Applied Intelligence (2022) 52:8226–8242
https://doi.org/10.1007/s10489-021-02838-w

Detection-by-tracking of traffic signs in videos

Yanting Zhang[1] · Zijian Wang[1] · Ruoning Song[2] · Cairong Yan[1] · Yonggang Qi[2]

Accepted: 10 September 2021 / Published online: 23 October 2021

Abstract
Continuously detecting traffic signs in a video sequence is necessary for autonomous or assisted driving scenarios, since a vehicle needs the information from the signs to facilitate navigation. Single-image based traffic sign detector may fail in many cases, when the car moves fast on the road, resulting in motion blur, partial occlusion, and abrupt environmental change. In this paper, we propose an effective methodology, called detection-by-tracking, for robust traffic sign detection in videos, so as to improve the detection performance beyond a basic object detector. We explore the temporal cues among frames to help with the proposal reasoning for further regression. The correlations of spatial location and appearance similarity for the same sign in adjacent frames are considered in our approach. Experimental results show that the proposed detection-by-tracking mechanism is helpful, with improved detection performance to a large extent. Moreover, the idea of the detection-by-tracking can also be generalized to other scenarios for object detection tasks in videos.

Keywords Traffic sign · Video object detection · Tracking · Shortest path

1 Introduction

Traffic signs are frequently observed on the road, providing abundant information like driving direction, speed limitation, danger warning, and so on. They are usually designed to be unique and distinguishable with simple shapes and uniform colors, standing out against the environment. Traffic sign detection is essential in real-world applications such as autonomous driving, traffic surveillance, driver assistance, and road network maintenance [3, 54]. Moreover, the instructions implied by the signs can help to secure driving safety, and support other applications in autonomous driving with their constant geo-location and planar property. For example, the prominent features located on the signs can be utilized to create reliable correspondence among frames for better constraints in simultaneous localization and mapping (SLAM) [41, 60]. Besides the ordinary traffic signs, some works [37, 39, 50] also explore to use planar makers in indoor environments where there are not enough structure features for localization purposes either in robot uses or micro air vehicles (MAVs). Yu et al. [56] exploit the Quick Response (QR) code landmarks in positioning and navigation system for library robots. In real-world road scenarios, traffic signs are and will still be important components on the roads to enhance driving safety. Intensive researches on traffic sign detection have been conducted by both academic and industrial communities all over the world, however, it is still a challenging task due to real-world complicated driving scenarios, such as motion blur, camera defocus, pose variation, occlusion, and changing lighting condition. It's meaningful to explore a robust scheme for traffic sign detection continuously in videos.

When it comes to detection, people pay attention to both the location and class information of objects in the image. It's necessary for a driving vehicle to perceive the

✉ Yanting Zhang
ytzhang@dhu.edu.cn

Zijian Wang
wang.zijian@dhu.edu.cn

Ruoning Song
songrn@bupt.edu.cn

Cairong Yan
cryan@dhu.edu.cn

Yonggang Qi
qiyg@bupt.edu.cn

[1] School of Computer Science and Technology, Dong Hua University, Shanghai, China

[2] School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China

environment and be aware of the sign's location and its type. People have explored and developed various kinds of methods for traffic sign detection. As is known to us all, traffic signs are usually designed as regular shapes with discriminating colors. There are three main traffic sign classes around the world [25, 63], i.e., prohibition signs (red circular), mandatory signs (circular blue), and danger signs (red triangle). Some random examples of traffic signs for each big category mentioned above are shown in Fig. 1.

Previously, traditional methods like color-based as well as shape-based methods are dominant approaches for traffic sign detection [45]. Many researchers try to analyze the color composition so as to generate good representations for signs [7, 17]. The edges and gradients on the sign plates also contribute to traffic sign localization and recognition [2, 5]. Either color-based or texture-based method has certain limitations, thus some authors make use of both to better detect the traffic signs. For example, the color cues are applied to localize the region of interest and the detection is completed with shape methods taking advantage of geometric information [46]. Giving the extracted hand-crafted features, a classifier is usually applied to determine the category, e.g., support vector machine (SVM) [14]. Color histogram and histogram of gradient (HoG) are commonly favorable choices for feature engineering. However, the representation ability of hand-crafted features is limited and they are easily affected by environment changes.

To improve the detection performance in an image, using more powerful representation features is a critical solution. In recent years, great progress has been witnessed owing to the adoption of convolutional neural networks (CNNs). The features can be learned through the deep learning frameworks and be further utilized in image classification. Great progress has been made in the field of object detection owing to the resurgence of deep convolutional neural networks compared to traditional approaches. Some classical work like R-CNN [43] and YOLO [42] families have become the basis for various applications with a significant improvement in detection accuracy and efficiency [59]. These detectors perform on single images and they are universal for different kinds of targets. Considering the fact that a car drives on the road, at the same time, the dash camera mounted on the car can record the environment into videos. Videos have increased one temporal dimension than pure images. Simply applying object detection methods like Faster R-CNN on images, there inevitably will be failure cases due to motion blur, partial occlusion, scale change, and bad viewing perspective. We show some results of using Faster R-CNN detector in sampled video frames in Fig. 2, from which we can see that missed detections (i.e., no bounding boxes are identified on the objects) or incorrect detections (e.g., incorrect classifications, false-positive detections, incorrect locations and/or sizes of bounding boxes) exist. Temporal correspondence among image frames within a video sequence deserves to be investigated to solve these problems.

Challenges exist to incorporate the temporal information effectively. The applications benefiting from video analyses to enhance traffic safety have become increasingly popular. Based on videos collected from surveillance systems, Chen et al. [10] propose a novel sparse representation model for understanding pedestrian abnormal trajectories to

Fig. 1 Representative examples of traffic signs belonging to three big categories: (**a**) prohibition signs. (**b**) mandatory signs. (**c**) danger signs. Note that, there are also other special signs which don't belong to any of the big categories shown above

Fig. 2 Both good detections and failure cases for traffic sign detection in different video sequences using a single-image based detector. Red boxes and green boxes denote incorrect and correct classifications, separately

improve public safety. VID dataset released by ImageNet [44] collects a large number of videos for common objects (not specifically traffic signs), which allows researchers to experiment with different approaches. Authors either try to increase per-frame feature quality for better detection performance [52, 61], or explore objects' location information among frames for association in the box level [22, 27–29]. Generally speaking, video object detection is kind of costly, since an end-to-end deep learning-based approach normally requires a mass of training data with considerable annotations. It's usually hard to collect and annotate such a large-scale dataset. Traffic signs are important components in driving scenarios. Most available traffic sign datasets are composed of still images. For example, UAH Dataset [35], CVL Dataset [30], German Traffic Sign Detection Benchmark(GTSDB) [25], Russian Traffic SignDataset(RTSD) [49], and Tsinghua-Tencent 100K(TT100K) [64]. Even though the MASTIF dataset [47] provides some annotated video sequences, only a few frames in a long sequence are labeled. Each sign is annotated 4-5 times at different distances from the car. The coarse-grained annotation does not meet the requirement for developing a useful algorithm in most cases. With autonomous driving gradually appearing in our daily life, it is critical for an autonomous driving system to reliably detect track traffic signs to ensure driving safety. The problem will be, if we don't have enough annotated video sequences, can we still design an approach that has better performance than the base object detectors? Or can we improve the video object detection performance given a trained base object detector?

In this paper, we try to improve the traffic sign detection performance beyond a basic object detector. We explore the temporal consistency among image frames to detect traffic signs continuously in videos. The contributions of this work can be summarized as three folds.

1) We propose a detection-by-tracking approach based on shortest path searches over candidate proposals.

This approach leverages the temporal correlation information of the same traffic sign among video frames. The regression for the promising proposal is greatly enhanced benefiting from neighboring frames' information.

2) By recovering the missing frames or correcting the mistaken frames, we can improve the detection performance beyond the single-image based models. The bottleneck of poor detection performance caused by limited training samples can thus be resolved to some extent since large-scale annotated video sequences are no longer needed.
3) We adapt several algorithms, which are firstly designed either for multiple object tracking (MOT) or VID, to detect the traffic signs in videos. The experimental results of these methods provide a reference regarding the detection performance. Post-processing, tracking-by-detection, and end-to-end approaches are all included.

The rest of the paper is organized as follows. In Section 2, we survey the related literature. Our proposed system together with several other approaches is introduced in Section 3. Experimental results and discussions are given in Sections 4 and 5, separately. Finally, the conclusion is given in Section 6.

2 Related works

We review the literature from three perspectives. First, color-based and shape-based methods are discussed as the traditional approach for traffic sign detection. Learning-based methods are then reviewed. At last, we address the topic of object detection in videos to gain some insights about how to incorporate temporal information within the detection framework.

