A Convolutional Capsule Network for Traffic-Sign Recognition Using Mobile LiDAR Data With Digital Images

Haiyan Guan, *Senior Member, IEEE*, Yongtao Yu, *Member, IEEE*, Daifeng Peng, Yufu Zang, Jianyong Lu, Aixia Li, and Jonathan Li, *Senior Member, IEEE*

***Abstract*— Traffic-sign recognition plays an important role in road transportation systems. This letter presents a novel two-stage method for detecting and recognizing traffic signs from mobile Light Detection and Ranging (LiDAR) point clouds and digital images. First, traffic signs are detected from mobile LiDAR point cloud data according to their geometrical and spectral properties, which have been fully studied in our previous work. Afterward, the traffic-sign patches are obtained by projecting the detected points onto the registered digital images. To improve the performance of traffic-sign recognition, we apply a convolutional capsule network to the traffic-sign patches to classify them into different types. We have evaluated the proposed framework on data sets acquired by a RIEGL VMX-450 system. Quantitative evaluations show that a recognition rate of 0.957 is achieved. Comparative studies with the convolutional neural network (CNN) and our previous supervised Gaussian–Bernoulli deep Boltzmann machine (GB-DBM) classifier also confirm that the proposed method performs effectively and robustly in recognizing traffic signs of various types and conditions.**

***Index Terms*— Convolutional capsule network, convolutional neural network, mobile LiDAR point clouds, traffic signs.**

I. Introduction

TRAFFIC-SIGN recognition is critical for transportation agencies to manage and monitor the status and usability of traffic signs [1], [2]. In addition, intelligent traffic-related applications, such as autonomous driving, require accurate localization and recognition of traffic signs for timely and accurate response in different driving situations. The mobile laser scanning or mobile LiDAR technology provides a promising solution to transportation-related surveys [3]–[6]. The current mobile LiDAR system is an integration of multiple sensors, including laser scanners and digital cameras [7]; therefore, point clouds provide accurate geometrical information, while digital images detail rich spectral information, which contributes to accurate detection and recognition of traffic signs.

The existing algorithms apply the geometric and spatial features of traffic signs, such as shape, position, and reflectance [8]–[10], or learn these features automatically [11]–[12] to achieve the traffic-sign detection tasks. Traffic-sign recognition is commonly achieved by integrating imagery data and point clouds together. Generally, these algorithms follow a two-step procedure—traffic-sign detection using LiDAR point clouds [8], [12], [13] and traffic-sign recognition using digital images [13]–[16]. In traffic-sign detection, most methods detect traffic signs from point clouds using the following geometrical and spatial attributes: topology, intensity, and geometrical dimension, relations, and shape. For example, the traffic-sign surfaces have a strong reflectance intensity, which guides the road users for safe driving. Traffic signs are limited to certain sizes and shapes (e.g., rectangle, circle, and triangle). These attributes are used for successfully detecting traffic signs from point clouds.

After that, the detected traffic-sign points are transformed into the camera coordinate system to obtain the corresponding traffic-sign patches. The traffic-sign recognition tasks are commonly used using machine learning or deep learning algorithms. Some machine learning methods, such as support vector machine (SVM) [8] and SVM-based weakly supervised metric learning (WSMLR) [16], are most commonly used in the imagery-based traffic-sign recognition tasks in the past years. However, these machine learning methods require the manually designed features that are subjective and mainly rely on the operator's prior knowledge and experience. In contrast, the deep learning methods, such as Gaussian–Bernoulli deep Boltzmann machine (GB-DBM) model [13] and deep neural networks (DNNs) [14], can automatically abstract high-level feature representations from voluminous data samples, which have become attractive in traffic-sign recognition. These deep learning methods are proven to generate superior experimental results.

