Visual Traffic Knowledge Graph Generation from Scene Images

Yunfei Guo[1,2], Fei Yin[1,2], Xiao-hui Li[1,2], Xudong Yan[3], Tao Xue[3], Shuqi Mei[3], Cheng-Lin Liu[1,2]
[1]MAIS, Institute of Automation of Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]T Lab, Tencent Map, Tencent Technology (Beijing) Co., Ltd.
guoyunfei2019@ia.ac.cn, {fyin,xiaohui.li}@nlpr.ia.ac.cn,
{owenyan,emmaxue,shawnmei}@tencent.com, liucl@nlpr.ia.ac.cn

Abstract

Although previous works on traffic scene understanding have achieved great success, most of them stop at a low-level perception stage, such as road segmentation and lane detection, and few concern high-level understanding. In this paper, we present Visual Traffic Knowledge Graph Generation (VTKGG), a new task for in-depth traffic scene understanding that tries to extract multiple kinds of information and integrate them into a knowledge graph. To achieve this goal, we first introduce a large dataset named CASIA-Tencent Road Scene dataset (RS10K) with comprehensive annotations to support related research. Secondly, we propose a novel traffic scene parsing architecture containing a Hierarchical Graph ATtention network (HGAT) to analyze the heterogeneous elements and their complicated relations in traffic scene images. By hierarchizing the heterogeneous graph and equipping it with cross-level links, our approach exploits the correlation among various elements completely and acquires accurate relations. The experimental results show that our method can effectively generate visual traffic knowledge graphs and achieve state-of-the-art performance. The dataset RS10K is available at `http://www.nlpr.ia.ac.cn/pal/RS10K.html`.

1. Introduction

Traffic scene understanding [26, 3, 5, 34] plays an important role in auto drive systems. Some of its subtasks, such as road segmentation [11, 2], lane detection [43, 36], and traffic sign detection [32, 22], have made significant progress in the recent years. However, such low-level output with limited information is far from sufficient for auto drive. In this work, we introduce Visual Traffic Knowledge Graph Generation (VTKGG), a high-level traffic scene understanding task to provide comprehensive and well-formatted traffic scene information for not only auto drive, but also positioning assistance and map correction.

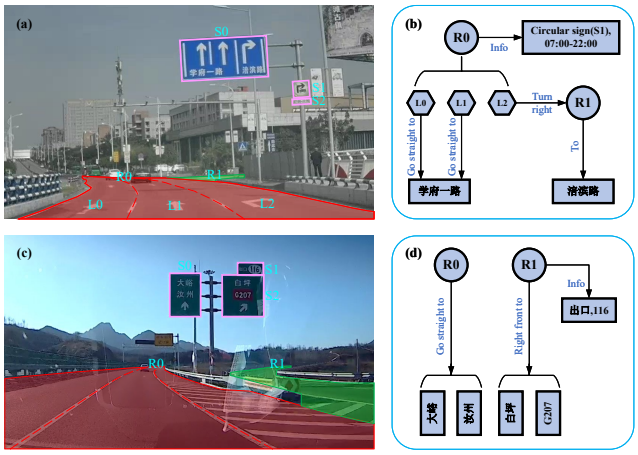

Figure 1. Traffic knowledge graphs for scene images in RS10K. Roads, lanes, and signs are noted with "R", "L", and "S", respectively. The current roads are masked by red and right roads by green. For better visualization, some annotations are simplified.

VTKGG aims to extract traffic information from traffic scene images and organize the information into a knowledge graph. Considering that the relevant techniques are quite developed, *VTKGG is carried out on the position-known traffic signs, roads, and lanes*. As shown in Figure 1, the traffic knowledge graph is a graph connecting roads, lanes, locations, warnings, etc. To accomplish this, we must parse the content of traffic signs and identify relations between sign components (i.e., texts, symbols, and arrowheads on signs) and their correspondence to traffic elements (i.e., roads and lanes), which is complicated, however. First, relations exist between traffic signs (S-S relations), which indicate one sign is a supplement to another, e.g., "S1" and "S2" in Figure 1(a) and Figure 1(c). Second, relations exist between sign components (C-C relations), which has been demonstrated in [12]. Third, relations exist

between components and traffic elements (C-T relations), since different components in the same sign, e.g., the three arrows of "S0" in Figure 1(a), may correspond to different traffic elements. All these relations are not independent. For example, C-C relations may span two related signs to connect their components and some C-T relations can be inferred from other C-C and C-T relations. Therefore, we can summarize two characteristics of VTKGG: multiple heterogeneous elements and complicated entangled relations.

