration. The class labels are directly assigned from the HD map annotations. In Fig. 3, in the case of the manual annotation, multiple clicks are required in a majority of frames. The numbers of clicks are proportional to the numbers of traffic landmarks in the frames. As can be seen in the cumulative annotation time graph, the annotation time increases rapidly in frames with a large number of traffic landmarks. In contrast, the semi-automatic annotation does not require clicks in a majority of frames, and requires a relatively large number of clicks only in misaligned frames. In the semi-automatic annotation, the average number of clicks per frame is 1.83, which is significantly lower than 10.68 required for the manual annotation. The speed-up factor in the annotation time is 3.19, as compared to the manual annotation of the reference points. In

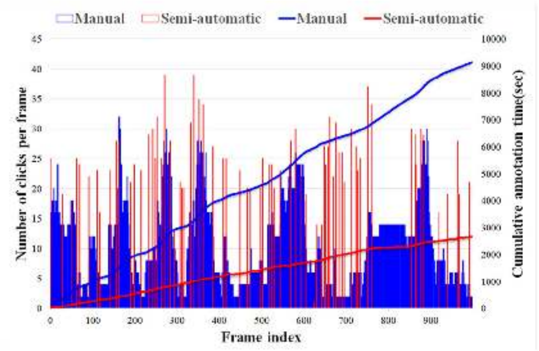


FIGURE 3. Our semi-automatic annotation versus manual annotation. The number of clicks and the cumulative time required to annotate 1,000 frames are compared. Our semi-automatic annotation requires only 1.83 clicks on average, which is considerably lower than 10.68 clicks in the manual annotation. The annotation speed-up factor is 3.19.

TABLE 2. Average distance and mean average precision results evaluated on our datasets.

Class	AD	$\mathrm{AP}_{\mathrm{dist}<32}$	$\mathrm{mAP}_{\mathrm{dist}<32}$
Warning sign	3.88	76.80	75.26
Prohibition sign	4.59	81.09	
Mandatory sign	4.68	76.18	
Supplementary sign	4.86	59.24	
Traffic light	4.59	79.34	
Road marking	4.05	78.93	

the case of bounding box annotation, a significantly higher speed-up factor is expected.

B. TRAINING TRAFFIC LANDMARK REFERENCE POINT DETECTION

Traffic landmarks are often detected in the form of a bounding box [5], [6], [7], [33], or they sometimes can be detected in the form of a grid-level mask [9]. In contrast, the traffic landmarks in the HD map are represented in the form of a reference point. Since the reference points are mainly used for HD map-based localization [34], it is beneficial to directly detect traffic landmarks in the form of reference point. Otherwise, additional post-processing is required to find the image coordinate corresponding to the reference point of the detected landmark [35].

To detect the reference points, we modify the single shot multibox detector (SSD) [10], which is often used for object detection. Unlike the SSD which regresses the center position and size of bounding boxes, our detector localizes traffic landmarks only with the coordinate of the reference points. Moreover, we do not apply the multibox of the original method because various sizes and shapes of the bounding box are not required for detection of the reference points. In addition, we change the base network to MobileNet [36] for efficiency.

In the training, we use an RMSProp [37] optimizer with an initial learning rate of 0.0005 and weight decay of 0.0005. We train our network over 800k iterations where the learning rate decreases every 200k iterations by a factor of 2. We used a batch size of 64. To accommodate a large batch, we reduce the size of the input images. Rescaling is usually performed in training general object detectors. However, as traffic landmarks, such as traffic lights and traffic signs, occupy small areas in the images, the image information of a majority of traffic landmarks will be lost after rescaling. For this reason, cropped images centered on traffic landmarks with random displacement were used for the training.