2.1 Color-based and shape-based methods

Traditionally, researchers apply color-based and shape-based methods to find the region of interest (RoI) in images, since man-made objects usually have distinct characteristics. Color is an important attribute for road signs. Several colors like red, blue, and yellow are commonly used in traffic signs. The color-based methodologies try to tell apart the foreground from the background. The most intuitive color space is the RGB system, while the hue, saturation, intensity (HSI) system is less sensitive to lighting changes. Fleyeh [17] proposes to use improved HSI color space for color detection and segmentation of road signs, so as to alleviate the influences from lightning variation. Escalera et al. [15] use color thresholding to segment the image, followed by shape analysis to detect the signs. Benallal et al. [7] have studied how the color appearance of road signs change under different outdoor illuminations during a day. The learned rules further help with the segmentation, which constitutes the first step towards identifying and locating the road signs. Though the remarkable color properties enable the traffic signs to stand out in the wild environment, yet it is still an arduous process relying only on the color appearance due to the sensitivity to various factors such as the reflection of sign surface, lightning changes in a day, weather condition, etc.

In order to overcome the problems existing in color-based methods, researchers explore shape-based methods. Generally speaking, shape-based methods are expected to be more robust. They take advantage of the edges and connect them to regular polygons or circles [2, 5] through Hough-like voting scheme or template matching. For example, authors in [18] find the contours in images first and then leverage the Hough transform of the edges to carry out the detection. Hechri et al. [23] apply a template-matching scheme in the color-segmented image to filter out the regions without any traffic sign. Larsson et al. [30] use shape descriptors in the frequency domain by applying Fourier transform to the contours, which are further combined with star-shaped object models as prototypes for classification. Barnes et al. [1] present an approach based on the radial symmetry operator to recognize speed signs. Behloul et al. [4] propose to employ a minimum rectangle to encompass the detected contour on the filtered pattern maps for pattern recognition. The score defined over the intersection between the detected pattern and the rectangle is utilized to determine the shape of signs.

Detecting road signs based on shape information can be tough when the traffic sign is very small in the image, and it is easy to be mixed up with other man-made objects such as commercial signs. Though gradients are less sensitive to luminance and faded color, it consumes much more time in the gradient computation process. Combining both color and shape properties is a compromise [55]. The faster speed of color segmentation and the higher accuracy of HoG calculation are well balanced. The traditional methods have played a great role in traffic sign detection before the adoption of deep learning based approaches.

2.2 Learning-based methods

Based on large amounts of annotated data, we could develop efficient algorithms taking advantage of machine learning. Previously, hand-crafted features are widely used. Viola and Jones [51] design Haar wavelet features and combine a set of classifiers in a cascade based on AdaBoost. The

system has achieved good detection rates in the domain of face detection. Karla et al. [9] extend the Viola–Jones approach to detect triangular traffic signs. Dalal and Triggs [12] adopt a linear SVM based on grids of HoG descriptors for human detection. Zaklouta et al. [57] also use HoG in detecting triangular warning signs and an approximate nearest neighbors search using a KD-tree to refine the result. In addition to above-mentioned approaches, various hand-crafted features such as LBP, SIFT, SURF, BRISK are explored in a learning-based manner for traffic sign detection in the field. However, the traditional hand-crafted features have a limited representation ability.

CNN features are more effective in representation as demonstrated by abundant work and they bring about extraordinary improvement compared to hand-crafted features. Sermanet and LeCun [48] use convolutional networks to learn invariant features of traffic signs in a supervised way. The results reach a high classification accuracy above human performance. In recent years, object detection [32, 33, 42, 43] in images has been thoroughly studied benefiting from the rapid development both in theories and platforms. Widely used object detection methods can be divided into single-stage and multi-stage detectors. YOLO [42] and SSD [33] are representatives of single-stage object detectors, which detect objects directly over a dense sampling of possible locations. On the other hand, the R-CNN approaches [20, 21, 43] and R-FCN [32], known as region-based methods, divide the object detection process into two stages: 1) region proposals are first generated through selective search or a regional proposal network (RPN); 2) the multi-task classification and bounding box regression are then carried out on the region candidates. In general, region-based methods, which pass the proposals with a higher objectiveness to the subsequent classification and bounding box regression tasks, can produce higher quality proposals compared to the single-stage methods.

In this paper, we also adopt a region-based approach, the Faster R-CNN pipeline [43], as the baseline single-image object detector. Differently, we not only apply Faster R-CNN to detect the traffic signs in the video frames, but also decompose its structure, and incorporate temporal information beyond tracking cues for the proposal selection, so as to carry out an explicit regression for the traffic sign localization in images. Note that, each individual traffic sign type can be quite similar within the big traffic sign category, e.g., the exemplar traffic signs shown in Fig. 1. Sometimes, the Softmax suppresses the true class label if a similar sign pattern exhibits a higher probability value, the classification can be misled in this way. Therefore, considering the video sequence' detection results can help guide the detection in the current frame. On the other hand, the spatial locations of the same traffic sign in a "track" is highly correlated. Thus, we predict the promising region where the sign is likely to appear for a straightforward proposal selection. Our proposed scheme can be regarded as an adaptation of the basic Faster R-CNN for video object detection.

2.3 Video-based tasks

With the explosion of large-scale video data, video object detection has significantly raised public attention recently in the community. ImageNet VID competition [44] has inspired a lot of works. Though many external factors such as motion blur, video defocus, partial occlusion and rare or bad pose can present a great challenge in the video object detection, nevertheless, the temporal consistency among video frames can still be leveraged to improve the detection performance [27, 29, 40, 52, 61]. The same object in adjacent frames possesses similar embedded features with correlated locations and sizes of bounding boxes, which can be a good reference for object detection in videos. Zhu et al. [61, 62] propose a flow-guided feature aggregation framework, which enhances the feature quality through aggregating them along motion paths on the feature level. Besides pixel-level feature calibration based on flow estimation, Wang et al. [52] also use instance-level calibration for better feature representation. Although we don't apply feature aggregation, we also leverage the feature similarity when carrying out the proposal selection in our proposed method. Kang et al. [27] propose a tubelet proposal network (TPN) to generate tubelet proposals, spatially aligned bounding boxes across time, in consecutive frames as the first stage of object detection in videos. Feichtenhofer et al. [16] use a multi-task objective to jointly tune a frame-based object detection and across-frame track regression network to simultaneously improve the detection and tracking performance. However, these methods all require extensive training based on a large-scale video dataset, which is usually hard to obtain.

Besides the video object detection task, multiple object tracking (MOT) is also related to video analysis. Traditional tracking methods, such as mean-shift [31], KCF [24], CSR-DCF [34], etc., can also localize the object according to appearance similarity among frames, but they are likely to face a severe drift problem along time especially the traffic signs move relatively fast with large size variations when observed from a fast-moving camera mounted on the car. Bergmann et al. [8] propose to use a detector to do the tracking, called "Tractor". Han et al. [22] propose "Seq-NMS" and Belkin et al. [6] propose "BBox-NMS". They both use good detections in nearby frames to boost detections with lower scores. These ideas are also well investigated in our work as baseline models. Similar to "Tractor", we also pay attention to region proposals for regression. Inspired by "Seq-NMS" and "BBox-NMS", we also boost the weaker detections with frames with higher

confidence scores. In all, we pay attention to both RoI feature similarity and location consistency, along with the class confidence scores of the proposals. Our method is supposed to result in a more robust and accurate detection result, which will be demonstrated in the following sections.

3 Methods

Given a video sequence, our goal is to detect the traffic signs in each frame, obtaining both location and category information in the images. Correlating information within adjacent frames is a general idea for video object detection tasks. We propose a scheme for traffic sign detection in videos called Shortest Path-based method [58]. By taking into account temporal information, this task can also be related to multiple object tracking since the temporal cues exist among frames and the tracked objects are what we aim to detect in each frame. We investigate several existing approaches, which are simple but effective. For handling multiple object tracking, Tractor [8] is one of the state-of-the-art techniques by combining spatio-temporal information with Faster R-CNN detection. It exploits the ability of a detector to regress and classify bounding boxes, without training or optimization during the tracking process. To compensate and boost the weaker detections benefiting from detections with higher scores in neighboring frames, Seq-NMS [22] is supposed to work during the post-processing phase beyond the Faster R-CNN detections. It uses the IoU information to do the sequence selection, followed by re-scoring. We borrow these two ideas and adapt the methods in our scenario. We refer above two methods as tractor-based approach and IoU-based approach, which serve as baselines in this paper. Moreover, we also carry out experiments on three video object detection approaches for comparison. The common characteristic of these methods is that they all leverage the Faster R-CNN framework, which also ensures a fair comparison. In this section, we will first give the problem statement, followed by Faster R-CNN architecture and our proposed approach for traffic sign detection in videos. Then, we will address in details the two adapted baseline algorithms. Finally, we also give a brief introduction to three other video object detection methods.