Manuscript received May 9, 2019; revised July 21, 2019 and August 29, 2019; accepted August 31, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 41671454, Grant 41971414, Grant 61603146, Grant 41801386, and Grant 41701529, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20160427 and Grant BK20180797, in part by the Natural Science Research in Colleges and Universities of Jiangsu Province under Grant 16KJB520006, and in part by the Natural Science Foundation of Zhejiang Province under Grant LQ15D010001. *(Corresponding author: Haiyan Guan.)*

H. Guan, D. Peng, and Y. Zang are with the School of Remote Sensing and Geomatics Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: guanhy.nj@nuist.edu.cn; daifeng@nuist.edu.cn; 3dmapzangyufu@nuist.edu.cn).

Y. Yu is with the Faculty of Computer and Software Engineering, Huaiyin Institute of Technology, Huaian 223003, China (e-mail: allennessy.yu@gmail.com).

J. Lu is with the Institute of Space Weather, School of Math and Statistics, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: jylu@nuist.edu.cn).

A. Li is with the College of Surveying and Municipal Engineering, Zhejiang University of Water Resources and Electric Power, Hangzhou 310018, China (e-mail: liax_zj@sina.com).

J. Li is with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@uwaterloo.ca).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2019.2939354

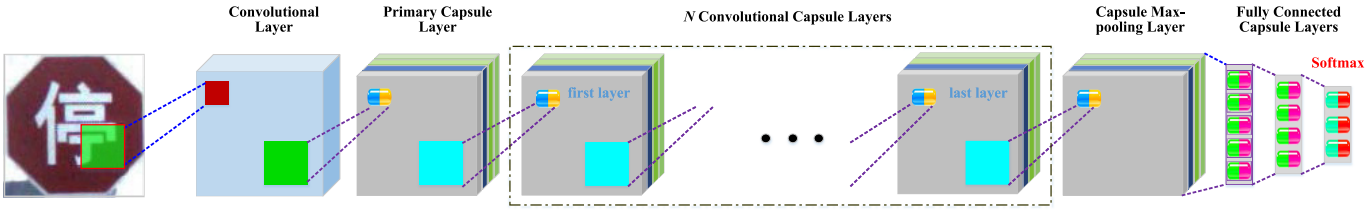

Fig. 1. Architecture of the convolutional capsule network.

Therefore, a generic framework for directly recognizing traffic signs from mobile LiDAR data and digital images can improve the robustness and reliability of the traffic-sign recognition tasks. The proposed framework is composed of 1) traffic-sign detection, which functions to extract potential traffic-sign regions, and 2) traffic-sign recognition, where a convolutional capsule network classifies the generated traffic-sign image patches into different types. The traffic-sign detection method is detailed in our previous work [13]. In this letter, we focus on traffic-sign recognition and propose a novel convolutional capsule network to recognize traffic signs of different categories. The remainder of this letter is organized as follows: Section II details the two-step traffic-sign detection and recognition method. Section III reports and discusses the experimental results of traffic-sign recognition. Section IV gives the concluding remarks.

II. Method

The proposed traffic-sign recognition method adopts a two-stage processing strategy. In the first stage, the geometrical features and attributes, provided by mobile LiDAR data, are first used to extract the traffic-sign interest regions. In the second stage, the extracted region proposals are projected onto the registered digital images to obtain their corresponding image patches, which are further fed into a convolutional capsule network to classify them into different categories of traffic signs. In Sections II-A and II-B, we will describe the traffic-sign detection and recognition framework in detail.

A. Traffic-Sign Detection Based on Geometrical Features and Attributes

This stage aims to extract the traffic-sign interest regions from mobile LiDAR data. Traffic signs usually stand out from their environments due to their special characteristics, such as shape, intensity, and color. The mobile LiDAR data provide accurate positional and intensity information of traffic signs; therefore, to facilitate traffic-sign detection, with the features and prior knowledge of traffic signs, an analysis is performed using the following factors: pole height (A_{PH}), road width (A_{RW}), intensity (A_{I}), geometrical structure (A_{G}), and traffic-sign size (A_{A}). To improve the processing efficiency when dealing with voluminous mobile LiDAR data, a supervoxel segmentation strategy is performed on the points. In our previous work, we achieved a detection accuracy of 86.8% and an advantageous computing performance (e.g., around 1 h for processing around 1 billion points). The comparative experiments have demonstrated the overall performance of our previous traffic-sign detection method. After traffic-sign detection, the detected traffic-sign points are projected onto the images to obtain the traffic-sign region proposals. Thus, in this letter, we adopt our previous traffic-sign detection method and focus on traffic-sign recognition from the detected traffic-sign region proposals.