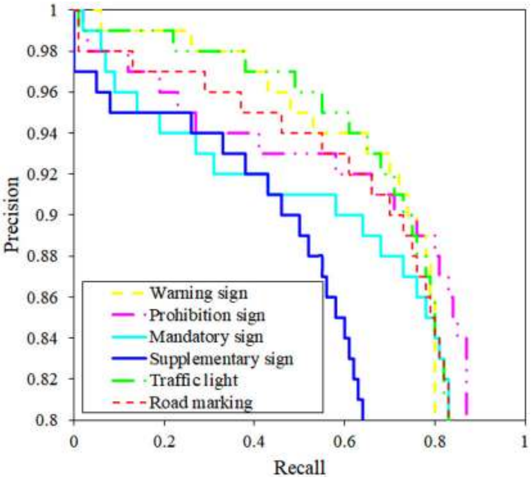


FIGURE 4. Precision-recall curves of each class. The overall performance of detecting classes is similar, except for the supplementary sign class. The detector performance tends to degrade because the supplementary signs do not have a common pattern unlike other signs.

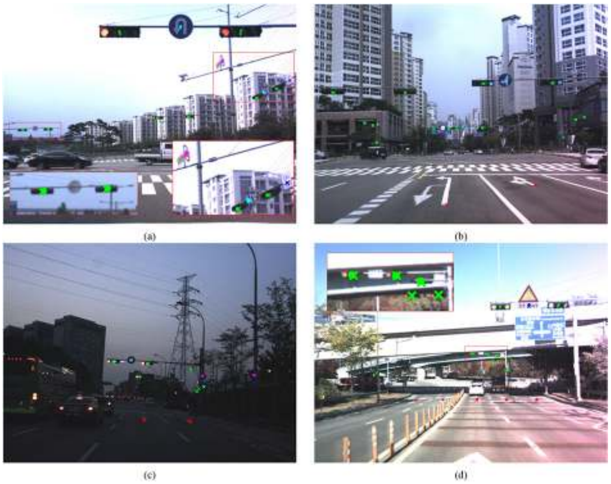
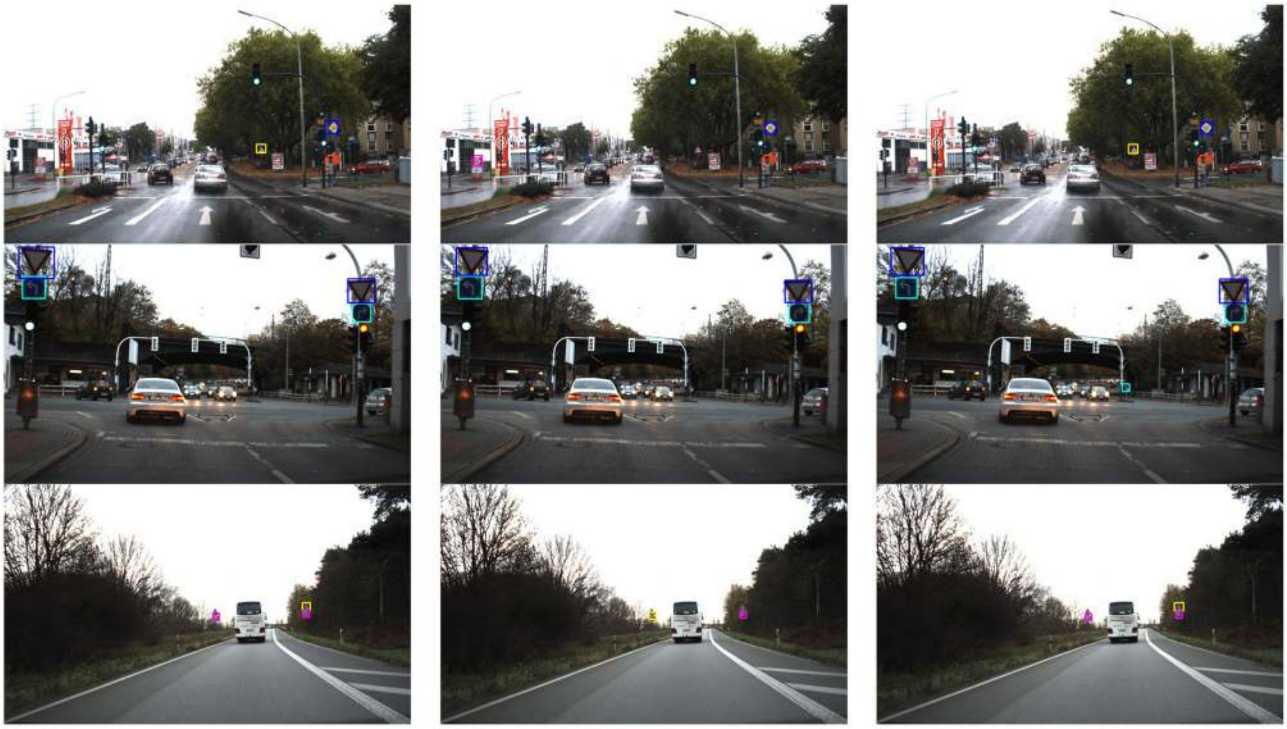