Problem statements The camera mounted on top of a car records videos when the car drives along the road. We assume there are T image frames in a video sequence, denoted as, $\{I_1, I_2, ..., I_T\}$. The traffic signs can be categorized into K classes. The goal of our task is to detect the traffic signs in every frame within the video sequence, including the position and category information of detected traffic signs. If N traffic signs are detected in the t-th frame, the result can be written as, $\{b_t^i, c_t^i | t \in T, c_t^i \in K, i \in N\}$, where $b_t^i = (x_t^i, y_t^i, w_t^i, h_t^i)$ implies the the bounding box position and size, and c_t^i denotes the class label of the i-th object.

3.1 Faster R-CNN

As a classical standard approach, Faster R-CNN applies a Region Proposal Network (RPN) to generate a multitude of proposals. Measures like Region of Interest (RoI) pooling or RoI align are applied to extract features based on the CNN maps. Then an object classification and a bounding box regression are carried out. The classification head assigns an object score to the RoI. The regression head refines the bounding box location tightly around the potential object. The final set of object detections are yielded by applying non-maximum-suppression (NMS). To train a Faster R-CNN detector, a multi-task loss is exploited, including L_{cls} for classification and L_{reg} for bounding box regression. L is defined as,

$$L(p, c, b^c, b_{gt}) = L_{cls}(p, c) + \lambda[c \geq 1]L_{reg}(b^c, b_{gt}), \quad (1)$$

where $L_{cls}(p, c) = -\log p_c$ is cross-entropy loss for true class c. The Iverson bracket [ ] is 1 when the inner condition is satisfied, otherwise it is 0. b_{gt} and b^c are bounding box regression targets on ground truth and predicted values for class c, individually. The hyper-parameter λ balances the two losses.

As long as we have trained the detector, we treat a video sequence as a collection of ordered images. During the test phase, we send each image frame $I_t \in \{I_1, I_2, ..., I_T\}$ into the single-image object detector one-by-one. We denote the detection result as $D_t = \{b_t^i, c_t^i\}$ where $t \in T, c_t \in K, i \in N$, if there are N traffic signs identified. Each item in D_t is a pair of bounding box b_t associated with its class c_t. The Faster R-CNN detector totally ignores the temporal consistency in consecutive frames. It inevitably leads to fluctuations in detection performance, e.g., some frames with traffic signs have either no traffic sign detected (missed detections) or incorrect detections with wrong classes or deviated locations/sizes of bounding boxes. Figure 3 shows a diagram indicating various scenarios for the detection result.

According to our observations, the signs can be detected correctly in most frames (especially, when the target is close to the camera with a big size in the recorded images). The difficulty of the task is to find the signs with wrong classifications and missed detections when the signs are pretty far from the camera, which results in a higher probability of making mistakes for the detector. Thus, it is necessary to take additional measures to deal with the failure cases. During the safety allowed reaction time, the information from frames in the future can be used to

Fig. 3 Fluctuation exists in Faster R-CNN detection result. Each geometric pattern represents a frame. Green circles represent good detections, while red rectangles and triangles represent detections with wrong class. Gray circles represent missing detections

improve the precision of detecting the traffic signs in the current frame. Higher precision of traffic sign detections can lead to better decision making in the reaction in return. Considering the different types of missing detection conditions, we divide the actual situations into two types. On one hand, the missing frames can happen between two tracklets within one potential trajectory. We call this scenario as "Two-End", as shown in Fig. 3. On the other hand, they might be associated with only one tracklet serving as one endpoint. Correspondingly, we call this scenario as "One-End".

3.2 Proposed shortest-path approach

The proposed shortest-path method works as a post-processing procedure to improve the detector's performance. As is shown in Fig. 4, the whole framework consists of several steps, which are also discussed in [58].

1) We assume a traffic sign c has been detected successfully both in frames $I_{t'}$ and $I_{t''}(t'' > t')$ with high confidence scores but missing detections occur in between. We denote that it is the i-th object in the sequence. Then we use the detection result in frame $I_{t'}$ and $I_{t''}$, i.e., $b^i_{t'}$ and $b^i_{t''}$, to generate a promising region which is supposed to contain the traffic sign. R^i is a smallest rectangle which covers both $b^i_{t'}$ and $b^i_{t''}$, as are shown in Fig. 5. For every frame I_t between $I_{t'}$ and $I_{t''}$, we use R^i to filter the whole image proposals provided from the Faster R-CNN and keep a set of candidate proposals $\{P_t\}$ within R^i. If it is "One-End" scenario, i.e., missed or incorrect detections occur before frame $I_{t''}$ without information in $I_{t'}$, we also infer a promising region according to the moving trajectory of the detected traffic signs from $I_{t''}$. The R^i can also help to identify a set of P_t, which is likely to be the traffic sign.
2) We extract the features f_t using the filtered proposals in P_t from frame I_t. To fix the length of extracted features, we apply RoI align operation on the feature map to get the embedding.
3) We construct a graph, in which the proposal boxes within R^i associated with its distinct features serve as intermedia nodes. The Euclidean distance values of every two adjacent nodes' features constitute the edges. The detections of $\{b^i_{t'}, f^i_{t'}\}$ and $\{b^i_{t''}, f^i_{t''}\}$ serve as the starting and ending nodes. In "One-End" mode, there is only either a starting or an ending node. As seen

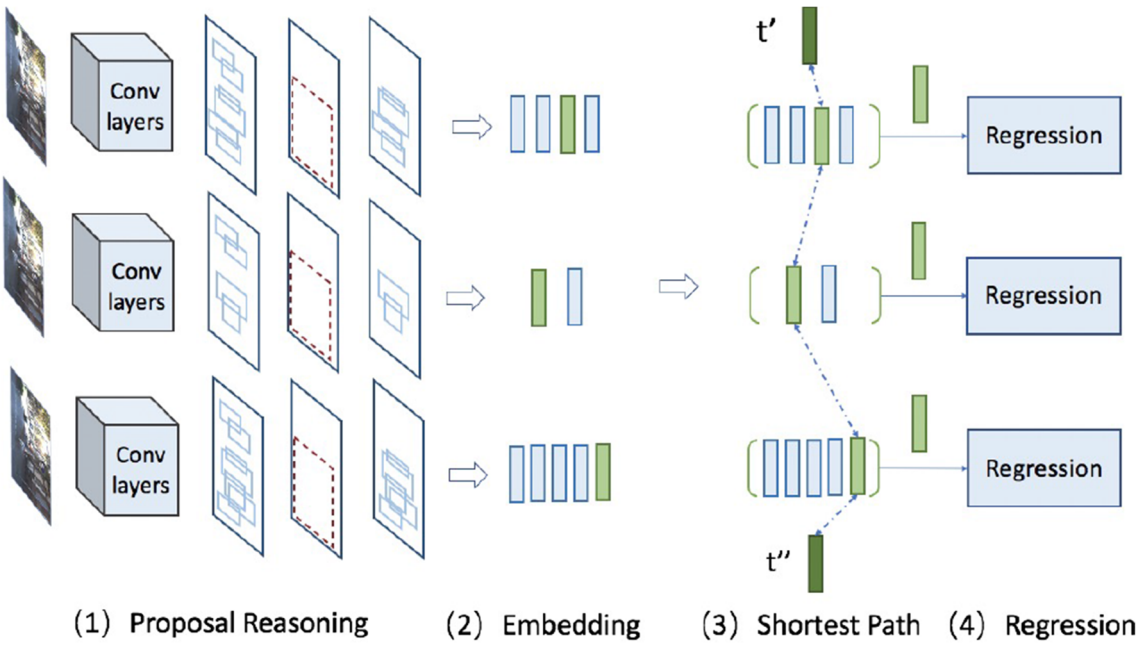

Fig. 4 The framework of our proposed algorithm

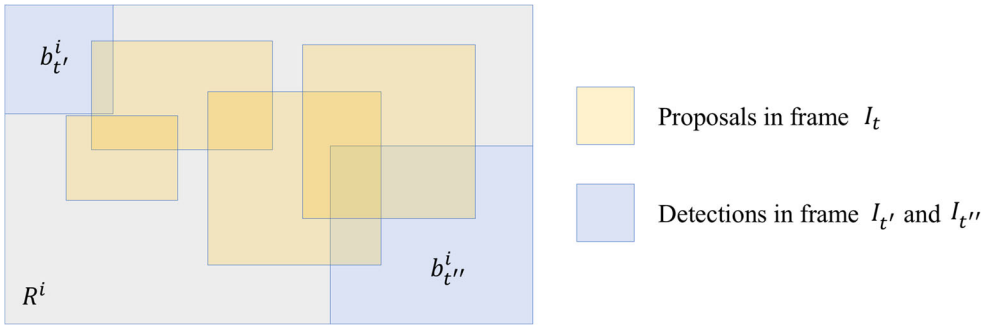

Fig. 5 Proposal filtering

from Fig. 4, a shortest path can be established for both cases through the Dijkstra algorithm [26]. A path from $\{b^i_{t'}, f^i_{t'}\}$ to $\{b^i_{t''}, f^i_{t''}\}$ with the smallest cost will be returned. In other words, the selected proposals on the shortest path have the most similar features with the detections in frame $I_{t'}$ and $I_{t''}$. It can ensure that we find the same traffic signs during these frames. The objective function in this process can be defined as,

$$\begin{aligned} j_t = \underset{j_{t'},\ldots,j_t,\ldots,j_{t''}}{argmin} \sum_{t'}^{t''} d(f_t(j_t), f_{t+1}(j_{t+1})), \\ s.t., t' \le t < t+1 \le t'', j_t \in \{P_t\}, \end{aligned} \tag{2}$$

where $d()$ is the distance measure.