B. Traffic-Sign Recognition Using Convolutional Capsule Network

To recognize traffic signs from the segmented image patches, we construct a convolutional capsule network. The capsule network, first proposed in [17] for classification tasks, is composed of entity-oriented vectorial capsules, which differs from the conventional CNNs that use scalar neurons to encode the probabilities of the existence of specific features. A capsule can be viewed as a vectorial combination of a set of neurons [17]. For a capsule, its instantiation parameters represent a specific entity type and its length represents the probability of the existence of that entity. The capsule networks have been demonstrated to be powerful and robust in various classification tasks. Thus, to obtain promising traffic-sign recognition performance, we extend the original capsule network (containing a conventional convolutional layer, a primary capsule layer, and a fully connected capsule layer) to construct a multi-layer convolutional capsule network.

Fig. 1 shows the architecture of our proposed multi-layer convolutional capsule network, which contains a conventional convolutional layer, a primary capsule layer, N convolutional capsule layers, a capsule max-pooling layer, and three fully connected capsule layers. Similar to the operations in a CNN model, the conventional convolutional layer uses convolution operations to extract low-level features from the input image patches. These features are further encoded into high-order capsules to represent different levels of entities. The conventional convolutional layer adopts the widely used rectified linear unit (ReLU) as the activation function to nonlinearly transform the outputs.

The primary capsule layer converts the low-level scalar feature representations in the convolutional layer into high-order vectorial capsule representations. This conversion is based on a conventional convolution operation sliding on the convolutional layer. Denote N_f as the number of feature maps in the primary capsule layer and C_d as the dimension of a capsule. A total of $N_f \times C_d$ different convolution kernels are performed on the convolutional layer, leading to $N_f \times C_d$ feature maps. After convolution operations, the generated feature maps are organized into N_f groups, each of which contains C_d feature maps, and further form a C_d-dimensional capsule at each position. As a result, in the primary capsule layer, the N_f capsules at each position are generated to encode different properties of an entity. Such that, the low-level scalar feature representations are converted into high-order vectorial entity representations. The capsules can estimate the probability of the existence of a specific entity through

the vector length, as well as depicting the orientation of the entity through the instantiation parameters. Thus, the vectorial capsule formulation contributes to detecting a feature and further to learning and detecting its variants.

The N convolutional capsule layers extract the high-order capsule features from low-order capsules by performing local convolution operations on a group of capsules and representing their features using a new capsule. For the capsules in the convolutional capsule layers, the total input to a capsule j is a weighted sum over all predictions from the capsules within the convolution kernel in the layer below

$$\boldsymbol{C}_j = \sum\nolimits_i a_{ij} \cdot \overline{\boldsymbol{U}_{j|i}} \tag{1}$$

where $\boldsymbol{C}_j$ is the total input to capsule j; a_{ij} is the coupling coefficient, indicating the degree of contribution that capsule i in the layer below activates capsule j; $\overline{\boldsymbol{U}_{j|i}}$ is the prediction from capsule i to capsule j and it is defined as follows:

$$\overline{\boldsymbol{U}_{j|i}} = \mathbf{W}_{ij} \cdot \boldsymbol{U}_i \tag{2}$$

