Zhang et al. Computational Urban Science (2021) 1:18
https://doi.org/10.1007/s43762-021-00019-6

Computational Urban Science

ORIGINAL PAPER Open Access

Automated detecting and placing road objects from street-level images

Chaoquan Zhang[1*], Hongchao Fan[1] and Wanzhi Li[2]

Abstract

Navigation services utilized by autonomous vehicles or ordinary users require the availability of detailed information about road-related objects and their geolocations, especially at road intersections. However, these road intersections are generally represented as point elements without detailed information, or are even not available in current versions of crowdsourced mapping databases including OpenStreetMap (OSM). This study proposes an approach to automatically detect road objects from street-level images and place them to correct locations according to urban rules. Our processing pipeline relies on two convolutional neural networks: the first one segments the images, while the second one detects and classifies the specific objects. Moreover, to locate the detected objects, we propose an attributed topological binary tree (ATBT) based on urban rules for each image in an image sequence to depict the coherent relations of topologies, attributes and semantics of the road objects. Then the ATBT is further matched with map features on OSM to determine the right placed location. The proposed method has been applied to a case study in Berlin, Germany. We validate the effectiveness of the proposed method on two object classes: traffic signs and traffic lights. Experimental results demonstrate that the proposed app roach provides promising results in terms of completeness and positional accuracy.

Keywords: Object placing, Attributed topological binary tree, Street-level images, OpenStreetMap, Completeness, Traffic lights, Traffic signs

1 Introduction

The rapid development of advanced driver assistance systems and autonomous vehicles in recent years has attracted the ever-growing interest in smart traffic applications. Such intelligent applications can provide detailed road asset inventories of all stationary objects, such as street furniture (traffic lights and signs, various poles, bench, etc.), road information (lanes, edges, shoulders, etc.), small façade elements (antennas, cameras, etc.), and other minor landmarks. However, these detailed road map productions are mainly generated by mobile mapping systems (MMS), which require high costs both in the investment of equipment and in labor-intensive data post-processing. In addition, data updating is again a huge challenge. For instance, official road maps suffer from a long update cycle that can last several months or even years (Kuntzsch et al., 2016).

The last decade has witnessed an explosion of geospatial data. An increasing number of crowdsourced geospatial data repositories/services allow volunteers to utilize information from various data sources when contributing data to a crowd-sourced platform. That is known as Volunteered Geographic Information (VGI) (Goodchild, 2007). Amongst them, OSM and Mapillary are the typical representatives of maps and street-level crowdsourcing platforms, respectively. The large amount of detailed map data provided by OSM not only enriches the data sources of map making, but also supports and promotes data-driven (Hachmann et al., 2018; Melnikov et al., 2016) and data-intensive (Chang et al., 2016; Gao et al., 2017) spatial analysis. Additionally, literature (Neis et al., 2012) has shown that OSM road data in Germany and Netherlands can be comparable to official data. With the introduction of Mapillary in 2014, it has

* Correspondence: chaoquan.zhang@ntnu.no
[1]Department of Civil and Environmental Engineering, Norwegian University of Science and Technology, Trondheim, Norway
Full list of author information is available at the end of the article

become the biggest and the most active crowdsourced street-level imagery platform around the world. Tens of billions of street view images covering millions of kilometres of roads and depicting street scenes at regular intervals are available (Solem, 2017).

Even though OSM has made remarkable achievements, it still has some drawbacks. For example, road intersections in OSM are mainly represented as point elements without any semantic information (e.g. traffic signs/lights), or are even not available for most cities/countries (Fig. 1). According to Ibanez-Guzman et al. (2010), a high percentage of traffic accidents occur at road intersections, which reflects the importance of road intersections for traffic safety. If we can provide more information about intersections to the relevant authorities and let them use this information together with trajectory data to optimize the setting of traffic lights or vehicle speeds, that may help reduce the incidence of traffic accidents to some extent. To the best of our knowledge, Mapillary submitted an additional layer to OSM where marked the traffic signs on the map. However, the locations of traffic signs in the layer differ greatly from the actual ones. Besides, multiple traffic signs with the same category would appear within a small area at the same time, which is obviously inconsistent with the actual situation. This may be related to the fact that Mapillary only adopted pure computer vision methods to detect the traffic signs without considering the correctness of their locations in the real world.

Considering all the above shortcomings, in this paper we aim to automatically detect and classify traffic lights/signs at road intersections by using deep learning method from street-level images, and localize their positions based on urban rules and proposed attributed topological binary trees (ATBT). In this way, we can further enrich the OSM data. Since these kinds of information are hard to be seen on satellite and aerial images

Fig. 1 Incomplete information at road intersections

and hence they cannot be mapped by volunteers on OSM, the proposed method provides a good solution for this issue. To summarize, the main contributions and innovations of our work are as follows:

- We propose a simple convolutional neural network, namely ShallowNet, for traffic sign classification, which is characterized by low model complexity, high detection accuracy and fast recognition speed.
- We propose 6 urban rules (or grammar) to assist the determination of the relative position and topological relationship of the road-related objects.
- Based on the urban rules, we propose an attributed topological binary tree (ATBT) for image sequences to effectively describe the coherent relations of topologies, attributes and semantics of the road objects.
- With the proposed ATBT, we can easily and accurately determine where the traffic lights and signs should be placed by matching with map features on OSM. Furthermore, our experiments show that the whole workflow performs well in terms of object completeness and positional accuracy.

The remainder of this paper is organized as follows. We first review some relevant state-of-the-art approaches in Section 2. Section 3 presents our complete detection and localization pipeline. A set of experimental analyses are presented in Section 4. Conclusions and future work are discussed at the end of this paper as Section 5.

2 Related work

Benefiting from the ubiquitous street view images accessible from Google Street View (GSV), Mapillary, etc., many efforts have been directed towards the intelligent use of them to assess urban greenery (Li et al., 2015; Li et al., 2018), to enhance existing maps with fine-grained segmentation categories (Mattyus et al., 2016), to explore urban morphologies by mapping the distribution of image locations (Crandall et al., 2009), to analyze the visual elements of urban space in terms of human perception (Zhang et al., 2018) and urban land use (Li et al., 2017). Furthermore, street view images have also been combined with aerial imagery to achieve tree detection/classification (Wegner et al., 2016), land use classification (Workman et al., 2017), and fine-grained road segmentation (Mattyus et al., 2016). Together with (Timofte & Van Gool, 2011), these methods rely on a simplified locally flat terrain model to evaluate object locations from street-level images.

The last ten years have witnessed the quick development of Convolutional Neural Network (CNN) and CNN-based image content analysis. It has been proven efficient in learning feature representations from a large-scale dataset (LeCun et al., 2015). And as a consequence, urban studies involving street-level images have been largely enhanced since it was proposed. By leveraging street view images, many studies employ deep learning for object detection and classification, as well as image semantic segmentation to monitor neighbourhood change (Naik et al., 2017), to quantify the urban perception at a global scale (Dubey et al., 2016), to estimate demographic makeup (Gebru et al., 2017), to predict the perceived safety responses to images (Naik et al., 2014), to predict the socio-economic indicators (Arietta et al., 2014), and to navigate without maps in a city (Mirowski et al., 2018). In contrast, less attention has been paid to extracting traffic elements within road intersections from street view imagery. Furthermore, all of these methods use GSV as input data, but GSV charges a fee after downloading a certain amount for free, which is no doubt not a good choice for teams or individuals with insufficient research funds. Therefore, we introduce Mapillary, a fully free, crowdsourced, almost real-time updated and ubiquitous street-level imagery, into our work.

In terms of localization, so far, several approaches have been made available to map particular types of objects from street view imagery: traffic lights (Jensen et al., 2016; Trehard et al., 2014), road signs (Soheilian et al., 2013), and manholes (Timofte et al., 2011). These methods determine the positions of the road assets from individual camera views based on position triangulation. All of them depend heavily on various visual and geometrical features to match when multiple objects appear in the same scene. As a result, the performance of these methods is poor when multiple identical objects exist at the same time. Hence, an improved method is proposed. Hebbalaguppe et al. (2017) describe the problem as an object recognition task, and then adopt a stereo-vision (Seitz et al., 2006) approach to estimate the object coordinates from sensor plane coordinates using GSV. However, different from GSV, Mapillary street view images do not contain any camera intrinsics and projective transformation in their EXIF information, and thus we cannot perform the camera calibration. In other words, we cannot apply the same method for traffic lights/signs localization using Mapillary images. Recently, Krylov et al. (2018) combine the use of monocular depth estimation and triangulation to enable automatic mapping of complex scenes with the simultaneous presence of multiple, visually similar objects of interest, and achieve the position precision of approximately 2 m.

In this study, we focus on the research of road intersections to enrich the objects related to OSM intersections, such as traffic signs and lights, and to locate them

for the reference of autonomous driving or navigation. We propose a complete pipeline to extract scene elements such as buildings, sky, roads, sidewalks, traffic lights and signs based on image semantic segmentation from road intersections images. For localization purposes, the hierarchy of semantic objects needs to be applied, as there are the coherent relations of topologies, attributes and semantics of the road objects. In further, together with the segmentation results, an attributed topological binary tree (ATBT) based on urban rules can be established to depict the topologies among road objects. These are then matched with map features on OSM. In the end, road objects can be localized as promising results.