4) For each proposal box (x, y, w, h) with extracted RoI feature f, we send them to the bounding box regression head of Faster R-CNN again, subjected to class c. We denote the regression function as $\phi()$ taking the RoI feature f as input. The output of the parameterized coordinates $(\phi_x(f), \phi_y(f), \phi_w(f), \phi_h(f))$ can be translated to the predicted bounding box location,

$$\begin{aligned} \hat{x} &= w\phi_x(f) + x, \\ \hat{y} &= h\phi_y(f) + y, \\ \hat{w} &= we^{\phi_w(f)}, \\ \hat{h} &= he^{\phi_h(f)}. \end{aligned} \tag{3}$$

3.3 Tractor-based method

By applying Tractor [8] into the traffic sign detection problem, we make several adaptations based on the original paper, which is mainly proposed for tracking purposes. In particular, Tracktor uses a detector for human association among frames based on the MOT dataset. Differently, we try to detect the traffic signs on the roads. In this case, traffic signs are less compact compared to the crowds in MOT datasets. We explore the temporal and spatial correlation among frames to carry out the association for the detected traffic signs with the same class. Besides, the original Tractor would use regressed boxes in later frames as detections if the score meets the requirement, with the Faster R-CNN results for initialization. In our case, we keep the Faster R-CNN detections in most trustable frames. We only recover those frames with fluctuations. Every traffic sign can be found as a trajectory in a video sequence, which is defined as the 2D bounding box coordinates along time. As for the missing part in the trajectory (e.g., gray and red patterns in Fig. 3), we do the following steps (as is shown in Fig. 6):

1) Firstly, we use the detection result $(x_{t-1}, y_{t-1}, w_{t-1}, h_{t-1})$ in frame I_{t-1} for bounding box initialization in frame I_t. We apply RoI align on the features of the current frame I_t, but with the previous bounding box coordinates $(x_{t-1}, y_{t-1}, w_{t-1}, h_{t-1})$.
2) Then, we use the regression head of Faster R-CNN to regress the bounding box of frame I_t to the object's new position (x_t, y_t, w_t, h_t) at frame I_t.
3) When we do Step 2, at the same time, the feature will be fed into the fully connected layer for classification, though we already know the class owing to the information from bilateral frames. The confidence score s_c will be obtained for the specific class c.
4) We set a threshold σ for the confidence score to decide whether we accept the new position in the current

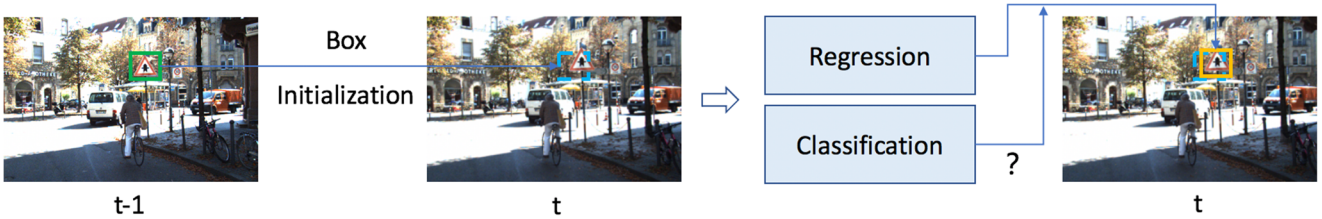

Fig. 6 The workflow of Tractor-based method

frame. If the calculated score $s_c > \sigma$, the regressed bounding box will be used to initialize the box in the next frame. If not, we terminate the trajectory.

5) Steps 1-4 are repeated for all subsequent frames till the next frame with good detection if the trajectory has not been terminated yet. Note that, in the "Two-End" mode, we also perform Steps 1-4 in the opposite direction, and then we keep the ones that possess higher confidence scores.

Note that, the assumption for using Tractor is that a target moves slightly among frames, which is usually ensured by high frame rates. It is known that the cars usually move fast on the road, thus, we set a low threshold in Step 4 to accept a potential target. Besides, the missing trajectory usually happens when the car is far from the sign, in this case, the locations of the sign in images do not change much if the car moves straightly. This observed phenomenon alleviates the influence of the assumptions from the original Tractor.

3.4 IoU-based method

Similar to Seq-NMS [22] and BBox-NMS [6], we test an approach that also utilizes IoU information and carries out NMS. We call this method as IoU-based scheme, as is shown in Fig. 7. The idea is to use high-scoring detections in nearby frames to boost scores of weaker detections. As is known to us all, during the post-processing phase of Faster R-CNN, in order to suppress false positives, a proper threshold β for the confidence score is set. Facilitated by the adjacent frames' information, we are aware of the class c of the traffic sign in the sequence. We assume frames $I_{t'}$ to $I_{t''}(t'' > t')$ are the ones that need to be recovered in a trajectory. Given a set of probable bounding boxes B in between, and their detection scores S as input, The IoU-based method works following four steps:

1) For each pair of candidate boxes in neighboring frames, a detection in the first frame can be linked with a

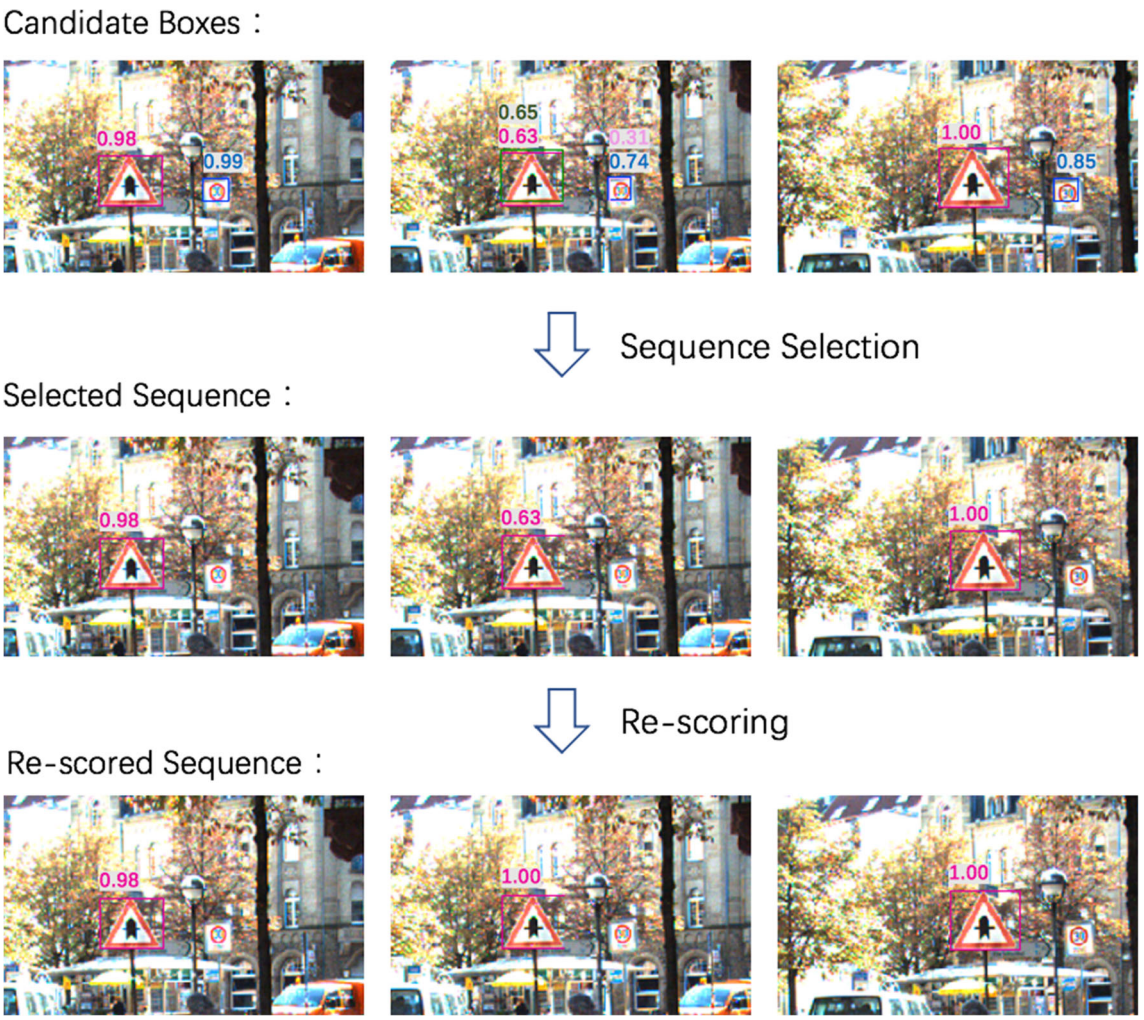

Fig. 7 The workflow of IoU-based method

candidate box affiliated to the same class in the second frame, if their IoU is above some threshold γ. Note that, considering the motion of moving cars, we set a relaxed value of γ to discover the missed target as much as possible. We find potential linkages in each pair of neighboring frames across the video segment.