where $\boldsymbol{U}_i$ is the output of capsule i. $\mathbf{W}_{ij}$ is the transformation matrix on the edge connecting capsules i and j. Specifically, the coupling coefficients between capsule i and all its connected capsules in the layer above sum to 1 and are determined by a dynamic routing process [17]. The dynamic routing process considers both the length of a capsule (i.e., the probability of the existence of an entity) and its instantiation parameters (i.e., the orientation of the entity) to activate another capsule. This is quite different from the classical CNN models that take into account only the probability. As a result, the capsule networks are more powerful and robust to abstract the intrinsic features of the objects. As mentioned above, the capsule length is used to predict the probability of the existence of an entity. Thus, for the convolutional capsule layers, the nonlinear "squashing" function [17] is adopted as the activation function, by which the capsules with short vectors result in low probability estimations and capsules with long vectors result in high probability estimations, whereas their orientations remain unchanged. The nonlinear squashing function is defined as follows:

$$\boldsymbol{U}_j = \frac{\|\boldsymbol{C}_j\|^2}{1 + \|\boldsymbol{C}_j\|^2} \cdot \frac{\boldsymbol{C}_j}{\|\boldsymbol{C}_j\|}. \tag{3}$$

By such a conversion, the capsules with short lengths are narrowed down to a length close to zero and the capsules with long lengths are shrunk to a length close to one.

The capsule max-pooling layer uses max-pooling operations, similar to the pooling operations in the CNN models, to perform feature down-sampling to reduce the network size. To this end, we adopt a max-pooling kernel with a size of $M_k \times M_k$. This kernel is slided on the feature maps of the last convolutional capsule layer with a stride of M_k. Within the $M_k \times M_k$ kernel in each feature map, only the capsule with the longest vector is retained and the others are ignored. In this way, the number of capsules and the network size are dramatically reduced, and thus the salient and representative capsules are selected. The selected capsules are further connected to a fully connected capsule layer to analyze the global features.

The three fully connected capsule layers consider all the capsules in the layer below to construct a high-order entity abstraction from a global perspective. The first fully connected capsule layer is obtained using a set of global capsule convolution kernels performing on the capsule max-pooling layer. Similarly, the dynamic routing process between two fully connected capsule layers is used to cast predictions and activate the capsules. In addition, the squashing function is used to normalize the outputs of the capsules to ensure that the shorter the capsules' lengths, the lower the probability estimations; whereas the longer the capsules' lengths, the higher the probability estimations. The last fully connected capsule layer is a softmax layer for classification purposes. The softmax layer is composed of V class-oriented capsules for encoding different categories of traffic signs and the background. We use the capsule length in the softmax layer to represent the probability of a traffic-sign image patch being an instance of a specific category (forbidden or warning). The category label of a traffic-sign image patch is defined as follows:

$$L^* = \arg\max_k \|\boldsymbol{U}_k\| \tag{4}$$

where $\boldsymbol{U}_k$ is the output of a capsule in the softmax layer.

The parameters in the convolutional capsule network are iteratively refined through the error backpropagation process. To effectively train the convolutional capsule network toward classification tasks, the margin loss [17] is used as the objective function to direct the error backpropagation process. For class k, the margin loss L_k is defined as follows:

$$L_k = T_k \cdot \max\left(0, m^+ - \|\boldsymbol{U}_k\|\right)^2 + \eta(1 - T_k) \cdot \max\left(0, \|\boldsymbol{U}_k\| - m^-\right)^2 \tag{5}$$

where $T_k = 1$ if and only if a training sample belongs to class k; otherwise, $T_k = 0$. m^+ and m^- are, respectively, the lower bound for the probability of a training sample being an instance of class k and the upper bound for the probability of a training sample not belonging to class k. They are configured as $m^+ = 0.9$ and $m^- = 0.1$. η is a weight regularization factor, which is set to be 0.5 by default. The total loss of the convolutional capsule network is defined as the sum of the losses of all class-oriented capsules on all training samples.