3 Methodology

In this section, we discuss a complete pipeline for the localization of traffic lights and signs from image sequences at road intersections. The pipeline has the following three modules: (1) data preprocessing and cleaning module; (2) object segmentation and recognition module; (3) localization module. Figure 2 depicts the whole framework. The first module is for preparing preprocessed and cleaned data for the next two modules (see Section 3.1). The second module mainly extracts road-related information by using image semantic segmentation as well as object detection and classification (see Section 3.2). In the last module, an attributed topological binary tree (ATBT) is constructed to represent the relative position relation between extracted objects at the intersections and to locate the objects with urban rules (see Section 3.3). Ultimately, the located objects can be integrated to enrich the OSM data.

3.1 Data preprocessing and cleaning

The main purpose of the data module is to prepare data for the next two modules. First of all, the intersections are identified by using the DBSCAN algorithm (Ester et al., 1996) based on incomplete traffic lights existing in OSM data. The incompleteness is reflected in, for example, there should have had four traffic lights at the intersection but only one or two are marked in the OSM. Additionally, their positions are roughly estimated which means traffic lights are with lower positional accuracy. After identifying the intersections, selecting experimented intersections mainly obeyed two rules: (1) the intersections can be clearly seen in the Mapillary images; (2) the image sequences can be corrected well by employing the SfM algorithm. Second, all the available images can be downloaded by querying the relevant Mapillary application programming interface (API). Specifically, a bounding box in geographic coordinates that covered the entire Berlin has been calculated in advance. To form a request to the Mapillary API, we then construct an API URL in which includes a common server address, user's unique client ID, bounding box and other search parameters such as start time, end time, searching radius, etc. Third, a buffer is then set up for each road intersection to extract image sequences within the buffer. An image sequence refers to a trajectory of a user traveling along the street. For an intersection with four road branches, we are able to theoretically build four image sequences by merging multiple image sequences according to their geolocations because of four kinds of rough driving directions, i.e. west-east, east-west, south-north and north-south. In addition, camera location including latitude and longitude, and camera angle are extracted.

Furthermore, we have found that the GPS positions of image sequences often drift, which may be associated with the geographical environment during the shooting (for example, tall buildings or heavy tree canopies block the GPS signal), or it may be because the GPS receiver built in the camera itself is inaccurate. Fortunately, one of the big advantages of Mapillary is that street view images of the same road segment may be uploaded repeatedly by different volunteers. And there is a certain degree of overlap between the two adjacent images,

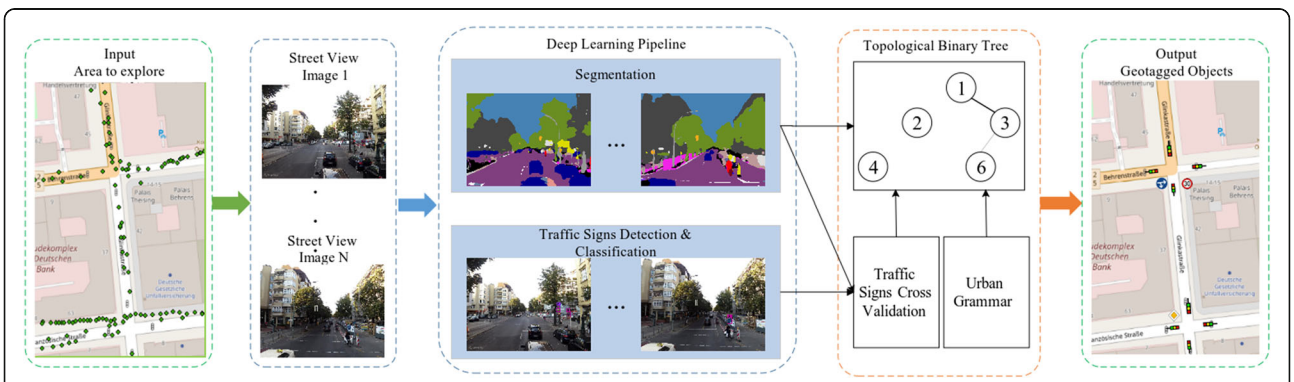


Fig. 2 Workflow of the methodology

which makes it possible for us to correct their shooting positions.

To reduce the error as much as possible and to improve the accuracy of localization, we employ a technique called Structure from Motion (SfM) (Snavely et al., 2008), as depicted in Fig. 3(a), to match features between images and reconstruct their surroundings in three-dimensional space to form point clouds. Each point has its position in three-dimensional space, so we can estimate the correct shooting positions of images along with the camera angles. As a result, these corrections can place misaligned images in their original positions as much as possible. In general, the more images feeding into the system from an area, the better the results could be. An SfM-corrected sequence is shown in Fig. 3(b). We can easily discover the original image shooting locations (green dots) swinging from one side of the road to the other in an "S" pose. Red dots symbolize the corrected locations, which now are fully aligned with roads. Additionally, if there are many overlaps between those images, these corrections can be very promising.

3.2 Object segmentation and recognition using deep learning

In theory, all road-related information can be extracted accurately from images only via semantic segmentation (see Section 3.2.1). Nevertheless, the quality of crowdsourced street-level images varies greatly, and hence it is difficult to ensure that all images can be segmented well, which would lead to inaccuracies or errors. Hence, in Section 3.2.2, we adopt an alternative strategy based on object detection to improve this problem.

3.2.1 Semantic segmentation using PSPNet

Image semantic segmentation is one of the key techniques used to understand a scene (Zhou et al., 2017), and is aimed at segmenting and recognizing object instances from images. Given an input image, the model can assign a class label for each pixel. One of the state-of-the-art semantic segmentation models with superior performance – PSPNet (Zhao et al., 2017) is applied in our study to perform object extraction. The PSPNet uses a new neural network sub-architecture, which retains global and local contextual information through a multi-scale representation of the previous convolutional layer's output. Because of the validated performance of the PSPNet trained on the PASCAL VOC 2012 (Everingham et al., 2010) and Cityscapes (Cordts et al., 2016) datasets, we are confident to segment road-related objects well by using PSPNet, such as buildings, sky, roads, sidewalks, traffic lights/signs, etc. These extracted objects will later be used as nodes of the attributed topological binary tree (ATBT).

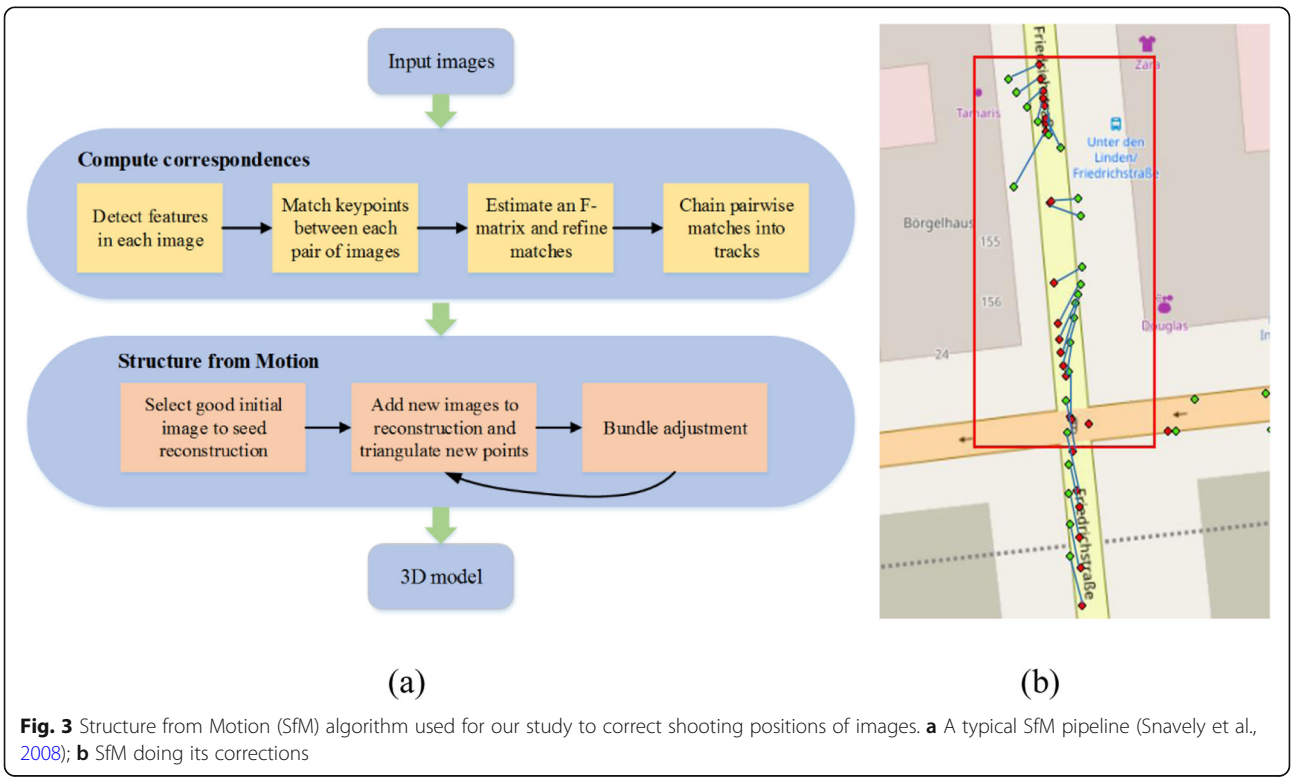


Fig. 3 Structure from Motion (SfM) algorithm used for our study to correct shooting positions of images. **a** A typical SfM pipeline (Snavely et al., 2008); **b** SfM doing its corrections

3.2.2 Object detection and classification using YOLOv3 and ShallowNet

After image semantic segmentation, we find that there are three limitations associated with semantic segmentation, especially for traffic signs. First, since we can only know this is a traffic sign through semantic segmentation, but we do not know which kind of traffic sign it belongs to. Second, if two signs are arranged together, semantic segmentation cannot identify them separately, which is not conducive to the supplement and enrichment of OSM semantic data. Third, our PSPNet model often misclassifies the isolation piles as traffic signs, or sometimes confuses two objects with similar features but they actually do not belong to the same category. The third limitation may be because the two objects have similar features (such as color, shape or texture), or the training dataset does not include such cases. As a result, the model did not learn the relevant features well.