FIGURE 5. Qualitative results of the traffic landmark reference point detection. Warning, prohibition, mandatory, and supplementary signs, traffic light, and road marking are marked with yellow, magenta, cyan, blue, green, and red circles, respectively. Ground truth landmarks are marked with crosses. Note that the reference point of road marking is the bottom right corner of the marking. Our detector successfully localizes the reference points of the traffic landmarks of various scales, in different directions, and under different lighting conditions.

C. EVALUATION METRICS FOR REFERENCE POINT DETECTION

The most commonly used metrics for measuring the performance of an object detector are the mean intersection over union (mIoU) and the mean average precision (mAP).



(a) Random initial

(b) Pretrained on VOC0712

(c) Pretrained on our dataset

FIGURE 6. **Qualitative results on GTSDB varying pretrained dataset. Mandatory, prohibition, danger, and other signs are marked with cyan, magenta, yellow, and blue bounding boxes, respectively. Although the network pretrained on our dataset detects traffic landmarks with reference points, it is more effective than the network pretrained on VOC0712 finetuned to detect German traffic signs using bounding boxes.**

The IoU between the ground truth bounding box and the detected bounding box measures the location accuracy of the detected box. As no bounding box is estimated in reference point detection, its performance cannot be measured using the IoU metric. Instead, it is appropriate to use the average distance (AD) between the ground truth reference point and the predicted reference point as a new evaluation metric.

$$\mathrm{AD} = \frac{1}{N}\sum_{i}\left\|\mathbf{x}_{\mathrm{D}}^{i} - \mathbf{x}_{\mathrm{GT}}^{\hat{j}}\right\|_{2} \tag{5}$$

The average distance is applied to the pair of a detected landmark and the nearest ground truth landmark whose class label is the same as the detected one. Following the same analogy with the IoU metric, if the distance is above a predefined threshold, the detected landmark is considered not to correspond to a ground truth landmark. The threshold is set to 32 in the experiments. The advantage of AD is that it is directly related to the localization error that can be expected when using the detected reference points for localization.

The mAP metric [38] evaluates the overall performance of the object detector. To calculate the mAP, the precision at each recall level should be obtained and then averaged for each class. The precision is the ratio of the total number of true predictions to the total number of predictions. Determining whether the prediction is true or false depends on the IoU threshold. As the IoU cannot be defined in the reference point detection, the IoU threshold is replaced with the predefined threshold of distance. The mAP determined by the distance threshold, *th*, is denoted by $\mathrm{mAP}_{\mathrm{dist}<th}$.

Although many traffic landmark detectors exist, it is difficult to perform a comparison with them, because they mostly detect landmarks with bounding boxes rather than reference points, and are optimized for the traffic data of different countries. Moreover, an integrated Korean dataset of traffic signs, traffic lights, and road markings is not available. For this reason, we evaluate our reference point detector on our dataset.