III. Results and Discussion

A. Data Set

The test data were collected by a RIEGL VMX-450 system, which is composed of two RIEGL VQ-450 laser scanners, four charge-coupled device (CCD) cameras, a set of Applanix POS LV 520 processing systems containing two global navigation satellite system (GNSS) antennas, an initial measurement unit (IMU), and a wheel-mounted distance measurement indicator (DMI). The survey was conducted along Huandao Road from Xiamen University to the International Conference and Exhibition Center (ICEC) in Xiamen Island, Xiamen, China. The surveyed area is a typical tropical urban environment with high buildings, dense vegetation, and traffic signposts along the surveyed road. Table I lists two scanned point cloud data, covering 10- and 11-km-long road sections, respectively. Fig. 2 provides a close view of the point cloud data of the test scene.

In traffic-sign detection, the parameters, pole height (A_{PH}), road width (A_{RW}), and traffic-sign size (A_{A}), were set to be 1.0 m, 12.0 m, and 0.2 m^2, respectively. The intensity

TABLE I
DESCRIPTION OF THE TWO MOBILE LIDAR SAMPLES

Dataset	Number of points	Length of road section
HDR	1,347,044,219	9922 m
HCR	1,400,438,347	10822 m

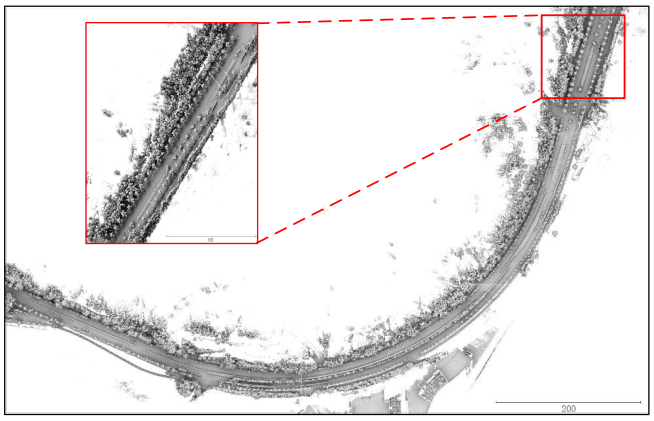

Fig. 2. Close view of the test data.

threshold (A_I) was estimated from the selected traffic signs in the surveyed area. The geometrical structure (A_G) was defined as planar. The surveyed road is a coastal landscape road, with several smooth turns, containing roughly 40 traffic-sign categories, according to functionality. The two data sets contain a total of 1268 traffic signs. The traffic-sign detection method, detailed in [13], extracted 1162 traffic signs, including 1101 correctly detected traffic signs and 61 nontraffic signs. The detection accuracy is 86.8%. As mentioned in our previous work, the detection errors were caused by incompletely scanned traffic signs and strong reflectance from attached advertising boards.

We downloaded 143 360 standard traffic signs from the Ministry of Transport, China, as the training data to train the convolutional capsule network. The detected traffic sign data sets contain 1101 traffic-sign images and 61 nontraffic sign objects. We manually labeled the 1162 image patches (containing 35 types of traffic signs and 1 type of background) of different image conditions as the reference data to evaluate the performance of our traffic-sign recognition method. All the training images and detected traffic-sign images were resized to a size of $M \times M$ pixels. To balance recognition performance and computational burden, we empirically set the image size at $M = 60$ pixels.

B. Data Training

The convolutional capsule network was trained using the Adam optimizer [18]. Before training, we randomly initialized all layers of the convolutional capsule network by drawing parameters from a zero-mean Gaussian distribution with a standard deviation of 0.01. The exponential decay rates for controlling the exponential moving averages of the gradient (the first moment) and the squared gradient (the second moment) were configured as 0.9 and 0.999, respectively. The learning rate was set at 0.001. The size of each training batch was configured to be 32 on each GPU. The network parameters can be trained by a total number of 2000 epochs in an end-to-end manner. To improve the efficiency of the capsule network, N convolutional capsule layers were added to extract the entity features from the input image. The more the number of convolutional capsule layers, the higher the levels of the extracted features. However, with an increase in the number of convolutional capsule layers, the computational complexity grows greatly. To tradeoff the feature extraction performance and the computational efficiency, we set $N = 3$. For dynamic routing to determine the coupling coefficients, we used three routing iterations, which was enough to obtain promising performance. To encode a proper entity representation, the dimension of a capsule was designed to be 16 for all capsule layers. Our framework took 32 h to obtain data satisfactory training results.