Fortunately, object detection can address the above limitations. Taking into account the processing speed and detection accuracy, we choose YOLOv3 (Redmon & Farhadi, 2018) as our object detection model after some researches. Thus, we specially train a YOLOv3 model based on GTSDB (Stallkamp et al., 2012) dataset for detecting traffic signs, and then cross-validate the results of object detection and semantic segmentation to reduce errors and provide rich and effective attribute information for localization.

In our study, we not only need to know this is a traffic sign, but also need to know which kind of sign it belongs to. Consequently, in terms of traffic sign classification, we design a new shallow convolutional neural network called ShallowNet. As illustrated in Fig. 4, the network contains only five layers with weights; the first three are convolutional and the remaining two are fully-connected. The output of the last fully-connected layer is fed to a 45-way softmax which produces a distribution over the 45 class labels. We adopt batch normalization (Ioffe & Szegedy, 2015) right after each convolution and before ReLU non-linearity (Nair & Hinton, 2010) to speed up the convergence of model training. Additionally, to reduce the size of feature maps as much as possible, the convolutional layers involved in the network are all performed filling operation, and each convolution is followed by downsampling.

The first convolutional layer filters the $48 \times 48 \times 3$ input image with 64 kernels of size $7 \times 7 \times 3$. The second convolutional layer filters the output of the previous layer with 128 kernels of size $4 \times 4 \times 64$. The third convolutional layer has 300 kernels of size $4 \times 4 \times 128$ connected to the outputs of the second convolutional layer. Then we expand the feature map and form 1500 feature vectors into the fully-connected layer. Moreover, to reduce the overfitting of the network, we introduce dropout (Hinton et al., 2012) at the first fully-connected layer.

In general, our proposed network model, ShallowNet, is characterized by:

- Simple network structure and low model complexity. With few parameters, it is easy to be deployed to mobile or embedded devices.
- High accuracy. It can correctly recognize the type of traffic signs
- Fast recognition speed. Real-time object recognition can be achieved.

3.3 Object localization

Since Mapillary street view images do not contain any camera intrinsics in their EXIF information, it is impossible to calculate the projective transformation matrix and then perform camera calibration. In other words, we cannot apply photogrammetry methods for traffic lights/signs localization using Mapillary images.

After observing a large number of images of road intersections, we note that many images show a structure where buildings are on both sides of the road and a portion of the sky appears between them, traffic lights and signs being often placed at street corners, as well as pedestrians and vehicles appearing on the road. We can

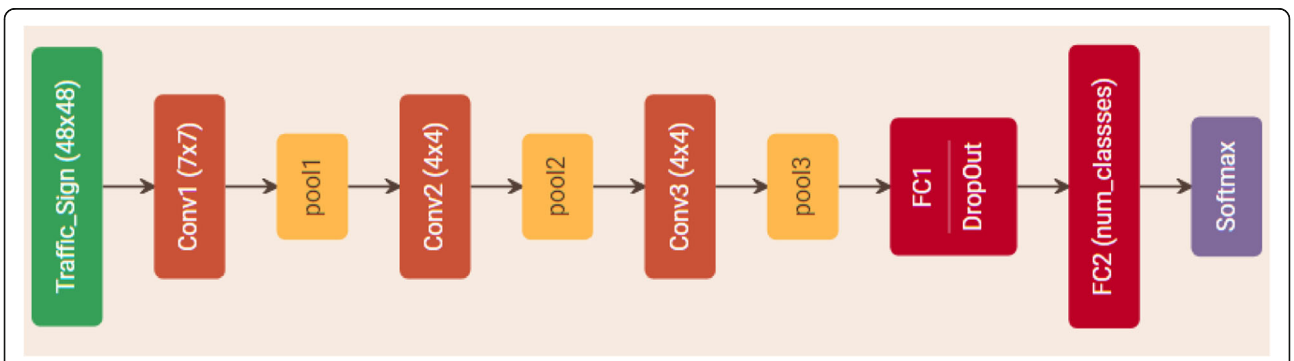


Fig. 4 Overview of our proposed ShallowNet

vaguely feel that there exist some certain arrangement rules between the objects in the images. Inspired by this, we propose a novel method to depict the coherent relations of topologies, attributes and semantics of the road objects at the intersections by establishing an attributed topological binary tree based on urban grammar (see Section 3.3.1). These objects (mainly traffic lights and signs) are then further matched with map features on OSM to determine the correctly placed location (see Section 3.3.2).

3.3.1 Attributed topological binary tree (ATBT) generation and updating

Taking the extracted objects through the image semantic segmentation as input, the ATBT can be created from top to bottom and from left to right. The left and right children of the binary tree can reflect the relative position relationship between the objects. We regard traffic lights, traffic signs and sidewalks as three types of nodes of the tree, and assign corresponding attributes to each type of node, such as centroid, area, height (optional), category, and role in the tree.

For traffic lights, there are two types of traffic lights: located on the sidewalk (low one) and located on the road (high one), which need to be recognized through following two urban rules (see Fig. 5):

(1). If the traffic light is surrounded by the sky, a ray (right red solid line in Fig. 5(b)) can be cast from the centroid of the segmented traffic light region downwards the road. If the distance between centroid and road surface is far more than twice the height of the tallest pedestrian (blue solid line in Fig. 5(b)), it can be inferred that this traffic light is at the road junction (i.e. high one), and its height is about 7 m (***another urban rule, searched from the Internet***).

(2). If the traffic light is surrounded by the buildings, a similar ray (left red solid line in Fig. 5(b)) can be cast from the centroid of the segmented traffic light region downwards the sidewalk. If the distance between centroid and sidewalk surface is less than or equal to twice the height of the tallest pedestrian, it can be inferred that this traffic light is located on the sidewalk (i.e. low one), and its height is about 4 m.

In fact, ***Rule1*** implies an "up and down" relationship, that is, traffic lights are surrounded by the sky and the sky is above the high traffic lights. Similar to ***Rule1***, ***Rule2*** also implies a "front and back" relationship, that is, the low traffic light is surrounded by buildings and buildings are behind the low traffic light.

As we can see from ***Rule2***, sidewalks are very important for our judgment. But in many cases, sidewalks are divided into multiple independent "blocks" by the pedestrian (as shown in Fig. 6(b)) according to the results of image semantic segmentation. In this case, it is necessary to judge whether the adjacent independent "sidewalk blocks" meet a certain distance threshold based on another empirical knowledge (i.e. the sidewalks on the same side are connected, ***Rule3***). If within this distance threshold, they are considered to be connected. Furthermore, many images do not capture the view of the whole intersections, but just a part of them as shown in Fig. 6(a). According to the urban rules, low traffic lights on the sidewalks tend to appear in pairs (***Rule4***). As long as there is a low traffic light on one side of the road, there definitely has another one on the other side of the road. This gives our topological binary tree the ability to reason.

Of course, there are also urban rules applicable to traffic signs. Since the study area of this paper is Berlin, Germany, we find that traffic signs at road intersections follow such patterns (***Rule5***, see Fig. 7): they either appear alone, or are usually close to the low traffic light above or both up and down, or arrange together. These are intrinsic combination patterns, and the distance between centroids of the internal objects of the combined pattern is within a small threshold.

(a) Original image (b) Segmented image

Fig. 5 Discrimination of different types of traffic lights based on urban grammar, which includes two rules. *Rule1*: (1) Surrounded by sky; (2) Distance > > height of 2 × max_pedestrian. *Rule2*: (1) Surrounded by buildings; (2) Distance <= height of 2 × max_pedestrian

(a) (b)

Fig. 6 a Only part of the intersection is photographed, and *Rule4* is summarized: low traffic lights appear in pairs. **b** A sidewalk is divided into multiple "blocks" by pedestrians, and *Rule3* is summarized: the sidewalks on the same side are connected

Finally, for each image in the image sequence, an ATBT can be established according to the results both from semantic segmentation and traffic sign detection/classification. The left subtree of the root node corresponds to the left side of the road, and the right subtree corresponds to the right side of the road. Additionally, for the convenience of computation, the node number of the tree is strictly in accordance with the node number of the complete binary tree. The left-right or top-bottom relationship between nodes is determined by the position of their centroids.