D. EVALUATION OF REFERENCE POINT DETECTION

The results of AD and $\mathrm{mAP}_{\mathrm{dist}<32}$ are shown in Table 2, and the precision-recall curve for each class is shown in Fig. 4. Compared to other classes, the supplementary sign class has noticeably lower AP. This seems to be because supplementary traffic signs do not have a common pattern, unlike other classes. Fig. 5 shows the qualitative results of our reference point detector. Fig. 5(a) shows that the center points of various traffic signs and traffic lights are well detected. In particular, in the right part of the image, traffic signs and traffic lights that do not face forward are detected, and in the left part, traffic lights in the opposite lane are also detected. Interestingly, the back of the traffic sign in the opposite lane is not detected because the class cannot be determined. In Fig. 5(b), traffic signs, traffic lights, and

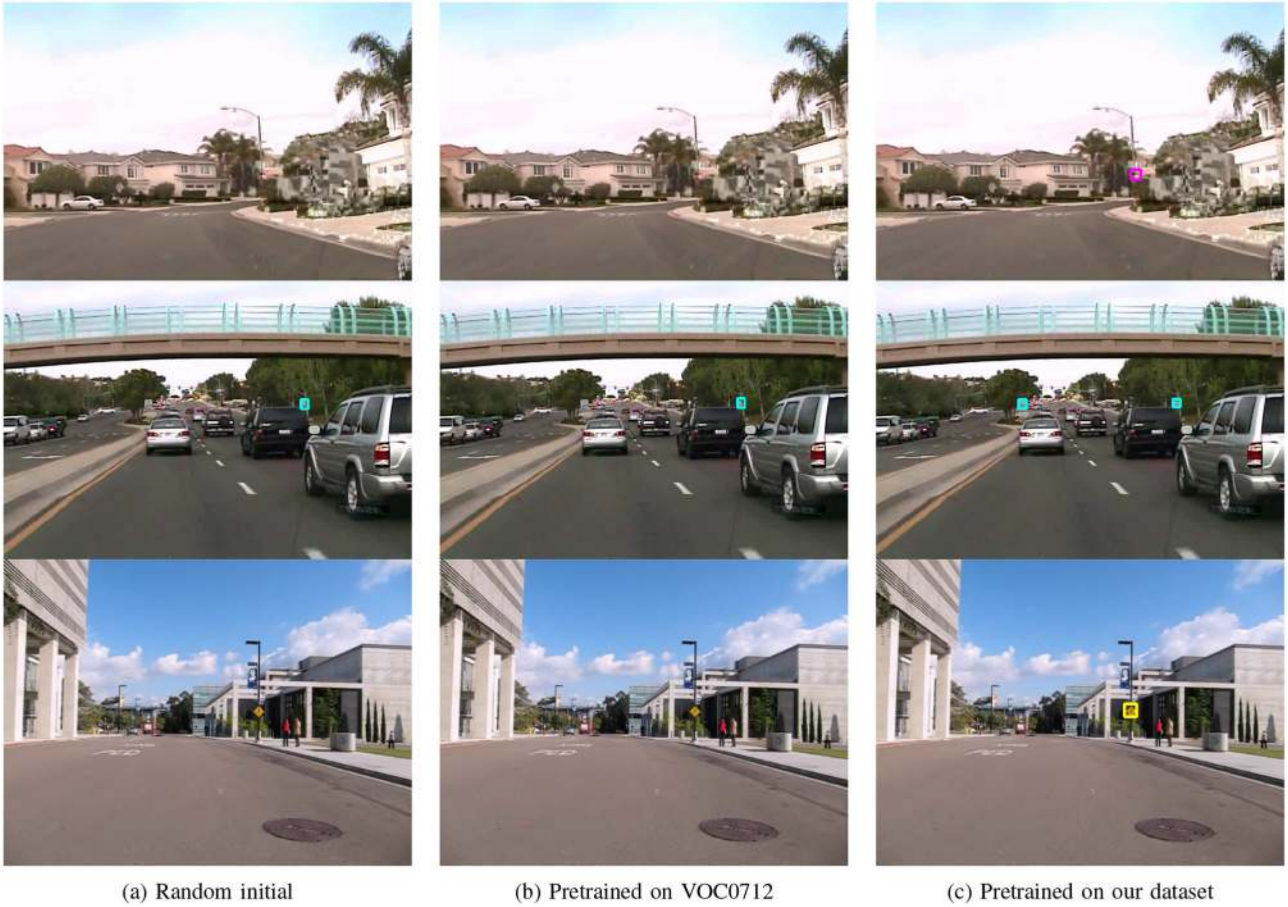

(a) Random initial
(b) Pretrained on VOC0712
(c) Pretrained on our dataset

FIGURE 7. Qualitative results on LISA varying pretrained dataset. Mandatory, prohibitory, and warning signs are marked with cyan, magenta, and yellow bounding boxes, respectively. The network pretrained on our dataset can be finetuned to detect U.S. traffic signs. It is more effective in the detection of small traffic signs, as compared to other detectors.

road markings are detected with similar performance. In particular, the lower right corners of the road markings, which are their reference points, are accurately detected. The reference points of these road markings are difficult to detect using bounding boxes. In the case of road marking on the far right, the reference point is located outside the boundary of the image and is not detected. Our dataset includes images acquired in the evening when the illumination is low, so traffic landmarks in low-illumination images can be also detected, as shown in Fig. 5(c). Our traffic landmark detector sometimes fails to detect distant traffic landmarks, as shown in Fig. 5(d). This is not because the size of the landmarks is small, but because some landmarks are missing in the non-maximum suppression process when several landmarks are adjacent. The traffic lights located in the distance in Fig. 5(d) have similar sizes, but only some of them are detected.