C. Traffic-Sign Recognition

This test set contained 1162 traffic-sign image patches covering 35 different categories of traffic signs and the background. At the test stage, the test images were fed into the convolutional capsule network to recognize traffic signs. For the output of the softmax layer of the convolutional capsule network, the capsule with the longest length corresponded to the category of an image patch. For an image patch labeled as a traffic sign, the length of the capsule encoded the probability of the image patch belonging to an instance of that traffic sign type. The proposed framework was capable of processing 18 traffic-sign patches per second.

To quantitatively evaluate the traffic-sign recognition accuracy, we used the recognition rate as the evaluation metric, which is defined as the proportion of correctly classified traffic signs. On average, our proposed framework achieved a traffic-sign recognition rate of 0.957 on the test set. Specifically, the traffic-sign images in the test set were captured in a wide range of condition variations in illumination, distance, background, view angle, and so on. Thus, the traffic signs exhibited with different qualities, distortions, and sizes. In addition, some traffic signs were pasted with other decorations or occluded by nearby objects. Fig. 3 presents some traffic sign samples of different special conditions. Fortunately, benefitting from the convolutional capsule network in characterizing highly salient and representative intrinsic features of the objects, our proposed framework obtained promising performance in recognizing such traffic signs.

D. Comparative Study

To further evaluate the performance of our proposed framework in recognizing traffic signs, we conducted comparative experiments with our previous method, a supervised GB-DBM classifier [13], and the CNN method [19]. Table II details the quantitative evaluation results obtained by these three methods with respect to the recognition rate. The data set includes 1101 traffic-sign images, covering 35 classes.

The training time for the supervised GB-DBM classifier was about 4.6 h. For the 1101 test samples, by means of the GB-DBM classifier, 1027 traffic signs of different shapes and conditions were correctly classified, whereas 74 traffic signs were misclassified. Quantitatively, a recognition rate of 93.3% was achieved. The CNN method presented that 1019 traffic signs were correctly detected out of the 1101 traffic sign samples, whereas 82 traffic signs were misclassified. Quantitatively, a recognition rate of 92.6% was achieved.

Fig. 3. Traffic-sign image patches.

TABLE II
TRAFFIC-SIGN RECOGNITION PERFORMANCE OBTAINED USING DIFFERENT METHODS

Method	Corrected traffic signs(#)	Missed traffic signs(#)	Recognition rate(%)
Proposed	1054	47	95.7
CNN	1019	82	92.6
GB-DBM	1027	74	93.3

Comparatively, the supervised GB-DBM classifier obtained similar classification accuracies to the CNN method. This is because both the CNN method and our previous method use high-level feature representations of traffic signs to improve the capability of handling various traffic-sign distortions, thereby achieving a good traffic-sign recognition performance.

As reflected in Table II, our proposed framework obtained a relatively better recognition rate than the other two methods. The lower performances of the supervised GB-DBM and CNN methods were mainly caused using the scalar-neuron-based feature representations. The scalar neurons of these two methods can only estimate the existing of specific features; however, the intrinsic properties and their variants cannot be well-exploited. However, our proposed convolutional capsule network can abstract high-level, salient, and distinctive entity representations using vectorial capsules. The capsules are more robust than scalar neurons in characterizing the intrinsic features of the objects. Through comparative analysis, we concluded that our proposed framework is feasible and effective for traffic-sign recognition from mobile LiDAR data and digital images.