However, the first image of an image sequence was generally taken at the farthest location from the intersection, which may lead to some road objects not being segmented/detected and then affect the establishment of ATBT. Therefore, with the camera approaching the intersection, the image scene becomes clearer and can capture additional objects that are missed in previous images. Then, ATBT would dynamically update itself by comparing the difference between the current tree and the previous tree as well as combining with the urban rules, as illustrated in Fig. 8. The image numbered 1–3 are gradually approaching the intersection.

Meanwhile, the scene depth information is also considered during the ATBT establishment and updating. For example, there are two low traffic lights (labelled by No. 9 and 19 in the third segmented image) as shown in Fig. 8. They belong to a pair as described in ***Rule4*** *(*i.e. *low traffic lights on the sidewalks tend to appear in pairs)*. Their height should be the exactly same in reality, however, from the image, the No.19's height is obviously lower than the No.9's. This is because photo imaging follows the law that the object is big when near and small when far. Hence, in this case, we used the scene depth information and *Rule4* to infer that these two traffic lights should belong to a pair and should locate on the sidewalk.

In summary, as the image gets closer to the intersection, ATBT can dynamically update itself according to

Fig. 7 Four combined patterns of traffic signs and lights

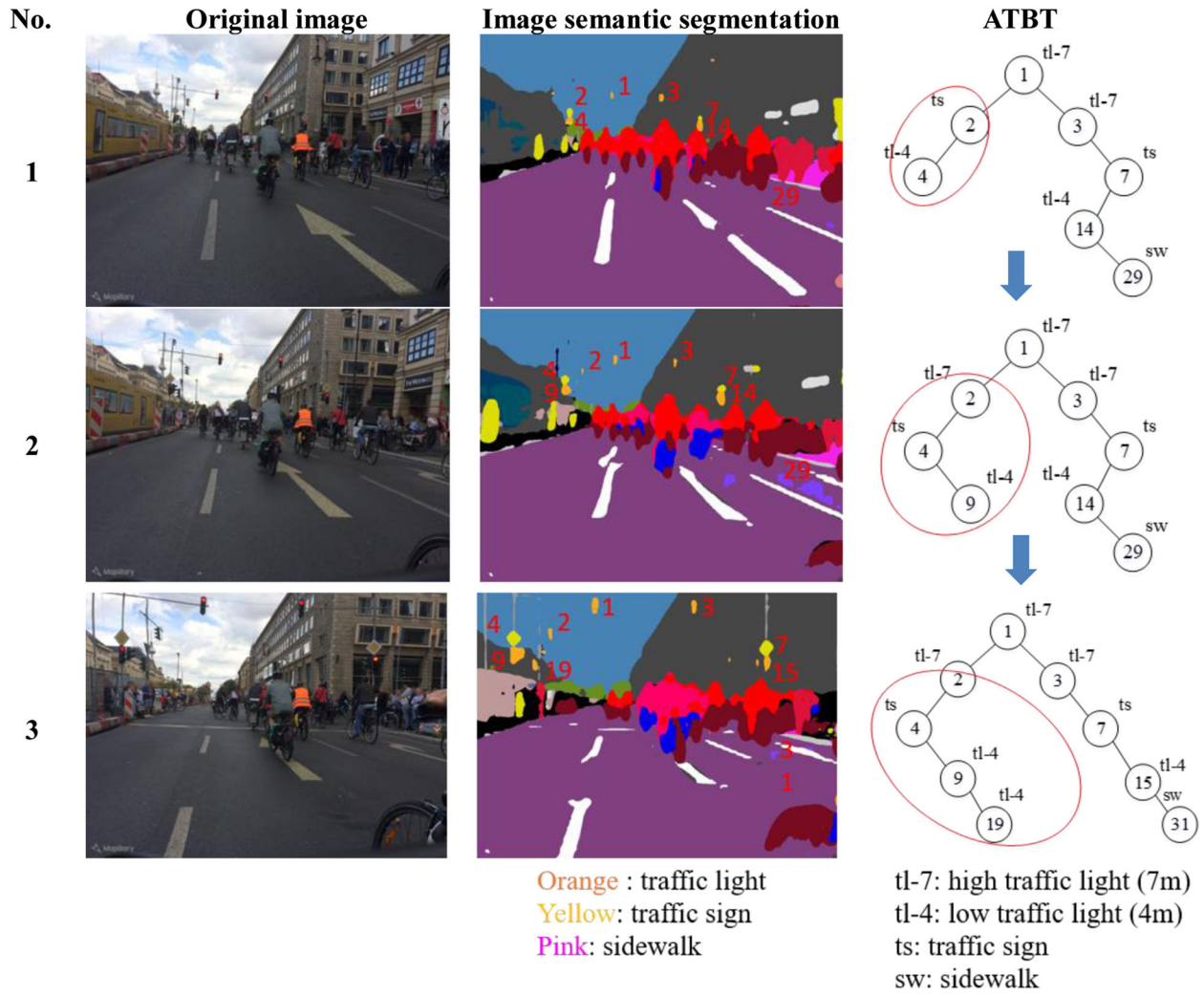


Fig. 8 Attributed topological binary tree (ATBT) generation and self-updating. The image numbered 1–3 are gradually approaching the intersection

the topological relationship of the objects, urban rules as well as scene depth information, and get the final ATBT.

3.3.2 Map matching

Based on the ATBT constructed earlier, we can use the shooting positions and camera angles provided by images as well as OSM footprints that are located around the intersections to match the left and right subtrees of the ATBT with the corresponding footprints. After that, the geographically placed locations of objects (e.g. traffic signs) in the real world can be determined.

Assuming there is an image sequence taken from west to east, the shooting positions of these images are represented by C1, C2 and C3 (as demonstrated in Fig. 9). Here, C1 is illustrated as an example. We take the red shooting point C1 as the centre of a circle, and draw the buffer with a radius of 26 m (determined by multiple experiments) to get footprints intersecting with the buffer. After calculating the distance from footprints to C1, we get that the yellow highlighted footprint is closest to the C1 (i.e. it corresponds to the right subtree of the ATBT), and similarly, the green highlighted footprint is closest to the C1 (i.e. it corresponds to the left subtree of the ATBT). From Fig. 9, the yellow and green highlighted footprints are indeed at the intersection, which indicates that the results we got are correct. In this way, the placed positions of traffic signs and lights can be determined.

We have inquired about the "Code for Urban Road Design", which clearly states that the minimum width of an ordinary sidewalk is 2 ~ 3 m (***Rule6***). Therefore, we place the low traffic lights and traffic signs about 2.5 m away from the corresponding footprint corner point (A1 or A2); the high traffic lights are placed at the midpoint

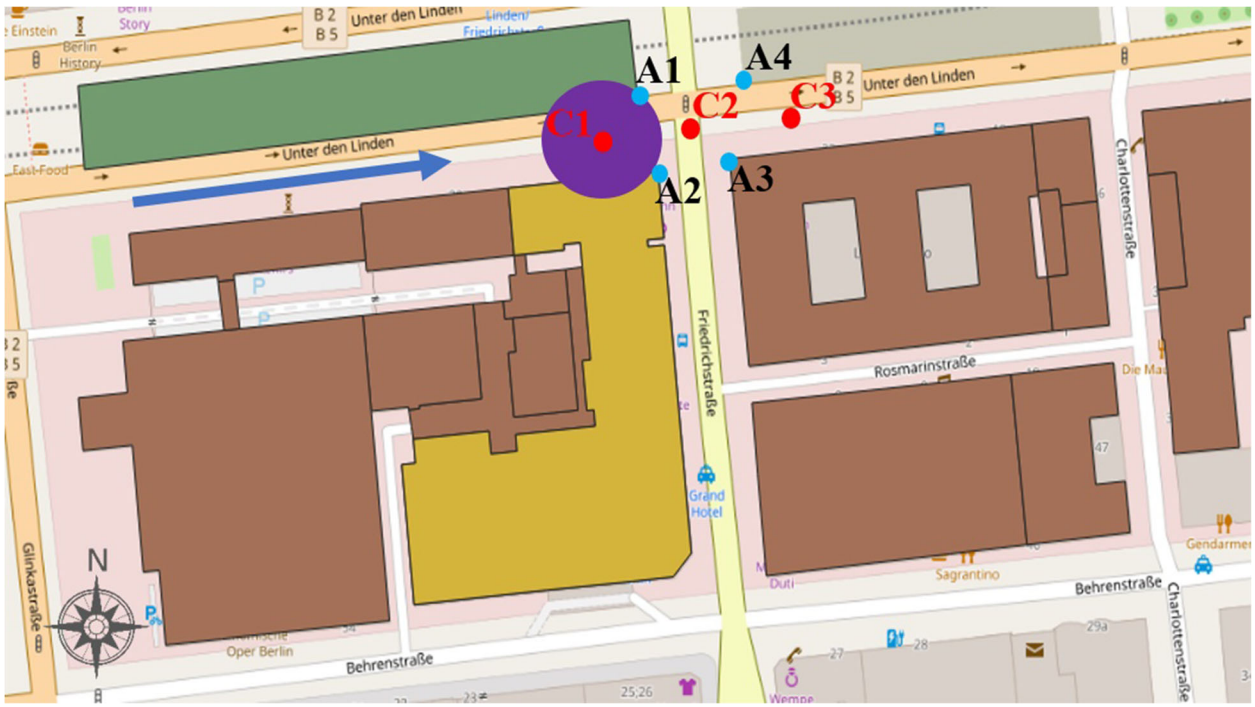


Fig. 9 Map matching between shooting point and OSM footprints

of connection between A1 and A2. In fact, this is not a precise localization, but it can indicate the approximate location of the traffic lights and signs.