Due to the aforementioned factors, pertinent studies are limited. Guo et al. [12] proposed traffic sign understanding, a task seeking to decipher traffic signs using sign components and their relations. However, it only concerns cropped distortion-free traffic panels without traffic elements. Greer et al. [10] tried to detect salient traffic signs pertaining to the current road, which is only a coarse-grained correspondence, unfortunately. To take consider all elements and obtain fine-grained relations, a straightforward approach is creating a fully connected graph that treats all the elements equally. However, this approach is inefficient for complicated scene images by neglecting the hierarchy of elements and relations. The S-S relations are high-level relations depending on high-level features. They further affect C-C relations since C-C relations may span two related signs. The C-C relations support the reasoning of C-T relations by helping the model to learn the layout of the sign. Based on the analysis above, we suggest a novel architecture using a Hierarchical Graph ATtention network (HGAT) to integrate the reasoning of S-S, C-C, and A-T relations considering their interactions, where "A" denotes the representative arrow elements selected from all components. Along with this architecture, a large dataset CASIA-Tencent Road Scene dataset (RS10K) is collected and annotated.

The overall framework proposed is shown in Figure 3. Concretely, given an input image, a sign component detector is applied to detect components on signs. Then node features are gathered from bounding boxes or masks of elements to build the input graph. After that, our HGAT performs feature refining and relation classification to get all the element relations. After all sign texts are recognized by a text recognizer and an attribute recognizer, the traffic knowledge graph is constructed from the above results. Our HGAT adopts a top-down manner to organize its layers. Three levels in each layer process subgraphs for S-S relations, C-C relations, and A-T relations respectively. To make the best of the correlation between different levels, cross-level links are created between adjacent levels to allow for cross-level interaction. Additionally, the affiliation and the relative position between traffic elements are embedded as edges between traffic elements, which we call T-T links. By deploying these techniques, HGAT can leverage the correlation between relations completely and obtain accurate relations for traffic knowledge graph generation.

RS10K has 10066 high-resolution images of various traffic scenes. Over all these images, there are 42923 traffic signs of different types. To facilitate VTKGG, comprehensive annotations, as well as knowledge graphs, are provided, including road mask, road direction, lane mask, lane type, components on signs, and all kinds of relations. As shown in Table 1, RS10K is a large versatile dataset that can also be utilized in the future for tasks like road and lane segmentation, traffic sign detection and understanding, etc.

Consequently, our contributions are as follows:

1. We propose a new valuable task VTKGG, and introduce a new large dataset RS10K with rich annotations to support it and other relevant research.

2. We propose a framework for VTKGG with HGAT to reason all relations hierarchically while leveraging their correlation effectively.

3. Experimental results on RS10K show that our framework is an effective solution for VTKGG, and achieves state-of-the-art performance compared with other relation reasoning methods.

2. Related Work

2.1. Traffic Scene Understanding

Although high-definition maps [20, 9, 29] exhibit high positioning precision and rich traffic information, the expensive development and maintenance costs greatly limit their application. In contrast, visual traffic scene understanding is cheap and real-time, thus having great application prospects. Previous work on traffic scene understanding mostly focused on low-level perception, such as road segmentation [11, 2], lane detection [43, 36, 8, 13], traffic sign detection [4, 45, 38, 41], etc., which is insufficient to achieve a complete understanding of traffic scenes. Guo et al. [12] proposed a multi-task learning framework composed of component detection, relation reasoning, classification, and semantic description to parse of traffic panels. However, the limitation is that it can only handle regular and distortion-free traffic panels and cannot directly be applied to traffic scene images like ours.

2.2. Visual Relation Detection

Visual relation detection aims to detect objects from scene images and then parse the relations between different objects. Previous methods [6, 33, 40, 35, 39] based on Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) utilize CNN or RNN or LSTM [16] to infer relations through paired object features, but they have limitations on the context fusion and inference speed. Graph Neural Network based methods [21, 37, 28] first construct a graph whose nodes stand for objects and edges for

Dataset	images	simple signs	traffic panels	roads	lanes	relations
GTSDB [17]	900	1213	-	-	-	-
CTSDB [38]	10000	17193	2072	-	-	-
CTSU [12]	5000	-	5000	-	-	31536
UAS [42]	6380	-	-	6380	-	-
CamVid [1]	700	-	-	700	-	-
CULane [26]	133235	-	-	-	298989	-
RS10K	10066	24136	18787	21058	41891	104428

Table 1. Comparison between RS10K and other popular datasets.