2) We find the linkages which possess the maximum score across the video clip of interest. In other words, we try to find the sequence of boxes that maximizes the sum of object scores. The objective function is defined as Eq.4,

$$\begin{aligned} j_t = & \underset{j_{t'+1},\ldots,j_t,\ldots,j_{t''-1}}{argmax} \sum_{t'}^{t''} s_t[j_t], \\ & s.t., t' < t < t+1 < t'', \\ & s.t., IoU(b_t[j_t], b_{t+1}[j_{t+1}]) > \gamma, \end{aligned} \tag{4}$$

where $s_t[\ ]$ denotes score for the j_t-th box in frame I_t. The optimization formula will return a set of indices $\{j_{t'+1}, \ldots, j_t, \ldots, j_{t''-1}\}$, which can be used to locate the corresponding candidate boxes.

3) As far as we have identified a sequence with the maximum likelihood, we take some measure to boost the boxes with lower confidence scores. Specifically, we apply a function F to the sequence scores S_c subjected to class c, i.e., $S'_c = F(S_c)$. In our implementation, we define the re-scoring function as assigning the max score in neighboring frames of I'_t and I''_t to current frame I_t, i.e., the s_t is updated by $s_t = max\{s_{t'}, s_{t''}\}, t \in (t', t'')$.

4) Finally, the missing frames with re-scored higher values will be recovered as new detections. At the same time, other boxes that are not selected in the sequence are removed from the current trajectory for the specific traffic sign class c.

The above steps are repeated for every traffic sign on the road. The result will be successive in image frames.

3.5 Video object detection methods

Besides, we also investigate several video object detection methods, which are originally performed on ImageNet VID.

1) Deep Feature Flow (DFF) proposed by Zhu et al. [62]. DFF performs image recognition in sparse key frames. Then, it propagates the deep feature maps from key frames to other frames via a flow field. Yet, FlowNet [13] is employed. DFF jointly trains flow and video recognition tasks in a deep learning framework.

2) Flow Guided Feature Aggregation (FGFA) proposed by Zhu et al. [61]. Similar to DFF, FGFA also works on feature level. It improves the feature quality by aggregating nearby features along the flow motion paths to the current frame according to an adaptive weighting network. The resulting aggregated feature maps are then fed to the detection network to produce the detection result on the reference frame.

3) Sequence Level Semantics Aggregation (SELSA) proposed by Wu et al. [53]. SELSA first extracts proposals in different frames. Then, it proposes to link proposals across space-time with their semantic similarities (generalized cosine similarity). At last, it aggregates the features from other proposals with larger similarities to get a more discriminative and robust feature. The enhanced proposal features are further fed into the detection head for classification and bounding box regression.

4 Experiments

4.1 Datasets

Unlike image-based datasets, large scale video-based dataset is harder for collecting and annotating. To investigate the traffic sign detection in videos in autonomous driving scenarios, we utilize several datasets.

1) The KITTI datasets [19]. KITTI is a real-world computer vision benchmark [19], widely used in autonomous driving. To collect the data, the car is equipped with two high-resolution color and grayscale video cameras. We choose several sequences from KITTI raw dataset (left color camera) to evaluate the proposed approach for video-based traffic sign detection. The sequences (0005, 0014, 0015, 0029, and 0084) contain in total 1578 frames, where traffic signs are frequently observed. The resolution of the images is 1242×375. Since there are no ground truth annotations in KITTI for different traffic signs, we label each sign with a tight bounding box together with its class type as the ground truth.

2) The GTSDB dataset [25]. We train a basic single-image based traffic sign detector on the German traffic sign benchmark (GTSDB) [25], which provides detailed ground truth annotations for the traffic signs that appeared in the images. The resolution of the images is 1360×1024 pixels. There are 43 classes in the ground truth labeling. In our experiments, we further cluster them into 21 gross classes based on color, shape, and pattern. The images are collected in different environments, which can help our detector to learn and generalize better in the wild. GTSDB and KITTI are both collected in German, resulting in the consistency of traffic sign types. Thus, the Faster R-CNN detector trained on the GTSDB can support the basic detection in KITTI datasets.

3) The LISA datasets [38]. The LISA traffic sign dataset collects traffic sign images and videos in the U.S. with several different vehicles and cameras. The dataset consists of some long sequences made up of many short video clips. Each short video clip contains at least one kind of traffic sign. The dataset has 7855 annotations on 6610 images. There are 47 classes. We have used 4098 annotations out of 3058 images for training. The resolution of the captured frames varies from 640×480 to 1024×522 pixels and the annotations vary from 6×6 to 167×168 pixels. To evaluate the performance, we test on a long sequence called "vid4", which includes 460 images with plenty of different signs. The dataset has been preprocessed, thus, there are 34 short segments for different traffic signs within the sequence and each segment contains around 5 to 50 images subjected to the same sign.

4.2 Experiment setup

To train a baseline single-image based traffic sign detector, we have implemented the Faster R-CNN based on [36]. We have used the same parameter settings for training both on GTSDB and LISA. The shared convolutional layers are initialized by a pre-trained model for ImageNet classification (ResNet R-50-C4). A learning rate of 0.0025 is used for the first 12K iterations and is then decreased by 0.1 each time for another 4K and further 2K iterations, until it stops at 18K iterations. We use a momentum of 0.9 with a weight decay of 0.005. The experiments are carried out on an NVIDIA Quadro GV100 32GB GPU. We resize the input images such that the shortest side is at least 480 and at most 800 pixels while the longest is at most 1333 pixels. To extract the features from RoI, we have adopted RoI align measure instead of RoI pooling for better feature representation. After several aggregation operations, the feature dimension of candidates in shortest path based search is 2048.

In Section 3, we have mentioned several thresholds. For KITTI sequences, in the Tractor-based method, we empirically set $\sigma = 0.001$ as a loose reference to decide whether we accept the recovered detection or not. In the IoU-based method, we empirically set $\beta = 0.02$ to keep enough potential detections in the missing frame. We empirically set $\gamma = 0.001$ to decide the linkages in selecting sequence. Note that, the frame rate in LISA is much less than KITTI. Thus, we make an even looser parameter threshold for LISA dataset. In specific, if σ and γ are larger than 0, the results can be accepted, while the β is still set as 0.02.

For the video object detection methods, we have tested the performances based on MMTracking [11]. It provides a flexible as well as standardized toolkit to reimplement existing methods, e.g., DFF [62], SELSA [53], and FGFA [61]. We use ResNet-50 as the backbone network and adopt the Faster R-CNN in the frameworks for object detection purpose.

4.3 Results

To evaluate the detection performance, we use Average Precision (AP) and Average Recall (AR) as the metrics. The results are calculated averaged over all traffic sign classes based on IoU=0.5. The Faster R-CNN detector, as trained on the GTSDB dataset, can achieve an overall AP of 0.800 in the single-image traffic sign detection task (calculated on the test set of GTSDB). Later, we use this single-image based detector to detect traffic signs in KITTI video sequences as the first step of our algorithm, which achieves an overall AP of 0.628. Note that, the performance for single-image based traffic sign detection is not high enough since the KITTI video sequence contains more small signs when the car is far away. It is pretty challenging to detect traffic signs continuously in the wild images.

The algorithm we have introduced considers the feature similarity of candidate proposals with neighboring detections. The shortest path based algorithm can improve the performance of the original single-image based detector to a large extent. The time complexity of the Dijkstra algorithm used in our shortest path search in this process is $O(|V|^2)$, where V represents the vertexes in the graph constructed based on filtered proposals. Besides, we also investigate two other approaches described in Sections 3.3 and 3.4 as comparisons. These methods all try to take advantage of the temporal information by looking through the detected objects in neighbor frames with high confidence scores. The class label can be determined based on the high confidence score in consecutive frames, then the potential boxes in missing frames can thus be retrieved through different measures. To demonstrate the effectiveness of the algorithms, besides KITTI, we also use LISA for experiments. We process the test video sequences obtained from real-world driving scenarios. The overall results are reported in Table 1, from which we can see that the three methods discussed above show better performances since they exploit the temporal information and work as a post processing step. In specific, in the KITTI dataset, the AP is 0.628 and AR is 0.626 for the basic Faster R-CNN detector. For the Tractor-based method, AP is 0.640 and AR is 0.647. IoU-based method achieves better performance compared to Tractor-based approach. The AP and AR are 0.673 and 0.672, separately. In our proposed shortest path based method, the AP and AR can be improved to 0.733 and 0.748. Similarly, we also test the performance on LISA dataset.