TABLE 3. Evaluation on GTSDB varying pretrained datasets.

Pretrained dataset	Class	Avg. IoU	AP	mAP
None (random initial)	Mandatory	75.96	44.27	57.76
	Prohibitory	84.48	78.78	
	Danger	82.05	71.08	
	Other	78.13	36.93	
VOC0712	Mandatory	78.41	51.54	63.82
	Prohibitory	80.22	78.73	
	Danger	82.62	72.27	
	Other	77.78	52.76	
MS COCO[25]	Mandatory	78.51	52.01	61.64
	Prohibitory	80.49	67.03	
	Danger	81.11	65.85	
	Other	N/A	N/A	
Our dataset	Mandatory	75.11	56.36	66.77
	Prohibitory	83.37	84.76	
	Danger	80.88	73.74	
	Other	80.90	52.19	

E. BOUNDING BOX DETECTION

Although our deep neural network was optimized to detect Korean traffic landmarks using reference points, we can finetune our network to detect traffic landmarks of other countries using bounding boxes. To verify the effectiveness of our dataset, we finetune the pretrained networks on the datasets for common object detection, such as VOC0712 [38] and the MS COCO [39], to detect traffic landmarks of public datasets and compare the results with ours. For a fair comparison, the results are compared by differentiating only the pretrained dataset, whereas the network structure and detection algorithm are MobileNet and SSD, respectively.

TABLE 4. Evaluation on LISA varying pretrained datasets.

Pretrained dataset	Class	Avg. IoU	AP	mAP
None (random initial)	Mandatory	84.02	89.68	92.51
	Prohibitory	84.44	93.36	
	Warning	86.77	94.51	
VOC0712	Mandatory	84.06	92.58	94.26
	Prohibitory	85.35	93.64	
	Warning	87.39	96.55	
MS COCO[41]	Mandatory	75.68	72.29	85.56
	Prohibitory	81.47	89.17	
	Warning	79.99	95.23	
Our dataset	Mandatory	84.89	94.78	95.15
	Prohibitory	86.20	94.84	
	Warning	87.97	95.81	

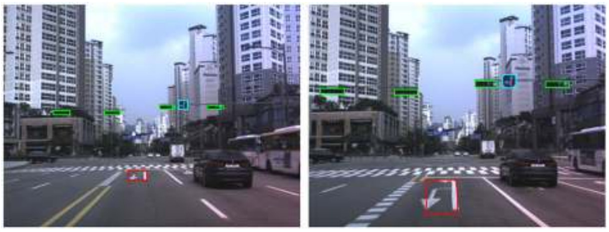


FIGURE 8. Extension to bounding box annotation. In the left figure, bounding boxes of traffic landmarks are manually annotated. The 3D locations of corner points of the bounding boxes can be obtained based on the aligned HD map. The 3D bounding boxes can be propagated to the following frames as an example is shown in the right figure.