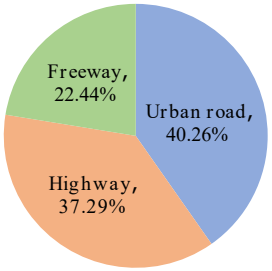

Figure 2. Proportions of different road types in RS10K.

relations, then fuse global context through message propagation. The final relations come from edge classification. Although these methods achieved great success in visual relation detection, the objects and relations they concerned are not as hierarchical as those in traffic scenes, which makes these methods not directly applicable to VTKGG.

3. Dataset

The CASIA-Tencent Road Scene dataset (RS10K) is created to support Visual Traffic Knowledge Graph Generation (VTKGG) since there is no such comprehensive dataset publicly available. This section presents the fundamental details of RS10K, involving some settings of VTKGG.

3.1. Images

RS10K contains 10066 traffic scene images which are taken by onboard cameras of various vehicles from 31 cities in China. In each city, we randomly choose some sections of high-level roads (roads above national standard level 6) to collect images. We categorize all images based on their road types and report the proportions, as shown in Figure 2, among which urban road account for a large part. Most images have a resolution of 1080×1920, and their scenes are widely distributed, including urban areas, small towns, the countryside, service centers, tollbooths, etc. The road structures are diverse, including simple roads, diversion and confluence sections, intersections, and overpasses. The whole dataset is divided into a training set with 7066 images and a testing set with 3000 images.

3.2. Annotations

RS10K contains comprehensive annotations to support VTKGG, including roads, lanes, simple signs, traffic panels, and multiple kinds of relations.

The roads and lanes for each scene image are annotated by masks of their region. For roads, we divide them depending on their locations or branches of a fork or crossroad, i.e., the current road, the left road, the right road, and the front road, which is a requisite for VTKGG but not provided in previous datasets [42, 1].

The positions of all traffic signs are annotated with quadrilateral boxes, including simple signs and complex panels. One sort of undirected S-S relation is annotated to indicate whether the two signs are related or not. In traffic panels, we annotated their components in detail, including texts, symbols (59 categories), and arrowheads (i.e., the heads of the arrow symbols, 8 categories), making a total of 68 categories. The definition of C-C relation is consistent with [12], that is, two types of undirected C-C relations: association relation and pointing relation. The association relation implies that multiple components represent a same place or a same piece of information. Pointing relation refers to the relation that a place is on the direction of an arrowhead. Notably, these relations may span across different traffic signs, but this only occurs between related signs. In addition, to distinguish different types of information, we provide attribute annotation for each text, i.e., road code, road name, place name, direction, and description.

In RS10K, one type of unidirectional C-T relation is defined to denote whether the component refers to the traffic element or not. Due to a large number of components in all signs and huge annotation costs, the C-T relation is condensed into the A-T relation, since the arrow elements, i.e., arrow symbols and arrowheads in components, have the highest correlation with traffic elements and all C-T relations can be inferred from the C-C and A-T relations. Therefore, we annotate the A-T relation rather than the C-T relation. For the signs without arrow elements, since most of them are simple and their components only pertain to one same traffic element, we directly annotate the corresponding relations between the sign and traffic elements. These signs will be represented by a virtual arrow element in our framework when reasoning A-T relations.

The traffic knowledge graphs can be generated from the above annotations. Each traffic knowledge graph is composed of several knowledge trees with roads or virtual roads as their roots and all other information is attached to the trees based on relations and information types, where virtual roads are indicated by arrow elements whose traffic elements are invisible. Please see the supplementary material for more details on traffic knowledge graphs.