Table 1 The performance of different methods. The bold number indicates the best value in the column

	AP@.50 KITTI	AR@.50 KITTI	AP@.50 LISA	AR@.50 LISA
Faster R-CNN [43]	0.628	0.626	0.661	0.663
Tractor-based [8]	0.640	0.647	0.675	0.680
IoU-based [22]	0.673	0.672	0.733	0.733
Ours (Shortest Path)	**0.733**	**0.748**	**0.775**	0.804
SELSA [53]	-	-	0.720	**0.890**
DFF [62]	-	-	0.300	0.683
FGFA [61]	-	-	0.458	0.708

The AP and AR are 0.661 and 0.663 for the basic Faster R-CNN detector. Our proposed shortest path based method can improve the metric values to 0.775 and 0.804, separately.

Above mentioned three methods (IoU-based, Tractor-based, and Shortest Path-based) all extend the Faster R-CNN and improve the detection performance by taking advantage of temporal consistency. These measures allow us to exploit a comparatively cheap way to leverage a detector for traffic sign detection in videos. We can directly benefit from the improved object detection methods to compensate for failure cases. Moreover, operating during the post-processing phase can be suitable for various detectors, in addition to Faster R-CNN, since it does not require large video data annotations for training when only images are available.

Note that, with a limited number of annotated video sequences from KITTI, it is inappropriate to train the competing VID techniques [27, 29, 52, 62] on this dataset. We carry out the competitive methods described in Section 3.5 leveraging the LISA dataset, which contains a certain number of video clips. From Table 1, we can see that the recall values of DFF, FGFA, and SELSA are pretty high. It demonstrates that correlating multiple frames information is helpful to find potential signs. The performance of SELSA is impressive compared to DFF and FGFA. Since the similarities in SELSA are built on the proposal level, they are more robust compared with the optical flow which is computed on each position in feature maps. The performances of average precision (AP) from DFF and FGFA are not good. The flownet trained on synthetic datasets may not be suitable for traffic sign detection in the wild.

Figure 8a, b, and c qualitatively show the detection results of our proposed shortest path algorithm in several video segments. Figure 8d, shows various traffic signs appeared in KITI test sequences, while Fig. 8e shows different traffic signs in LISA dataset. From Fig. 8, we can observe some phenomenon: 1) Either "One-End" or "Two-End", it works fine as the post processing step beyond the Faster R-CNN detector. The traffic signs in "bad" frames can be effectively recovered (orange boxes); 2) Compared to the red bounding boxes (wrong classifications) returned by Faster R-CNN, our recovered boxes fit well with the ground truth. 3) Detection for very small sign is still a challenge, e.g., "speed limit" sign in Fig. 8a, in spite that our method can make some improvement to a large extent.

Figure 9 shows an example of the qualitative performance using three competing methods described in Section 3. The information from neighboring frames is exploited to boost the discovery of signs in the missed and incorrect frames. In the video segment shown in Fig. 9, where the triangular sign denotes "slippery road". The traffic signs in the first three frames prior the first purple line can utilize the information from the frame denoted by the purple line. As for the two subsequent frames, they can be detected using the information from the bilateral frames denoted by the two purple lines. In other words, the first three frames satisfy "One-End" mode, the subsequent two frames meet the condition of "Two-End" mode. From this figure, we can obviously see that Tractor-based method tends to experience severe drift when the car has drastic motion. The reason can be traced back to the working procedure of the Tractor, which uses the detected box in the former frame as the initialization of the proposal box in current frame. When the car moves fast on the road, there might be little overlapping for the two signs in the image frames. Then the feature extracted from current frame cannot fully represent a sign, resulting in inaccurately regressed bounding box location. IoU-based method fails in the first three frames, due to no correct detection is achieved by Faster R-CNN on current class label. However, in the last two frames, IoU-based method can achieve much better localization results than Tractor-based method. Compared to the aforementioned two approaches, our proposed shortest path-based method not only can detect the sign successfully, but also localize the box more precisely. This is because we do not restrict the proposal box localization to the same one with other frames, which alleviates the influence from

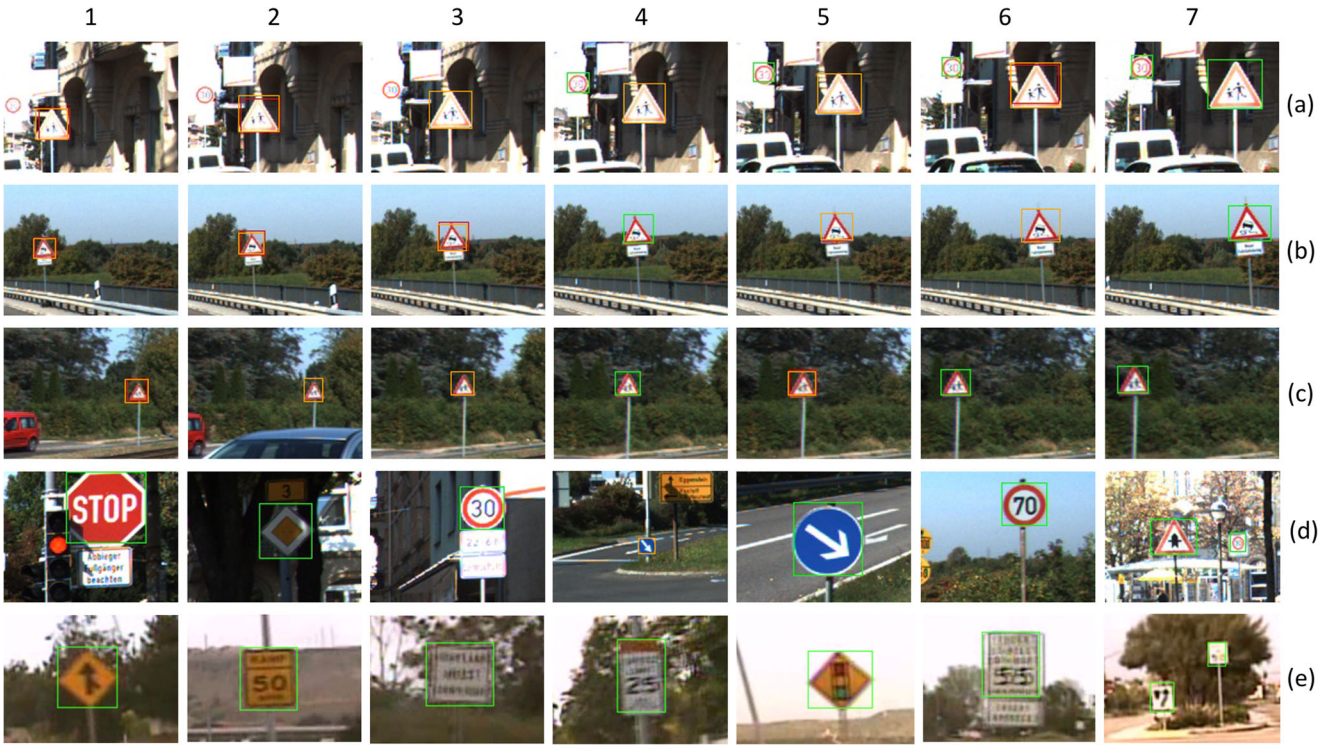

Fig. 8 Qualitative performance of our proposed shortest path based method. Example video segments are shown in (**a**), (**b**) and (**c**) with recovered or improved bounding boxes from originally missed/incorrect detections. Red boxes denote wrong detections, green boxes denote good detections with high confidence, and orange boxes denote the recovered boxes by our proposed method. **d** shows various signs appeared in the KITTI dataset. **e** shows different signs in the LISA dataset