The GTSDB [2] and LISA [8] datasets are used for traffic sign detection. Traffic signs in GTSDB are classified into four categories: mandatory, prohibitory, danger, and other. As both the Korean and German traffic signs are based on the Vienna convention traffic rules [40], traffic signs from both countries include common features. Accordingly, the network pretrained on our dataset is expected to be more effective. As shown in Table 3, when using a network pretrained on our dataset, higher mAP can be achieved than using other datasets. This confirms that our dataset is more suitable for extracting common features of traffic landmarks than datasets for common object detection. Qualitative results are compared in Fig. 6. As shown in the top row, when using our dataset, the position accuracy of the bounding box and the overall performance of traffic sign detection are both improved. As the dataset includes images of traffic landmarks of various scales, signs of various sizes in the middle low can be successfully detected. However, some signs are not detected, when several small signs are adjacent, as shown in the bottom row.

Unlike GTSDB dataset, the U.S. traffic signs of the LISA dataset are not standardized by the Vienna convention. Although performance gaps with other datasets were reduced, the highest mAP is obtained when using our datasets in Table 4. As the LISA dataset has more images than the GTSDB dataset, the detection performance of all networks finetuned to the LISA dataset is high, but when using our dataset, small traffic signs are detected with better accuracy, as shown in Fig. 7.

V. CONCLUSION AND FUTURE WORK

In this article, we suggested a novel semi-automatic annotation framework for dataset generation to reduce laborious manual annotation. Using the annotation framework, we were able to speed up the annotation process, and thereby build a large traffic dataset. By training a deep neural network on the generated dataset, we successfully detected reference points of traffic landmarks and evaluated its effectiveness. In addition, we showed that our reference point detection network can be used for detection of traffic landmarks of different countries using bounding boxes.

There is room for extending our framework to direct bounding box tagging. To probe the feasibility, we manually annotate bounding boxes for one frame, compute the 3D locations of the bounding boxes based on the aligned reference points, propagate the 3D locations of the bounding boxes in time, and project them onto the image plane at each frame, finally producing the 2D bounding box annotation. As shown in Fig. 8, this simple procedure works well for traffic lights and traffic signs, but was not quite suitable for road markings which are sensitive to severe perspective transformations. Making the semi-automatic bounding box annotation versatile for serious perspective transformations is not straightforward, and we leave it as future work.

We also plan to develop a good way of including a neural network in the loop of semi-automatic annotation. This may involve reinforcement learning of a neural network (or fine-tuning of a pretrained one) with sequential reward/supervision from the human annotator. When properly trained, the neural network is expected to start providing very accurate results, which further reduces the need for human intervention as time elapses. In that way, a lot of manual labor can be greatly assisted by computers. Furthermore, we plan to collect more data under harsh conditions such as nighttime, rainy weather, etc.

ACKNOWLEDGMENT

The authors thank the members of Multimedia Processing Lab for testing the annotation tool and providing a lot of useful suggestions, and Dr. Sunghoon Hong for guiding this project. The authors are also grateful to the anonymous reviewers for providing valuable comments that helped improve the quality of this article.

REFERENCES

[1] A. De La Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, “Road traffic sign detection and classification,” *IEEE Trans. Ind. Electron.*, vol. 44, no. 6, pp. 848–859, Dec. 1997.

[2] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, “Detection of traffic signs in real-world images: The German traffic sign detection benchmark,” in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2013, pp. 1–8.