The situation of C2 is a little bit complex. Since C2 is located in the middle of the intersection and none of the footprints is around it. If C2 is the centre of a circle and the corner points obtained by intersecting with footprints are A1 and A2, it indicates that C2 just passed one side of the intersection. Since the content of an image is always the scene in front of C2, but at this time A1 and A2 are behind the C2, so these two corners are not the corner points we want. Similarly, if the circle with C2 as centre intersects with footprints and yields A3 and A4 as their corner points, which are what we want because they are in front of C2.

Compared with the situation of C2, the situation of C3 is much simpler. Since C3 is about to leave the intersection area, the corner points that C3 intersecting with footprints are always behind it. This situation is not what we want as well.

For a more intuitive view of the urban rules used in this paper, we summarize and list them in Table 1 as shown below.

4 Experimental results

4.1 Study area and data

As the capital and largest city of Germany, Berlin was chosen as our study area. The study area has various intersection types, which range from the most common

Table 1 Summarized urban rules used in attributed topological binary trees (ATBT)

Rule No.	Description of the urban rules
Rule1	1) Traffic light is surrounded by sky; 2) distance between the traffic light and road surface is far more than twice the height of the tallest pedestrian. ***Conclusion***: high traffic lights
Rule2	1) Traffic light is surrounded by buildings; 2) distance between the traffic light and road surface is less than or equal to twice the height of the tallest pedestrian. *Conclusion*: low traffic lights
Rule3	The sidewalks on the same side are connected.
Rule4	Low traffic lights on the sidewalks tend to appear in pairs.
Rule5	Traffic signs in Germany either appear alone, or are usually close to the low traffic light above or both up and down, or arrange together.
Rule6	The minimum width of an ordinary sidewalk is 2 ~ 3 m.

intersections with three/four road branches to the complicated intersections, like roundabouts.

The datasets used in this study include OSM building footprints data, Mapillary street view images, Mapillary Vistas, German Traffic Sign Detection Benchmark (GTSDB), and German Traffic Sign Recognition Benchmark (GTSRB) (Stallkamp et al., 2011). The OSM building footprints data was collected from Geofabrik. The Mapillary street view images were downloaded via querying Mapillary APIs including the metadata of each image, from 2014 to 2018. To facilitate further study, we only extracted images located in the intersection buffer. Mapillary Vistas was from Neuhold et al. (2017), which contains 25,000 high-resolution images annotated into 66 object categories. They are used as the training set for the semantic segmentation model—PSPNet. Last but not the least, GTSDB and GTSRB were from Stallkamp et al. (2011, 2012), and are applied for training object detection model—YOLOv3 and proposed object classification model—ShallowNet, respectively.

In summary, above all are reasons why we choose Berlin as our study area. Fig. 10 depicts the example area of Berlin as well as the distribution of Mapillary street view camera locations and OSM building footprints.

4.2 Extraction of road-related objects with PSPNet

4.2.1 Training

For the segmentation task, our implementation is based on the public framework TensorFlow. Like the Zhao et al. (2017), we also use the "poly" learning rate policy (the learning rate is multiplied by $(1-\frac{iter}{max\_iter})^{power}$). We set the base learning rate to 0.01 and power to 0.9. The training is performed on three NVIDIA GTX 1080Ti GPUs using stochastic gradient descent (SGD) with momentum m = 0.99 and weight decay = 0.0001. Due to limited physical memory on GPU cards, we set the "batchsize" to 4 for each GPU card during training. In addition, we crop the Mapillary training images to a size of 720 × 720, and start with a pre-trained ResNet34 (He et al., 2016) model with the dilated network strategy (Yu & Koltun, 2015) to extract the feature map. For data augmentation, we adopt random mirror, rotations [− 5°, 5°], random resize between 0.5 and 2, and small enhancements in the image's color, sharpness, and brightness for Mapillary Vistas. This comprehensive data augmentation scheme makes the network resist overfitting.

4.2.2 Evaluation and comparison

The performance on Mapillary street-level images was evaluated with PSPNet. Figure 11 shows several

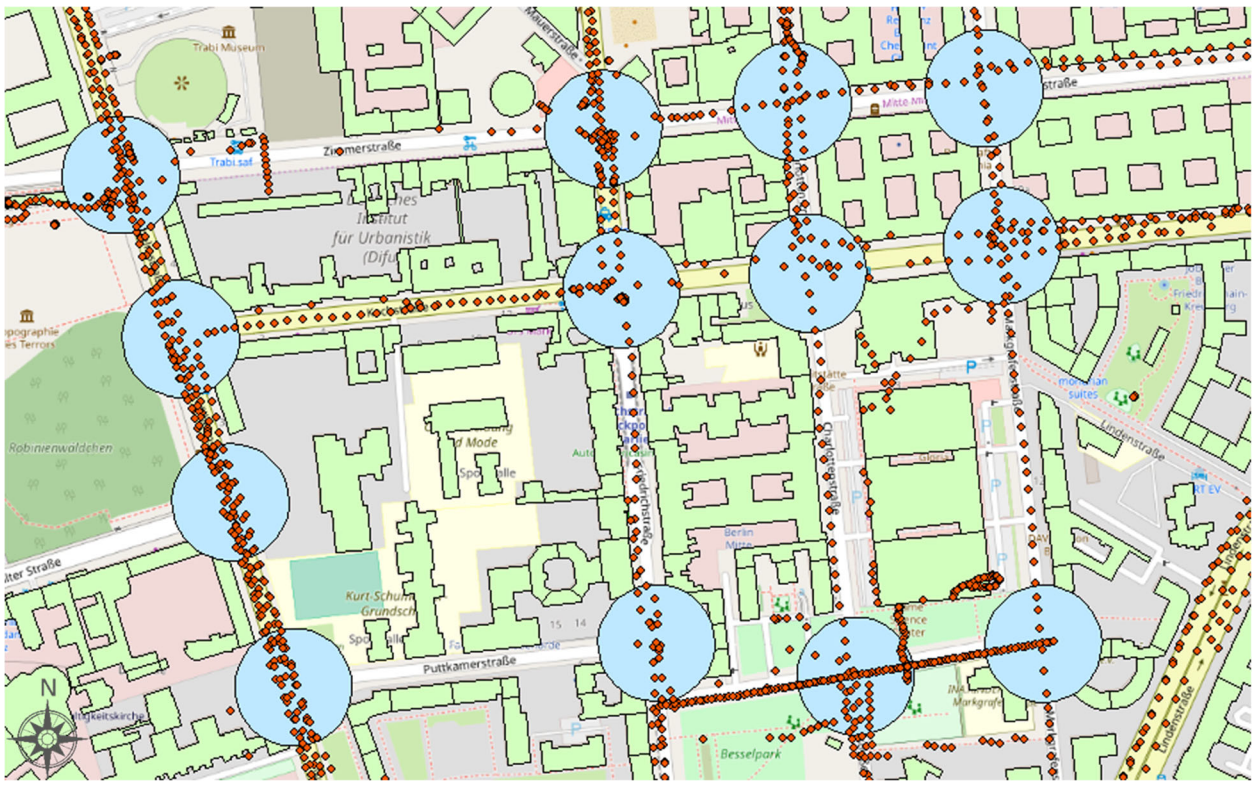


Fig. 10 Example area of Berlin (map:© OpenStreetMap contributors). Red dots, green polygons, and blue circles are the Mapillary street view camera locations, OSM building footprints, and intersection buffers, respectively

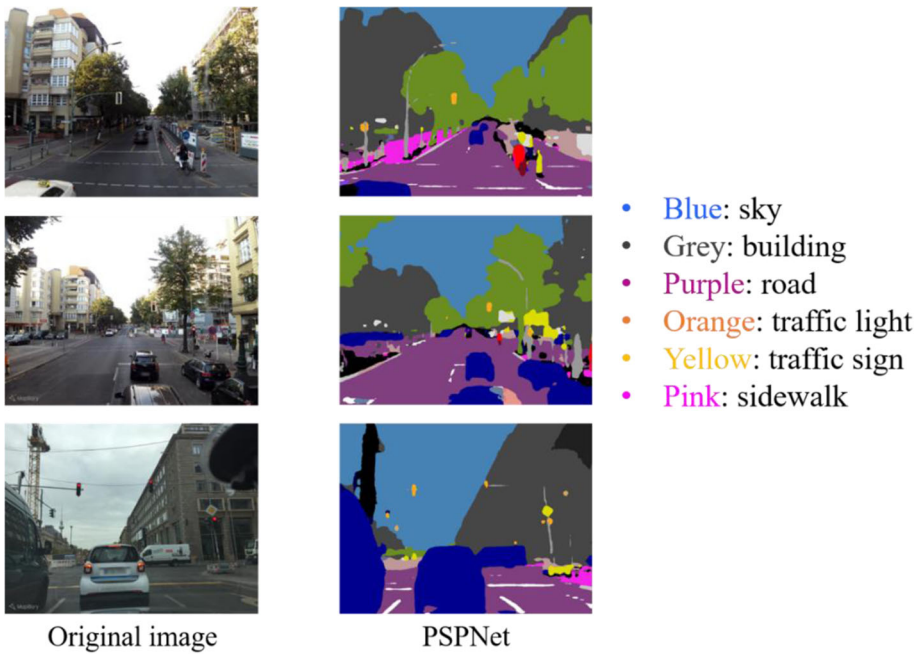


Fig. 11 Examples of PSPNet results. The first column lists the original images. The second column represents the segmented results

segmented examples, where the first column represents the sample images located at the intersections, the second column corresponds to the segmented results. From the segmentation results, the PSPNet model we trained can well segment the sky, buildings, roads, traffic signs and other objects that we want. Furthermore, to prove the superiority of our PSPNet model, a comparison test is conducted with the state-of-the-art model, DeepLabv3+ (Chen et al., 2018). In Table 2, our trained PSPNet model achieves Mean IoU 34.17% and Pixel Acc. 91.3%, and both of them outperform the DeepLabv3 + .