IV. CONCLUSION

This letter has presented a complete processing chain for traffic-sign detection and recognition. This is a two-stage processing strategy composed of traffic-sign detection from mobile LiDAR points and traffic-sign recognition from digital images by a convolutional capsule network. With regard to the geometric features and attributes of traffic signs in the surveyed scene, we detect traffic signs from mobile LiDAR data. The extracted traffic-sign points are further projected onto the digital images to obtain traffic-sign patches. The convolutional capsule network is then used for recognizing different types of traffic signs. The contributions include the following: 1) this is the first study to apply capsule networks to detect traffic signs and 2) a novel deep convolutional capsule network with capsule convolution and max-pooling operations for complete and effective traffic-sign recognition.

We have examined our proposed framework on the RIEGL test sets. Quantitative evaluations showed that our proposed framework achieved a recognition rate of 0.957. In addition, comparative studies with two existing methods also confirmed that the proposed method was feasible and effective in correctly recognizing traffic signs using capsule networks.

REFERENCES

[1] J. M. Lillo-Castellano, I. Mora-Jiménez, C. Figuera-Pozuelo, and J. L. Rojo-Álvarez, "Traffic sign segmentation and classification using statistical learning methods," *Neurocomputing*, vol. 153, pp. 286–299, Apr. 2015.

[2] Y. Yu, J. Li, H. Guan, and C. Wang, "Automated extraction of urban road facilities using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2167–2181, Aug. 2015.

[3] H. Guan, J. Li, S. Cao, and Y. Yu, "Use of mobile lidar in road information inventory: A review," *Int. J. Image Data Fusion*, vol. 7, no. 3, pp. 219–242, Jun. 2016.

[4] R. Rybka, "Autodesk and bentley systems talk about mobile LiDAR," *LiDAR*, vol. 1, no. 2, pp. 41–44, 2011.

[5] K. Williams, M. J. Olsen, G. V. Roe, and C. Glennie, "Synthesis of transportation applications of mobile LiDAR," *Remote Sens.*, vol. 5, no. 9, pp. 4652–4692, 2013.

[6] J.-A. Beraldin, F. Blais, and U. Lohr, "Laser scanning technology," in *Airborne and Terrestrial Laser Scanning*, G. Vosselman and H. Mass, Eds. Scotland, U.K.: Whittles Publishing, 2010, pp. 1–42.

[7] B. Riveiro, L. Díaz-Vilariño, B. Conde-Carnero, M. Soilán, and P. Arias, "Automatic segmentation and shape-based classification of retro-reflective traffic signs from mobile LiDAR data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 1, pp. 295–303, Jan. 2016.

[8] C. Wen *et al.*, "Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 27–37, Jan. 2016.

[9] Y.-W. Seo, J. Lee, W. Zhang, and D. Wettergreen, "Recognition of highway workzones for reliable autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 708–718, Aug. 2014.

[10] A. Golovinskiy, V. G. Kim, and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 2154–2161.

[11] H. Gonzalez-Jorge, B. Riveiro, J. Armesto, and P. Arias, "Evaluation of road signs using radiometric and geometric data from terrestrial LiDAR," *Optica Appl.*, vol. 43, no. 3, pp. 421–433, 2013.

[12] J. Levinson *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2011, pp. 163–168.

[13] H. Guan, W. Yan, Y. Yu, L. Zhong, and D. Li, "Robust traffic-sign detection and classification using mobile LiDAR data with digital images," *IEEE J. Sel. Topics Appl. Earth Observat. Remote Sens.*, vol. 11, no. 5, pp. 1715–1724, May 2018.

[14] Á. Arcos-García, M. Soilán, J. A. Á. lvarez-García, and B. Riveiro, "Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems," *Expert Syst. Appl.*, vol. 89, pp. 286–295, Dec. 2017.

[15] Y. Yu, J. Li, C. Wen, H. Guan, H. Luo, and C. Wang, "Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 113, pp. 106–123, Mar. 2016.

[16] M. Tan, B. Wang, Z. Wu, J. Wang, and G. Pan, "Weakly supervised metric learning for traffic sign recognition in a LIDAR-equipped vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1415–1427, May 2016.

[17] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 3856–3866.

[18] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Rep.*, San Diego, CA, USA, 2015, pp. 1–15.

[19] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.