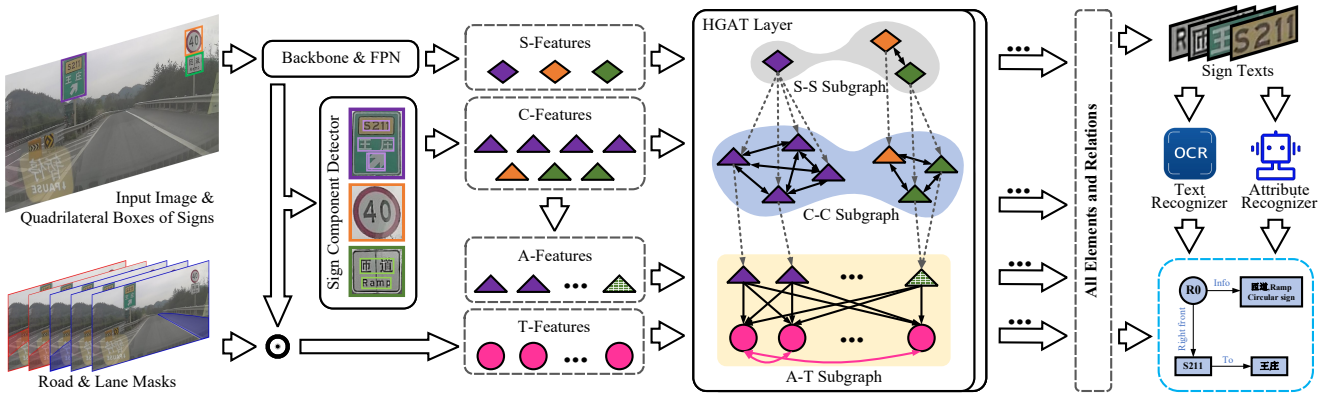

Figure 3. The overall framework of our proposed method. HGAT is composed of multiple layers. The structure of the input graph is shown in the HGAT layer, where nodes denoted by the same shape are homogeneous and the triangle with grids is a virtual arrow element. Edges representing potential relations are denoted by black solid lines, while the cross-level links are represented by dotted lines. Pink lines stand for the T-T links between traffic elements. S: sign, C: sign component, A: arrow element, T: traffic element, $\odot$: element-wise production.

3.3. Metrics

We first evaluate the two subtasks of VTKGG: sign component detection and all relation classification. For detection, we evaluate its multi-class recall, precision, and F1-measure. For relation classification, since there are S-S, C-C, and A-T relations, we calculate the recall and precision of these three relations, and finally calculate the overall F1-measure and mean F1-measure of all relations.

To assess the traffic knowledge graph more reasonably, we offer a new evaluation method TFPM based on Tree Full Path Matching. For a leaf node, only when the entire path from the root node to the leaf node is completely matched with the ground truth can it be considered as a true positive. The recall, precision, and F1-measure are adopted as the final metrics of TFPM. Depending on the node matching mechanism used, TFPM is divided into TFPM-B for matching by bounding boxes, TFPM-T for matching by texts, and TFPM-BT for matching by both of them.

3.4. Statistics

RS10K is the first large-scale and comprehensive dataset for VTKGG with rich annotations. As shown in Table 1, we compare it to some popular datasets for traffic scene parsing. Clearly, these datasets only concentrate on one or several subtasks and are unable to offer sufficient data to support in-depth traffic scene understanding. While RS10K includes various annotations, which can be utilized for not only VTKGG, but also other tasks like road and lane segmentation, sign detection, traffic sign understanding, etc. The challenges of RS10K can be summarized as follows:

Diversity of scenes. RS10K includes a variety of scenes and a large part of them are in urban areas where there are numerous vehicles, pedestrians, and billboards, which will cause occlusion and interference.

Heterogeneity of data. The annotations of different elements are heterogeneous, e.g., masks for roads and lanes while bounding boxes for sign and sign components. The scale of elements also varies widely from tiny symbols on signs to roads with large areas.

Complexity of relations. The three sorts of relations have many problems to address including entanglement, hierarchies, and ambiguity, which will confuse the model if not properly handled.

4. Approach

The pipeline of our framework is depicted in Figure 3. Given an input image and quadrilateral boxes of signs, a sign component detector is first adopted to acquire the quadrilateral bounding boxes of components inside the signs. Then node features are extracted from the multi-scale backbone features to build the input graph, where traffic element features are collected by the given road and lane masks. After that, the input graph is sent to HGAT for feature refining and relation classification to obtain all element relations. In the end, the sign texts are recognized by the text recognizer and attribute recognizer to support the final traffic knowledge graph generation. In the following, we will describe these four procedures sequentially. Some basic notations are given in Table 2.