4.3 Detection and classification of traffic signs

In this subsection, to prove the superiority of our used or proposed network, we will conduct a series of comparative experiments on detection network YOLOv3 and the classification network ShallowNet.

4.3.1 Traffic signs detection

4.3.1.1 Training Due to some differences between the street view at intersections and the ordinary street view, we add 300 extra annotated Mapillary images at road intersections into the GTSDB dataset to form a hybrid dataset. The dataset is divided into 750/450 images for training and testing. We train the YOLOv3 with Darknet on an NVIDIA GTX 1080Ti GPU card, and set "batchsize" to 8. The warm-up strategy is adopted in the training phase, i.e. starting with a very small learning rate at the beginning of training. As the number of iterations increases, the initial learning rate gradually increases to 0.001. Starting from the second epoch, the normal gradient descent is made with 0.001 as the initial learning rate. Meanwhile, to augment the data, we use rotations [− 5°, 5°], random flipping, random scale [20, 200], and color space conversion.

Table 2 Comparison of mIoU and pixel accuracy between our trained PSPNet and DeepLabv3 +

Method	Mean IoU(%)	Pixel Acc.(%)
PSPNet	**34.17**	**91.3**
DeepLabv3+	33.97	90.2

4.3.1.2 Evaluation and comparison To prove that our trained YOLOv3 model is excellent at both processing speed and detection accuracy, we compare YOLOv3 with the previous best-performing method (Faster R-CNN (Ren et al., 2015)) on the testing set. In Table 3 our trained YOLOv3 model yields mAP (mean Average Precision) 94.7% and sec/img (second per image) 0.025 s, and both of them outperform the Faster R-CNN. The detection speed of approximately 30FPS is much faster than two-stage detector like Faster R-CNN. In addition, the performance of traffic sign detection on Mapillary street-level images is evaluated with YOLOv3. Figure 12 shows several example results.

Table 3 Comparison of mAP and detection time between our trained YOLOv3 and Faster R-CNN on the GTSDB + Mapillary images hybrid testing set

Method	Input size	mAP(%)	Model size(M)	Sec/img(s)
YOLOv3(Darknet-53)	**608 × 608**	**94.7**	**246.4**	**0.025**
Faster R-CNN (ResNet)	1280 × 720	90.5	267	0.230

4.3.2 Traffic signs classification

4.3.2.1 Training The public GTSRB dataset contains only 43 types of traffic signs, but it does not cover signs that often appear at intersections. Hence, we add two more categories to reach 45 categories in total. The dataset is divided into 75 K/12 K images for training and testing. Due to the uneven number of different categories of traffic signs, we also use a data augmentation technique during training, which includes histogram equalization of color images, affine transformation, contrast enhancement, Gaussian blur, Gaussian random noise, color space conversion, and random inactivation of pixel values. The training is performed on an NVIDIA GTX 1080Ti GPU using Adam Optimizer with Cross Entropy Loss Function.

4.3.2.2 Ablation study for ShallowNet To evaluate ShallowNet, we conduct experiments with several settings, including batch normalization (BN), dropout, and data augmentation. As listed in Table 4, the accuracy of manual recognition is 98.84%. Although the accuracy of manual recognition is very high, the automation degree is low, which is not conducive to information extraction. For the simplest ShallowNet (only convolution, pooling and full connection operation), the test accuracy on GTSRB is 95.89%. While it does not work better than manual recognition, it has higher automation degree and faster forward propagation speed (it only takes 3.6 ms on average to detect an image in CPU mode).

Even though the ShallowNet structure is very simple, the number of neurons in the fully-connected layer is large, which may lead to overfitting to some extent. Hence, we introduce dropout at the first fully-connected layer and successfully increase accuracy by nearly 1.6%. Besides, batch normalization is adopted in ShallowNet_Drop to reduce the difference in the distribution of original data, and to help speed up the convergence of training. ShallowNet_BN_Drop has a similar performance to manual recognition. Finally, we explore whether data augmentation improves the accuracy of the model or not, and augment the data on ShallowNet_BN_Drop. It achieves the accuracy of 99.52% on the testing set, which surpasses the accuracy of manual recognition, and increases by over 1% compared to ShallowNet_BN_Drop. Through this experiment, it can be proved that data augmentation is very critical to improve the accuracy of the model. Figure 13 shows several examples.

4.4 Localization of traffic lights and signs

In this subsection, we apply the method introduced in Section 3.3 for locating the traffic signs and lights based on ATBT and urban rules. The experimented image sequences are merged from multiple image sequences according to their geolocations and meanwhile, misaligned images are corrected using Structure from Motion (SfM). Each image sequence refers to a trajectory of a volunteer user traveling along the road; and, over time, the same road segment may be covered by multiple sequences that are uploaded by different volunteers. One

Fig. 12 Examples of traffic sign detection results based on YOLOv3

Table 4 Investigation of ShallowNet with different settings. 'Drop', 'BN' and 'Aug' represent dropout, batch normalization and data augmentation, respectively

Method	Accuracy(%)	Sec/img (ms)
Human performance	98.84	/
ShallowNet	95.89	3.6
ShallowNet_Drop	97.47	/
ShallowNet_BN_Drop	98.49	/
ShallowNet_BN_Drop_Aug	**99.52**	**3.6**

hundred intersections with four or three branches are tested in the experiment, with over 350 image sequences and more than 3400 images.

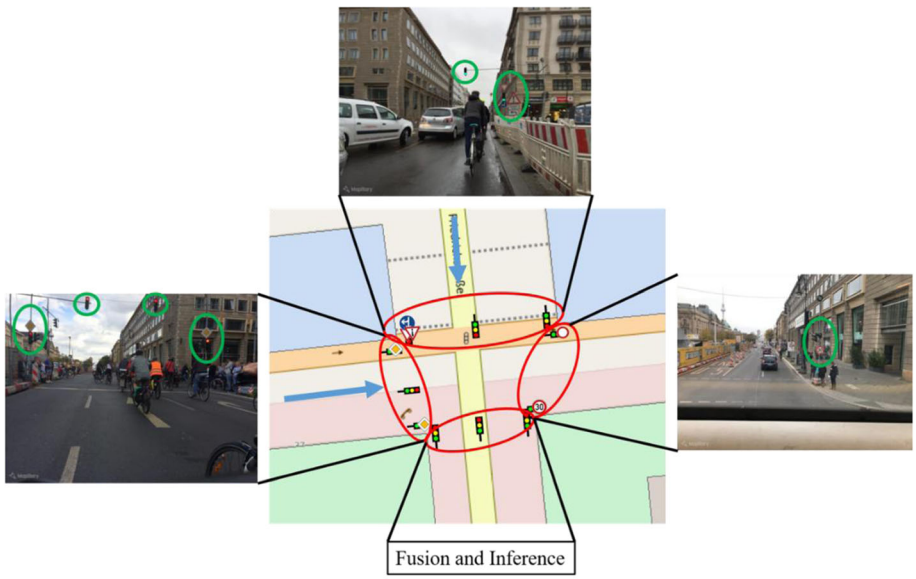
Using hybrid results both from semantic segmentation and traffic signs detection & classification, a parsed scene with detailed semantic and attributed information can be established. For this purpose, the hierarchy of semantic objects needs to be applied, as there are coherent relations of topologies, attributes and semantics of the road objects. Therefore, an ATBT can be created based on urban rules for each image in the image track to depict the topologies among road objects. Then, we integrate the final updated ATBTs, rather than only using the result of one ATBT. Because some important items (such as traffic signs) in a certain image may be occluded by cars but the next image does not, which can play a role of verification and supplement. Ultimately, it can produce the final localization results along the driving direction (or camera shooting direction). Please note that this is not a precise localization, but in fact, it can indicate the estimated location of the traffic lights and signs. In Fig. 14, the qualitative localization results of one crossroad and one T-junction examples are displayed.