[3] Y. Rehman, I. Riaz, X. Fan, and H. Shin, “D-patches: Effective traffic sign detection with occlusion handling,” *IET Comput. Vis.*, vol. 11, no. 5, pp. 368–377, 2017.

[4] G. Bak and D. Kim, "Traffic light recognition with HUV-histogram from daytime driving-view images," in *Proc. 17th Int. Conf. Control Autom. Syst.*, Jeju, South Korea, 2017, pp. 1099–1102.

[5] M. P. Philipsen, M. B. Jensen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Traffic light detection: A learning algorithm and evaluations on challenging dataset," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Las Palmas, Spain, 2015, pp. 2341–2345.

[6] K. Behrendt, L. Novak, and R. Botros, "A deep learning approach to traffic lights: Detection, tracking, and classification," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Singapore, 2017, pp. 1370–1377.

[7] T. Chen, Z. Chen, Q. Shi, and X. Huang, "Road marking detection and classification using machine learning algorithms," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Seoul, South Korea, 2015, pp. 617–621.

[8] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.

[9] S. Lee *et al.*, "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, 2017, pp. 1965–1973.

[10] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2015, pp. 91–99.

[12] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 2110–2118.

[13] *HD Maps for Autonomous Driving and Driver Assistance|HERE*. Accessed: Mar. 3, 2020. [Online]. Available: https://www.here.com/products/automotive/hd-maps

[14] *HD map|TomTom*. Accessed: Mar. 3, 2020. [Online]. Available: https://www.tomtom.com/products/hd-map/

[15] *Rethinking Maps for Self-Driving*. Accessed: Aug. 31, 2020. [Online]. Available: https://medium.com/lyftlevel5/https-medium-com-lyftlevel5-rethinking-maps-for-self-driving-a147c24758d6

[16] *The Why and How of Making HD Maps for Automated Vehicles*. Accessed: Aug. 31, 2020. [Online]. Available: https://newsroom.intel.com/articles/why-how-making-hd-maps-automated-vehicles/

[17] *Lvl5, Inc. Computer Vision, HD Maps, and Data for Autonomous Cars*. Accessed: Aug. 31, 2020. [Online]. Available: https://lvl5.ai/hd-mapping.html

[18] *TRI-AD and CARMERA Team up to Build High Definition Maps for Automated Vehicles Using Camera Data*. Accessed: Aug. 31, 2020. [Online]. Available: https://global.toyota/en/newsroom/corporate/26879165.html

[19] *HD Maps for the Masses. Comma.AI Made a Promise on Twitter | by Comma AI | Medium*. Accessed: Aug. 31, 2020. [Online]. Available: https://medium.com/@comma_ai/hd-maps-for-the-masses-9a0d582dd274

[20] *Open Autodrive*. Accessed: Aug. 31, 2020. [Online]. Available: https://www.openautodrive.org

[21] *National Geographical Institutes Precision Map*. Accessed: Mar. 3, 2020. [Online]. Available: http://map.ngii.go.kr/ms/pblictn/preciseRoadMap.do

[22] S. Ham, J. Im, M. Kim, and K. Cho, "Construction and verification of a high-precision base map for an autonomous vehicle monitoring system," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 11, p. 501, 2019.

[23] V. Ilci and C. Toth, "High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation," *Sensors*, vol. 20, no. 3, p. 899, 2020.

[24] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.

[25] A. Arcos-García, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332–344, Nov. 2018.

[26] R. Qian, Q. Liu, Y. Yue, F. Coenen, and B. Zhang, "Road surface traffic sign detection with hybrid region proposal and fast R-CNN," in *Proc. 12th Int. Conf. Natural Comput. Fuzzy Syst. Knowl. Discov. (ICNC-FSKD)*, Changsha, China, 2016, pp. 555–559.

[27] J. Kim, H. Cho, M. Hwangbo, J. Choi, J. Canny, and Y. P. Kwon, "Deep traffic light detection for self-driving cars from a large-scale dataset," in *Proc. IEEE 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, 2018, pp. 280–285.