4.1. Sign Component Detector

In our approach, we take FCOS [30] for rotated object detection from [44] as our base sign component detector. FCOS is a one-stage and anchor-free detector, whose core idea is to regress the distances between the pixels in the central area and the bounding box of the target. To detect targets of different scales, it assigns targets of different sizes to feature maps of different scales. However, this method is

e_*	One of the elements, where e in $\{s, c, a, t\}$, $*$ in $\{i, j, k, l\}$, corresponding to sign, component, arrow element, and traffic element respectively.	F_b	Three-scale feature output from FPN [23], in form of $\left[F_b^0, F_b^1, F_b^2\right]$ sized $h \times w \times d$, $h/2 \times w/2 \times d$, and $h/4 \times w/4 \times d$ respectively.
Qbox_{s_i}	The quadrilateral box of s_i, sized 1×8.	F_{bs_i}	Three-scale feature RoIAligned by Rbox_{si} from F_b, in form of $\left[F_{bs_i}^0, F_{bs_i}^1, F_{bs_i}^2\right]$ sized $W \times W \times d$, $W/2 \times W/2 \times d$, and $W/4 \times W/4 \times d$ respectively.
Rbox_{s_i}	The regular bounding rectangle of Qbox_{s_i}, sized 1×4.	L_{e_*}	Category of element e_*.
$\mathrm{Qbox}_{c_j}^{s_i}$	The quadrilateral box of c_j in s_i, sized 1×8.	N_{e_*}	Node feature of element e_*, sized $1 \times d$.
M_{s_i}	The mask of s_i in $F_{bs_i}^1$, sized $W/2 \times W/2$.	V_{e_*}	Visual feature of element e_*, sized $1 \times d$.
$M_{c_j}^{s_i}$	The mask of c_j in $F_{bs_i}^0$, sized $W \times W$.	Y_{e_*}	Semantic feature of element e_*, sized $1 \times d$.
M_{t_l}	The mask of t_l in F_{bs}^2, sized $h/4 \times w/4$.	P_{e_*}	Spatial feature of element e_*, sized $1 \times d$.

Table 2. Overview of some basic notations mentioned in the approach section.

still insufficient to detect texts and symbols on distant or small signs. To address this problem, for the i-th sign s_i, we first resize its rectangular area inside Rbox_{s_i} in F_b into square window features F_{bs_i} by RoIAlign [14], and then apply the detection head $\mathrm{RFCOSHead}$ of rotated FCOS to the largest-scale window $F_{bs_i}^0$ to obtain sign components. The whole process can be described as:

$$F_{bs_i} = \mathrm{RoIAlign}\left(F_b, \mathrm{Rbox}_{s_i}\right), \tag{1}$$

$$\mathrm{Qbox}_{c_0}^{s_i}, \mathrm{Qbox}_{c_1}^{s_i}, ... = \mathrm{RFCOSHead}\left(F_{bs_i}^0\right). \tag{2}$$

4.2. Input Graph Construction

After sign component detection, we get all of the following: quadrilateral boxes and categories of signs (given), quadrilateral boxes and categories of sign components (detected), masks and categories of traffic elements (given). To obtain the input graph for each scene image, we gather node features first. All node features are fused by visual features, spatial embedding, and semantic embedding features.

$$N_{e_*} = V_{e_*} + P_{e_*} + Y_{e_*}. \tag{3}$$

S-Features. For s_i, visual feature is calculated as the mean vector of the window feature $F_{bs_i}^1$ inside the quadrilateral box after several convolution layers Conv.

$$V_{s_i} = \mathrm{sum}\left(\mathrm{Conv}\left(F_{bs_i}^1\right) \odot M_{s_i}\right) / \mathrm{sum}\left(M_{s_i}\right), \tag{4}$$

where sum means the sum operation on the first two dimensions; $\odot$ stands for element-wise production. Semantic feature of s_i is the embedding of its category L_{s_i}.

$$Y_{s_i} = \mathrm{Embedding}(L_{s_i}). \tag{5}$$

The spatial feature of s_i is generated by a linear function $f_r : R^{12} \rightarrow R^d$ to transform the position of s_i.

$$P_{s_i} = f_r\left(\mathrm{Rbox}_{s_i} \parallel \mathrm{Qbox}_{s_i}\right), \tag{6}$$

where $\parallel$ denotes the concatenation operation.

C-Features. The visual feature $V_{c_j}^{s_i}$, semantic feature $Y_{c_j}^{s_i}$, and spatial feature $P_{c_j}^{s_i}$ of c_j in s_i are obtained in the same manner as signs but on $F_{bs_i}^0$ to include more details.

$$V_{c_j}^{s_i} = \mathrm{sum}\left(\mathrm{Conv}\left(F_{bs_i}^0\right) \odot M_{c_j}^{s_i}\right) / \mathrm{sum}\left(M_{c_j}^{s_i}\right), \tag{7}$$

$$Y_{c_j}^{s_i} = \mathrm{Embedding}(L_{c_j}^{s_i}), \tag{8}$$

$$P_{c_j}^{s_i} = f_r\left(\mathrm{Rbox}_{s_i} \parallel \mathrm{Qbox}_{c_j}^{s_i}\right). \tag{9}$$

A-Features. The node features of arrow elements in the A-T subgraph are cloned from their features in the C-C subgraph. For the sign without arrow symbols or arrowheads, we create a virtual arrow element to represent the whole sign, whose feature is the mean vector of all components on the sign.