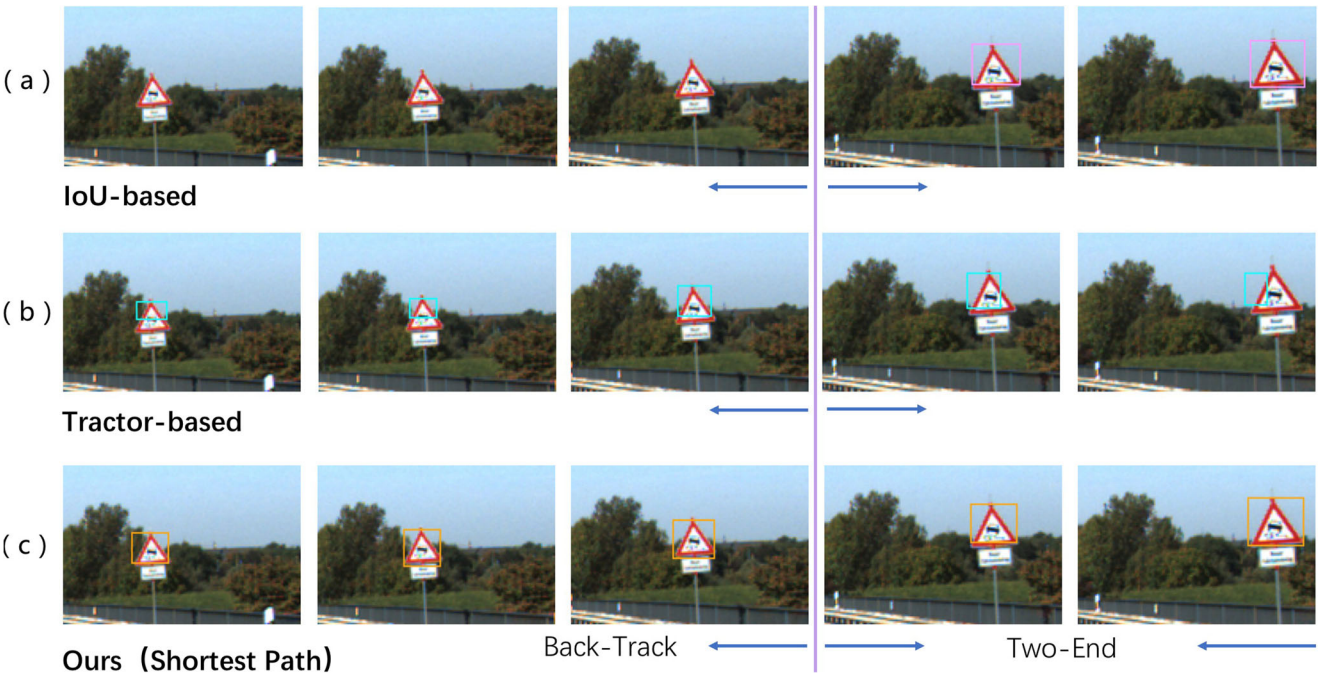

Fig. 9 The performance of three methods for recovered bounding boxes from originally missed/incorrect detections on KITTI. The purple lines represent frames with correctly detected high confidence detections. **a** Pink boxes denote the detection result using IoU-based method. **b** Cyan boxes denote the detection result using Tractor-based method. **c** Orange boxes denote the detection result using our shortest path based method

camera motion. On the other hand, we try to use the feature similarity to locate somewhere which is most likely to be a sign. A further regression step would return a better fitting box.

5 Discussion

5.1 Proposal selection scheme

One key step in our proposed approach (Section 3.2) is regarding proposal selection. Currently, we use the shortest path scheme to determine which proposals are likely to be the traffic signs based on the feature similarity. Besides, we also try a maximal confidence score based method for selecting the proposals. In other words, for every proposal in the promising region, we send them to the classification head to get the confidence score. Regressed proposal with the maximal score of the same sign category is chosen as the recovered detection. We use (5) to derive the objective function for the maximal confidence score based method.

$$j_t = \underset{j_t}{argmax} \sum_{t'}^{t''} S_c(f_t(j_t)), s.t., t' < t < t'', j_t \in \{P_t\}, \tag{5}$$

where $S_c(\,)$ mimics the confidence score for the same sign category c, obtained through the classification branch, by taking the j_t-th proposal feature $f_t(j_t)$ as input .

Since both methods operate on the same set of proposals, the selection procedures are different. Inspired by the metrics used in [27], we also apply mean absolute pixel difference (MAD), mean relative pixel difference (MRD), and mean IoU between predicted boxes and target boxes to evaluate the quality of detected bounding boxes. The comparative results are listed in Table 2. Both shortest path based and max score based selection procedures perform well with a slight offset with the ground truth annotations. The shortest path based method exhibits lower values of MAD and MRD, higher values of IoU, which indicates that it is superior to max score based method when it comes to localization precision. According to our observation, the regressed boxes returned by shortest path based method might hold a lower score through the Faster R-CNN classification head than max score based method. However, the shortest path based method fully leverage the RoI feature consistency in adjacent frames and thus lead to a better localization performance. We also randomly choose 3 KITTI sequences (0005, 0029, 0084) to show the performances. Considering the dataset's properties, the bigger the detected signs are, the higher locating accuracy can be obtained, e.g., the signs in the missing frame in "0084" are much bigger than the missing signs in the other two sequences.

5.2 Regression necessity

Some people may claim that we already select the proposal with the most consistent features, do we need to send them into the regression head again to get the final detection result as described in Section 3.2? The answer is yes. The selected proposals through the shortest path based search can provide a good embedding of feature extracted from the backbone, but the bounding box regression stage is still necessary. As is shown in Fig. 10, the candidate proposal can be more accurately located on the objects after running through the bounding box regressor. The bounding box regression plays a critical role in refining the location and size of a chosen proposal RoI. We thus use the regressed region as a better interpretation for the traffic signs.

5.3 Generalization ability

It's unfair to compare our proposed method with other kinds of single-image based object detector, since the goal of our algorithm is to improve a detector's performance in video object detection. We take additional measures beyond a vanilla detector so as to improve its original performance. In this paper, we have chosen a two-stage classical approach of Faster R-CNN as the base detector. We treat our method as a post processing step attached to a basic Faster R-CNN detector. We select the proposal which is likely to be a sign

Table 2 The comparative performance of maximum score (MS) and shortest path (SP) methods (in KITTI dataset)

	Overall			Two-End			One-End		
	MAD	MRD	IoU	MAD	MRD	IoU	MAD	MRD	IoU
MS	2.31	0.104	0.681	2.65	0.119	0.634	1.81	0.082	0.751
SP	2.06	0.093	0.706	2.31	0.103	0.667	1.68	0.079	0.765
	Seq. 0005			Seq. 0029			Seq. 0084		
	MAD	MRD	IoU	MAD	MRD	IoU	MAD	MRD	IoU
MS	3.16	0.137	0.609	1.99	0.100	0.687	2.00	0.052	0.814
SP	2.94	0.130	0.629	1.69	0.085	0.715	1.66	0.044	0.833

Fig. 10 A traffic sign example in Seq. 0005 from KITTI, showing proposal box with regressed box. **a** Proposal box in blue color; **b** Regressed box in orange color. Pink rectangle denotes the promising region for proposal selection

but was missed detected by the base detector. A further step of regression can help the proposal to localize the object more precisely. This kind of idea is simple but effective. It can be generalized to other two-stage base detectors when solving the video object detection problem, e.g., R-FCN. Compared to Tractor-based approach, our proposed scheme is better at dealing with the scenario where large motion exists in the moving camera. Thus, our method performs better in the subject of traffic sign detection in videos. Otherwise, Tractor is also supposed to be effective, e.g., on the MOT dataset, with slow-moving pedestrians captured by fixed cameras.

There might be some special signs or V2X smart transportation infrastructure for autonomous vehicles in the future. Our proposed scheme for more reliably detecting traffic signs continuously in videos can also be used for detecting other kinds of objects. The idea of detection-by-tracking can boost the detection performance taking advantage of the "tracking" information.

6 Conclusions

The problem of detecting and recognizing road signs from cameras mounted on a car is gaining more and more interest with the advent of advanced driver assisted systems or autonomous driving applications. In this paper, we try to improve the detection performance based on a single image based object detector. We have surveyed different measures which could be incorporated into the system. We investigate and implement the Tractor-based, IoU-based approach, DFF, FGFA, and SELSA. We also propose a framework, acting as a post processing procedure based on the two-stage object detector of Faster R-CNN. The detection performance can be greatly improved by leveraging the temporal information from neighboring frames. The experimental results prove the effectiveness of adopting the "track" information to boost the weaker or wrong detections. The core idea of our proposed scheme is to employ the shortest path based search over the graph constructed by proposals. The recovered detections are expected to have the most consistent features with the traffic sign of interest. The explorations we have made beyond a detector can give some insights for future researchers. Our method can also help to enhance the detection performance of other detectors, in addition to Faster R-CNN. Taking advantage of the temporal information from videos, we can detect the traffic signs more effectively beyond a single image based object detector. Considering that robustness is crucial for traffic sign detection in the vision system of autonomous cars, in the future, we will investigate into the adversarial attacks regarding the road signs in the wild.

Acknowledgements The work is supported by the Fundamental Research Funds for the Central Universities (2232021D-25) and Shanghai Sailing Program (21YF1401300).