In general, for the localization task, two spatial data quality elements should be assessed: completeness and positional accuracy. While positional accuracy is the best-established indicator of accuracy in mapping science (Mobasheri et al., 2018), official position data (ground truth) of traffic signs and lights are not available. We cannot compare our generated positions with ground truth data on positional accuracy. However, we still manually collect the locations of traffic signs/lights from Google satellite map through visual observation and make these locations as "reference data" to access the completeness and positional accuracy of our localization results. In terms of completeness level, we get over 97% in all 100 testing intersections. Please note that only when all traffic lights and signs are detected and their predicted locations are not far away from the real locations at an intersection, then we would consider it as a complete and correct case. Figure 15 shows two examples corresponding to Fig. 14a-b respectively, where red dot 1 in the right figure of (a) contains three signs, and each of the red dots 1,2,3 in the right figure of (b) contains two signs because they are overlapped. As can be seen in the figures, both examples have obtained approximate positional accuracy compared to the annotated "reference data".

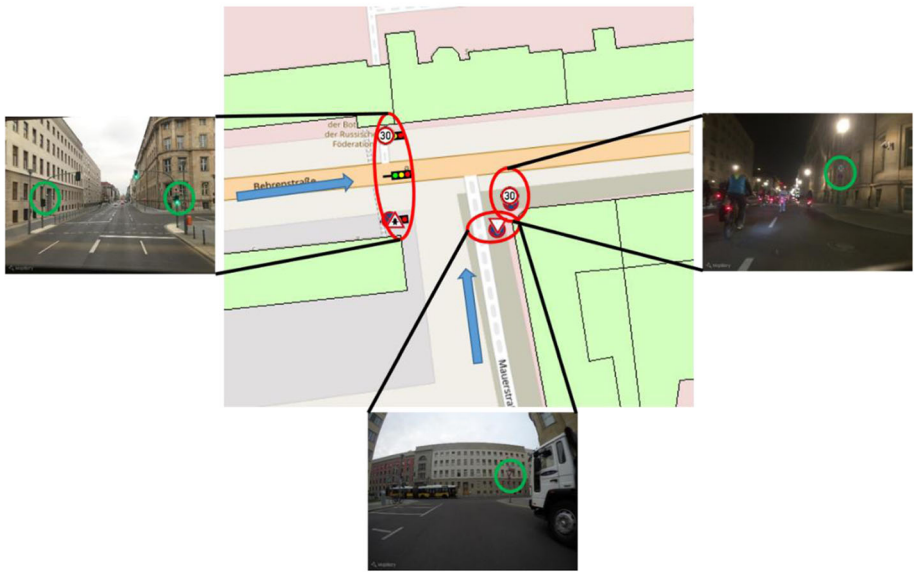
5 Conclusions and future work

In this paper, we propose an automatic approach to detect and place traffic lights/signs at road intersections in relatively high completeness and positional accuracy. The proposed method relies on two deep learning pipelines (one for image semantic segmentation and the other for traffic sign detection & classification), as well as novel ATBTs based on six urban rules for traffic lights/signs localization. The method has been tested at multiple intersections using Mapillary street view images

Fig. 13 Examples of traffic sign classification results based on ShallowNet



(a)



(b)

Fig. 14 Examples of qualitative localization results of traffic lights and signs at two types of intersections. **a** localization results at the crossroad; **b** localization results at the T-junction

in Berlin, Germany. We validate the effectiveness of the proposed approach on two object classes: traffic signs and traffic lights, and introduce two spatial data quality elements: completeness and positional accuracy. Experimental results demonstrate that our approach obtains great objects completeness level (over 97% among 100 testing intersections) and relatively high positional accuracy compared to the manually collected "reference data". Therefore, the proposed method provides a promising solution for enriching and updating OSM intersection data.

So far, to the best knowledge of the authors, there have not been digital maps with so detailed information. However, this kind of information is of vital importance

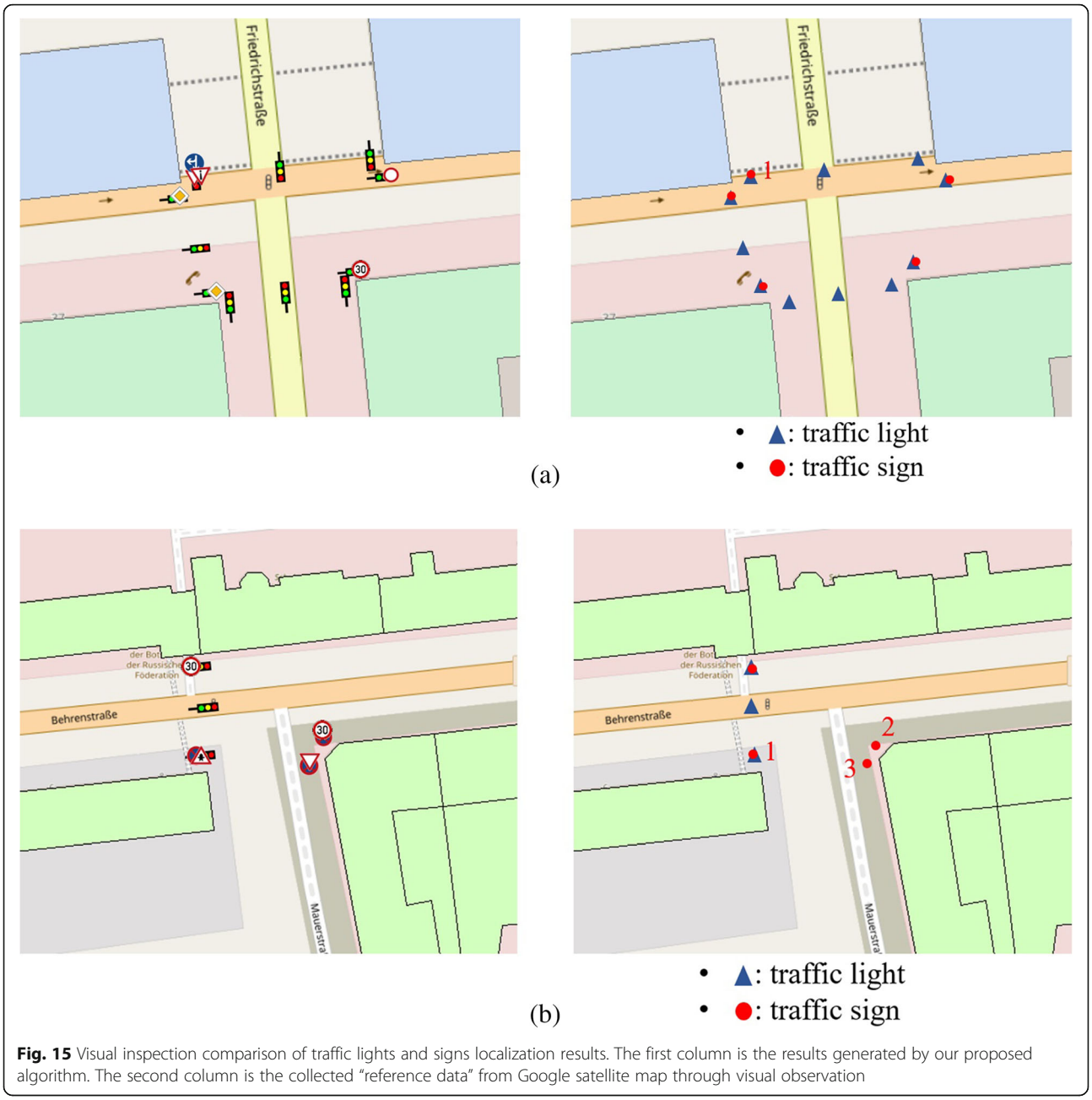


Fig. 15 Visual inspection comparison of traffic lights and signs localization results. The first column is the results generated by our proposed algorithm. The second column is the collected "reference data" from Google satellite map through visual observation

for many applications. For instance, together with trajectory data, information of traffic signs at road intersections may help offer more reasonable explanations for many spatial analyses related to urban structure and urban transportation. In this context, it is very useful for urban planning recommendations.

At present, the proposed method can only be applied to intersections with four or three branches and it is difficult to handle with complex intersections, such as roundabouts or five-branch intersections. In addition, the premise of employing this method is that there needs to have at least one traffic light marked in OSM data. Otherwise, we cannot identify and select the intersections by using DBSCAN. That is the second limitation of the proposed approach. However, Europe's OSM data is the richest compared to other continents, so the proposed method can be applied at least in Europe. In the future, we will further optimize the proposed approach and aim to resolve and overcome the above limitations. On the other hand, the GTSRB dataset used in this work only includes 45 categories, which does not cover all types of traffic signs in Germany or other

countries. Hence, another area for future research will be the extension of GTSRB dataset to increase the generalization of ShallowNet. Ultimately, we want to create and contribute a separate intersection layer to OSM, where contains the number of lanes, the width of roads and other road-related objects, and to provide some help for autonomous driving or navigation.

5.0.0.1 Code availability The code is currently stored on a local area network (LAN) of university and have not submitted to like Github. If this paper was accepted, we will release the code there immediately.

Authors' contributions

Hongchao Fan formed the research idea and revised the manuscript. Chaoquan Zhang designed and conducted the experiment, and drafted the manuscript. Wanzhi Li co-conducted the experiment and analyzed the results. All authors read and approved the final manuscript.