[28] W. H. Lee, K. Jung, C. Kang, and H. S. Chang. (2021). *SAIT Traffic Landmark Dataset*. [Online]. Available: https://dx.doi.org/10.21227/gar1-z597

[29] U. Krey and A. Owen, *Basic Theoretical Physics—A Concise Overview*. Berlin, Germany: Springer, 2007.

[30] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EP*n*P: An accurate $O(n)$ solution to the P*n*P problem," *Int. J. Comput. Vis.*, vol. 81, no. 2, pp. 155–166, 2008.

[31] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, 1965.

[32] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder–Mead simplex method in low dimensions," *SIAM J. Optim.*, vol. 9, no. 1, pp. 112–147, 1998.

[33] K. Lim, Y. Hong, Y. Choi, and H. Byun, "Real-time traffic sign recognition based on a general purpose GPU and deep-learning," *PLoS ONE*, vol. 12, no. 3, 2017, Art. no. e0173317.

[34] Z. Xiao, D. Yang, T. Wen, K. Jiang, and R. Yan, "Monocular localization with vector HD map (MLVHM): A low-cost method for commercial IVs," *Sensors*, vol. 20, no. 7, p. 1870, 2020.

[35] J. K. Suhr, J. Jang, D. Min, and H. G. Jung, "Sensor fusion-based low-cost vehicle localization system for complex urban environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1078–1086, May 2017.

[36] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: arXiv:1704.04861.

[37] T. Tieleman and G. Hinton, "Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude," *COURSERA Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

[38] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[39] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 740–755.

[40] *Convention on Road Traffic of 1968 and European Agreement Supplementing the Convention (2006 Consolidated Versions)*, UNEC Eur. Transport Division, Geneva, Switzerland, 2007.

[41] M. Lopez-Montiel, Y. Rubio, M. Sánchez-Adame, and U. Orozco-Rosas, "Evaluation of algorithms for traffic sign detection," in *Proc. SPIE Opt. Photon. Inf. Process. XIII*, vol. 11136, 2019, pp. 135–151.

WON HEE LEE received the B.S. and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2008 and 2015, respectively.

In 2015, he joined Samsung Advanced Institute of Technology, Suwon, South Korea, where he is a Staff Researcher. He was the recipient of a Fellowship from the Korea Foundation for Advanced Studies during his doctoral study. His major research interests include computer vision, image processing, and machine learning.

KYUNGBOO JUNG received the B.S. degree in electronic engineering from the Kumoh National Institute of Technology, Gumi, South Korea, in 2004, and the M.S. and Ph.D. degrees from the Division of Electrical and Computer Engineering, Hanyang University, Seoul, South Korea, in 2006 and 2012, respectively.

He is currently a Principal Researcher with the Samsung Advanced Institute of Technology of Samsung Electronics. His research interests include computer vision, artificial intelligence, augmented reality, and 3-D reconstruction.

CHULWOO KANG received the B.S., M.S., and Ph.D. degrees in mechanical and aerospace engineering from Seoul National University, Seoul, South Korea, in 2007, 2009, and 2014, respectively.

He is currently serving as a Staff Researcher with the Samsung Advanced Institute of Technology of Samsung Electronics, Suwon, South Korea. His research interests include sensor fusion, localization, target tracking, perception, image processing, and machine learning for advanced vehicle applications.

HYUN SUNG CHANG (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2012.

From 1999 to 2004, he worked with the Electronics and Telecommunications Research Institute, Daejeon, South Korea, as a member of the Research Staff. From 2012 to 2013, he was with the Computer Science and Artificial Intelligence Laboratory, MIT, as a Postdoctoral Associate. In 2013, he joined the Samsung Advanced Institute of Technology, Suwon, South Korea, where he is a Principal Member of the Research Staff. During his doctoral study, he was a recipient of the Kwanjeong Educational Foundation Fellowship. His major research interests include computer vision, machine learning, and computational imaging.

Dr. Chang is a member of ACM.