References

1. Barnes N, Zelinsky A (2004) Real-time radial symmetry for speed sign detection. In: Intelligent vehicles symposium, 2004 IEEE
2. Barnesi N, Loy G, Shaw D, Robles-Kelly A (2005) Regular polygon detection. In: Tenth IEEE international conference on computer vision
3. Bayoudh K, Hamdaoui F, Mtibaa A (2021) Transfer learning based hybrid 2d-3d cnn for traffic sign recognition and semantic road detection applied in advanced driver assistance systems. Appl Intell 51(1):124–142
4. Behloul A, Saadna Y (2014) A fast and robust traffic sign recognition. Int J Innov Appl Stud 5(2):139–149
5. Belaroussi R, Tarel JP (2009) Angle vertex and bisector geometric model for triangular road sign detection. In: Applications of computer vision
6. Belkin I, Tkachenko S, Yudin D Traffic sign recognition on video sequence using deep neural networks and matching algorithm. In: 2019 International Conference on Artificial Intelligence: Applications and Innovations (IC-AIAI). IEEE, pp 35–354
7. Benallal M, Meunier J (2003) Real-time color segmentation of road signs. In: Electrical and computer engineering, 2003. IEEE CCECE 2003. Canadian conference on
8. Bergmann P, Meinhardt T, Leal-Taixe L (2019) Tracking without bells and whistles. In: Proceedings of the IEEE International Conference on Computer Vision, pp 941–951

9. Brkic K, Pinz A, Šegvic S (2009) Traffic sign detection as a component of an automated traffic infrastructure inventory system. In: Proceedings of the annual Workshop of the Austrian Association for Pattern Recognition, vol 1. Citeseer
10. Chen Z, Cai H, Zhang Y, Wu C, Mu M, Li Z, Sotelo MA (2019) A novel sparse representation model for pedestrian abnormal trajectory understanding. Expert Syst Appl 138(112):753
11. Contributors M (2020) MMTracking: OpenMMLab video perception toolbox and benchmark. https://github.com/open-mmlab/mmtracking
12. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE Computer society conference on computer vision and pattern recognition (CVPR'05), vol 1. IEEE, pp 886–893
13. Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van Der Smagt P, Cremers D, Brox T (2015) Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 2758–2766
14. El Margae S, Sanae B, Mounir AK, Youssef F (2014) Traffic sign recognition based on multi-block lbp features using svm with normalization. In: 2014 9Th international conference on intelligent systems: theories and applications (SITA-14). IEEE, pp 1–7
15. de la Escalera A, Moreno L, Salichs M, Armingol J (1997) Road traffic sign detection and classification. IEEE Trans Ind Electron 44(6):848–859
16. Feichtenhofer C, Pinz A, Zisserman A (2017) Detect to track and track to detect. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3038–3046
17. Fleyeh H (2004) Color detection and segmentation for road and traffic signs. In: IEEE Conference on cybernetics and intelligent systems, vol. 2, pp 809–814. IEEE
18. Garcia-Garrido MA, Sotelo MA, Martm-Gorostiza E (2006) Fast traffic sign detection and recognition under changing lighting conditions. In: Intelligent transportation systems conference, 2006. ITSC '06. IEEE
19. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on computer vision and pattern recognition (CVPR)
20. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
21. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
22. Han W, Khorrami P, Paine TL, Ramachandran P, Babaeizadeh M, Shi H, Li J, Yan S, Huang TS (2016) Seq-nms for video object detection. arXiv:1602.08465
23. Hechri A, Hmida R, Mtibaa A (2015) Robust road lanes and traffic signs recognition for driver assistance system. Int J Comput Eng 10(1/2):202–209
24. Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
25. Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C (2013) Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: The 2013 international joint conference on neural networks (IJCNN). IEEE, pp 1–8
26. Johnson DB (1973) A note on dijkstra's shortest path algorithm. J ACM (JACM) 20(3):385–388
27. Kang K, Li H, Xiao T, Ouyang W, Yan J, Liu X, Wang X (2017) Object detection in videos with tubelet proposal networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 727–735
28. Kang K, Li H, Yan J, Zeng X, Yang B, Xiao T, Zhang C, Wang Z, Wang R, Wang X et al (2018) T-cnn: Tubelets with convolutional neural networks for object detection from videos. IEEE Trans Circ Syst Video Technol 28(10):2896–2907
29. Kang K, Ouyang W, Li H, Wang X (2016) Object detection from video tubelets with convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 817–825
30. Larsson F, Felsberg M (2011) Using fourier descriptors and spatial models for traffic sign recognition. In: Scandinavian conference on image analysis. Springer, pp 238–249
31. Leichter I, Lindenbaum M, Rivlin E (2010) Mean shift tracking with multiple reference color histograms. Comput Vis Image Underst 114(3):400–408
32. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
33. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision. Springer, pp 21–37
34. Lukezic A, Vojir T, Cehovin Zajc L, Matas J, Kristan M (2017) Discriminative correlation filter with channel and spatial reliability. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 6309–6318
35. Maldonado-Bascón S, Lafuente-Arroyo S, Gil-Jimenez P, Gómez-Moreno H, López-Ferreras F (2007) Road-sign detection and recognition based on support vector machines. IEEE Trans Intell Transp Syst 8(2):264–278
36. Massa F, Girshick R (2018) maskrcnn-benchmark: fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch https://github.com/facebookresearch/maskrcnn-benchmark
37. Meier L, Tanskanen P, Heng L, Lee GH, Fraundorfer F, Pollefeys M (2012) Pixhawk: a micro aerial vehicle design for autonomous flight using onboard computer vision. Auton Robot 33(1):21–39
38. Mogelmose A, Trivedi MM, Moeslund TB (2012) Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Trans Intell Transp Syst 13(4):1484–1497
39. Munoz-Salinas R, Marin-Jimenez MJ, Medina-Carnicer R (2019) Spm-slam: Simultaneous localization and mapping with squared planar markers. Pattern Recogn 86:156–171
40. Oneata D, Revaud J, Verbeek J, Schmid C (2014) Spatio-temporal object detection proposals. In: European conference on computer vision. Springer, pp 737–752
41. Qu X, Soheilian B, Paparoditis N (2015) Vehicle localization using mono-camera and geo-referenced traffic signs. In: 2015 IEEE Intelligent vehicles symposium (IV). IEEE, pp 605–610
42. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
43. 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
44. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-fei L (2015) ImageNet Large Scale Visual Recognition Challenge. Int J Comput Vis (IJCV) 115(3):211–252. https://doi.org/10.1007/s11263-015-0816-y
45. Ruta A, Li Y, Liu X (2010) Real-time traffic sign recognition from video by class-specific discriminative features. Pattern Recogn 43(1):416–430

46. Saadna Y, Behloul A (2017) An overview of traffic sign detection and classification methods. Int J Multimed Inf Retr 6(3):193–210
47. Šegvić S, Brkić K, Kalafatić Z, Pinz A (2014) Exploiting temporal and spatial constraints in traffic sign detection from a moving vehicle. Mach Vis Appl 25(3):649–665
48. Sermanet P, Lecun Y (2011) Traffic sign recognition with multi-scale convolutional networks. In: International joint conference on neural networks, pp 2809–2813
49. Shakhuro VI, Konouchine A (2016) Russian traffic sign images dataset. Comput Opt 40(2):294–300
50. Teixeira L, Raposo AB, Gattass M (2013) Indoor localization using slam in parallel with a natural marker detector. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp 27–33
51. Viola PA, Jones MJ (2001) Rapid object detection using a boosted cascade of simple features. In: 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition
52. Wang S, Zhou Y, Yan J, Deng Z (2018) Fully motion-aware network for video object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 542–557
53. Wu H, Chen Y, Wang N, Zhang Z (2019) Sequence level semantics aggregation for video object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 9217–9225
54. Wu Y, Li Z, Chen Y, Nai K, Yuan J (2020) Real-time traffic sign detection and classification towards real traffic scene. Multimed Tools Appl 79(25):18,201–18,219
55. Youssef A, Albani D, Nardi D, Bloisi DD (2016) Fast traffic sign recognition using color segmentation and deep convolutional networks. In: International conference on advanced concepts for intelligent vision systems
56. Yu X, Fan Z, Wan H, He Y, Du J, Li N, Yuan Z, Xiao G (2019) Positioning, navigation, and book accessing/returning in an autonomous library robot using integrated binocular vision and qr code identification systems. Sensors 19(4):783
57. Zaklouta F, Stanciulescu B (2011) Warning traffic sign recognition using a hog-based k-d tree. In: Intelligent vehicles symposium
58. Zhang Y, Qi Y, Yang J, Hwang JN (2020) Improved traffic sign detection in videos through reasoning effective roi proposals. In: 2020 IEEE International conference on multimedia and expo (ICME). IEEE, pp 1–6
59. Zhang Y, Wang Z, Qi Y, Liu J, Yang J (2018) Ctsd: a dataset for traffic sign recognition in complex real-world images. In: 2018 IEEE Visual communications and image processing (VCIP). IEEE, pp 1–4
60. Zhang Y, Yang J, Zhang H, Hwang JN (2019) Bundle adjustment for monocular visual odometry based on detected traffic sign features. In: IEEE International conference on image processing (ICIP). IEEE
61. Zhu X, Wang Y, Dai J, Yuan L, Wei Y (2017) Flow-guided feature aggregation for video object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 408–417
62. Zhu X, Xiong Y, Dai J, Yuan L, Wei Y (2017) Deep feature flow for video recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2349–2358
63. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2110–2118
64. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2110–2118