Funding

This research was supported by research fund from the Norwegian Public Roads Administration under project name Road.

Availability of data and materials

- https://www.mapillary.com/data
- http://download.geofabrik.de/

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

All authors approve that the Journal of Computational Urban Science can publish our article.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Civil and Environmental Engineering, Norwegian University of Science and Technology, Trondheim, Norway. [2]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China.

Received: 19 April 2021 Accepted: 18 July 2021
Published online: 04 August 2021

References

Arietta, S. M., Efros, A. A., Ramamoorthi, R., & Agrawala, M. (2014). City forensics: Using visual elements to predict non-visual city attributes. *IEEE Transactions on Visualization and Computer Graphics, 20*(12), 2624–2633. https://doi.org/10.1109/TVCG.2014.2346446.

Chang, V., Walters, R. J., & Wills, G. B. (2016). Organisational sustainability modelling—An emerging service and analytics model for evaluating cloud computing adoption with two case studies. *International Journal of Information Management, 36*(1), 167–179. https://doi.org/10.1016/j.ijinfomgt.2015.09.001.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 801–818).

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., et al. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213–3223).

Crandall, D. J., Backstrom, L., Huttenlocher, D., & Kleinberg, J. (2009). *Mapping the world's photos*, *Proceedings of the 18th international conference on world wide web* (pp. 761–770).

Dubey, A., Naik, N., Parikh, D., Raskar, R., & Hidalgo, C. A., 2016. Deep learning the city: Quantifying urban perception at a global scale. In Computer Vision – *ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 196–212.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, no. 34, pp. 226–231).

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision, 88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4.

Gao, S., Li, L., Li, W., Janowicz, K., & Zhang, Y. (2017). Constructing gazetteers from volunteered big geo-data based on Hadoop. *Computers, Environment and Urban Systems, 61*, 172–186. https://doi.org/10.1016/j.compenvurbsys.2014.02.004.

Gebru, T., Krause, J., Wang, Y., Chen, D., Deng, J., Aiden, E. L., & Fei-Fei, L. (2017). *Using deep learning and google street view to estimate the demographic makeup of neighborhoods across the United States*, *Proceedings of the National Academy of Sciences* (pp. 13108–13113).

Goodchild, M. F. (2007). Citizens as voluntary sensors: Spatial data infrastructure in the world of web 2.0 (editorial). *International Journal of Spatial Data Infrastructures Research (IJSDIR), 2*, 24–32.

Hachmann, S., Arsanjani, J. J., & Vaz, E. (2018). Spatial data for slum upgrading: Volunteered geographic information and the role of citizen science. *Habitat International, 72*, 18–26. https://doi.org/10.1016/j.habitatint.2017.04.011.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hebbalaguppe, R.; Garg, G.; Hassan, E.; Ghosh, H.; Verma, A., 2017. Telecom Inventory management via object recognition and localisation on Google Street View Images. *In Proceedings of the 2017 IEEE winter conference on applications of computer vision (WACV)*, Santa Rosa; pp. 725–733. IEEE.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Ibanez-Guzman, J., et al. (2010). Vehicle to vehicle communications applied to road intersection safety, field results. In *International IEEE conference on intelligent transportation systems* (pp. 192–197). Funchal Portugal: IEEE.

Ioffe, S., & Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In International conference of machine learning* (pp. 448-456). PMLR.

Jensen, M. B., Philipsen, M. P., Møgelmose, A., Moeslund, T. B., & Trivedi, M. M. (2016). Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems, 17*(7), 1800–1815. https://doi.org/10.1109/TITS.2015.2509509.

Krylov, V., Kenny, E., & Dahyot, R. (2018). Automatic discovery and geotagging of objects from street view imagery. *Remote Sensing, 10*(5), 661. https://doi.org/10.3390/rs10050661.

Kuntzsch, C., Sester, M., & Brenner, C. (2016). Generative models for road network reconstruction. *International Journal of Geographical Information Science, 30*(5), 1012–1039. https://doi.org/10.1080/13658816.2015.1092151.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539.

Li, X., Ratti, C., & Seiferling, I. (2018). Quantifying the shade provision of street trees in urban landscape: A case study in Boston, USA, using google street view. *Landscape and Urban Planning, 169*, 81–91. https://doi.org/10.1016/j.landurbplan.2017.08.011.

Li, X., Zhang, C., & Li, W., 2017. Building block level urban land-use information retrieval based on Google street view images. *GIScience & Remote Sensing*, 54(6), 819–835.

Li, X., Zhang, C., Li, W., Ricard, R., Meng, Q., & Zhang, W. (2015). Assessing street-level urban greenery using Google street view and a modified green view index. *Urban Forestry & Urban Greening, 14*(3), 675–685. https://doi.org/10.1016/j.ufug.2015.06.006.

Mattyus, G.; Wang, S.; Fidler, S.; Urtasun, R., 2016. HD maps: Fine-grained road segmentation by parsing ground and aerial images. *In Proceedings of the 2016 IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas; pp. 3611–3619. IEEE.

Melnikov, V. R., Krzhizhanovskaya, V. V., Lees, M. H., & Boukhanovsky, A. V. (2016). Data-driven travel demand modelling and agent-based traffic simulation in Amsterdam urban area. *Procedia Computer Science, 80*, 2030–2041. https://doi.org/10.1016/j.procs.2016.05.523.

Mirowski, P., Grimes, M. K., Malinowski, M., Hermann, K. M., Anderson, K., Teplyashin, D., ... Hadsell, R., 2018. Learning to navigate in cities without a map. *arXiv preprint arXiv:1804.00168*.

Mobasheri, A., Huang, H., Degrossi, L., & Zipf, A. (2018). Enrichment of openstreetmap data completeness with sidewalk geometries using data mining techniques. *Sensors, 18*(2), 509.

Naik, N., Kominers, S. D., Raskar, R., Glaeser, E. L., & Hidalgo, C. A. (2017). Computer vision uncovers predictors of physical urban change. *Proceedings of the National Academy of Sciences, 114*(29), 7571–7576. https://doi.org/10.1073/pnas.1619003114.

Naik, N., Philipoom, J., Raskar, R., & Hidalgo, C. (2014). Streetscore-predicting the perceived safety of one million streetscapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (pp. 779–785).

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807–814).

Neis, P., Zielstra, D., & Zipf, A. (2012). The street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011. *Future Internet, 4*(1), 1–21.

Neuhold, G., Ollmann, T., Rota Bulo, S., & Kontschieder, P. (2017). The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision* (pp. 4990–4999).

Redmon, J., & Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91–99).

Seitz, S.M.; Curless, B.; Diebel, J.; Scharstein, D.; Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas; Volume 1, pp. 519–528. IEEE.

Snavely, N., Seitz, S. M., & Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision, 80*(2), 189–210. https://doi.org/10.1007/s11263-007-0107-3.

Soheilian, B., Paparoditis, N., & Vallet, B. (2013). Detection and 3D reconstruction of traffic signs from multiple view color images. *ISPRS Journal of Photogrammetry and Remote Sensing, 77*, 1–20. https://doi.org/10.1016/j.isprsjprs.2012.11.009.

Jan Erik Solem., 2017. Mapillary: Celebrating 200 million images. Available online: https://blog.mapillary.com/update/2017/10/05/200-million-images.html. Accessed May 2020

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C., 2011. The German traffic sign recognition benchmark: A multi-class classification competition. *In The 2011 international joint conference on neural networks* (pp. 1453-1460). IEEE.

Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012). Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks, 32*, 323–332. https://doi.org/10.1016/j.neunet.2012.02.016.

Timofte, R.; Van Gool, L., 2011. Multi-view manhole detection, recognition, and 3D localisation. *In Proceedings of the 2011 IEEE international conference on computer vision workshops (ICCV workshops)*, Barcelona; pp. 188–195. IEEE.

Trehard, G.; Pollard, E.; Bradai, B.; Nashashibi, F., 2014. Tracking both pose and status of a traffic light via an interacting multiple model filter. *In proceedings of the international conference on information FUSION (FUSION)*, Salamanca, pp. 1–7. IEEE.

Wegner, J.D.; Branson, S.; Hall, D.; Schindler, K.; Perona, P., 2016. Cataloging public objects using aerial and street-level images-urban trees. *In Proceedings of the 2016 IEEE conference on computer vision and pattern recognition*, Las Vegas; pp. 6014–6023. IEEE.

Workman, S.; Zhai, M.; Crandall, D.J.; Jacobs, N., 2017. A unified model for near and remote sensing. *In Proceedings of the 2017 IEEE conference on computer vision and pattern recognition (CVPR)*, Honolulu; Volume 7. IEEE.

Yu, F., & Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*.

Zhang, F., Hu, M., Che, W., Lin, H., & Fang, C. (2018). Framework for virtual cognitive experiment in virtual geographic environments. *ISPRS International Journal of GeoInformation, 7*(1), 36. https://doi.org/10.3390/ijgi7010036.

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J., 2017. Pyramid scene parsing network. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. (pp. 2881-2890). IEEE.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralba, A., 2017. Scene parsing through ADE20K dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 633-641). IEEE.

Publisher's Note