T-Features. For t_l, its visual feature is extracted from the backbone feature F_b^2 by its mask M_{t_l} since F_b^2 contains more high-level information.

$$V_{t_l} = \mathrm{sum}\left(\mathrm{Conv}\left(F_b^2\right) \odot M_{t_l}\right) / \mathrm{sum}\left(M_{t_l}\right), \tag{10}$$

We embed for current road, left road, right road, front road, and lanes to get the semantic features.

$$Y_{t_l} = \mathrm{Embedding}(L_{t_l}). \tag{11}$$

At last, the mask M_{t_l} is transformed into P_{t_l} by a linear projection $f_u : R^{hw} \rightarrow R^d$.

$$P_{t_l} = f_u\left(\mathrm{flatten}\left(M_{t_l}\right)\right), \tag{12}$$

where $\mathrm{flatten}$ means a flattening operation.

Graph Structure. To reduce redundant connections and calculations, the input graph is built into a globally sparse but locally fully connected graph, as shown in Figure 3.

In the S-S subgraph, the bidirectional edge (represented by two oppositely directed edges in the graph) is adopted for the reasoning of the undirected S-S relation. Some S-S relations can be prejudged by their relative distance and alignment using the heuristic method. For example, there are no relations between distant and spatially misaligned signs. Therefore, such a heuristic method is employed to determine edges in the S-S subgraph to make it sparse.

In the C-C subgraph, all C-C relations are inferred by bidirectional edges. The relation between components is hard to predict in advance [12]. Consequently, all components of a single sign or all components of two signs with an edge in the S-S subgraph are connected fully to infer C-C relations within a sign or across different signs.

In the A-T subgraph, the A-T edge flows from every arrow element to every traffic element, while there is no connection between the arrow elements.

The S-S, C-C, and A-T edges in the three subgraphs are represented by black solid lines in Figure 3. For one of these edges from e_m to e_n, the visual and semantic features are the sum of its two endpoint features.

$$V_{mn} = V_{e_m} + V_{e_n}, Y_{mn} = Y_{e_m} + Y_{e_n}. \tag{13}$$

The spatial feature is the difference.

$$P_{mn} = P_{e_m} - P_{e_n}. \tag{14}$$

The final edge feature is as follows.

$$E_{mn} = V_{mn} + Y_{mn} + P_{mn}. \tag{15}$$

There are also some relations between traffic elements, such as the affiliation between roads and lanes: a lane subordinate to a road, or a road superordinate to a lane, and the relative position between roads or lanes: on left, on right, in front of, behind. All of them are known given the category and location of traffic elements. These relations are embedded as features of T-T edges between traffic elements, which we call T-T links as shown in Figure 3. T-T links only exist between the road and its lanes or two traffic elements adjacent to each other.

Cross-level links only exist from a sign to its components, the same arrow element from C-C subgraph to the A-T subgraph, and all components on a sign without an arrow element to its virtual arrow elements, which is single-directional. Cross-level links do not contain edge features, but only provide message propagation channels.

4.3. Hierarchical GAT

The Hierarchical GAT (HGAT) is proposed to deal with the input graph with heterogeneous nodes and edges in a top-down manner hierarchically. It includes three update mechanisms: Bidirectional Attention (BiAtt), Cross-Level Attention (CLAtt), and Bipartite Matching (BiMat). By utilizing them, HGAT performs feature refining layer by layer as shown in Figure 3.

BiAtt (for S-S and C-C subgraphs). BiAtt is imposed on subgraphs with homogeneous nodes to update its node and edge features, whose motivation is for context fusion in the subgraph. N_m stands for initial node feature of e_m, and E_{mn} for feature of edge e_m to e_n. N'_m and E'_{mn} denote the updated features.

For N_n, we first calculate its attention coefficient with its in-flow edges, denoted by α_{mn}.

$$\alpha_{mn} = \frac{\exp\left(\sigma\left(f_\alpha\left(E_{mn} \parallel N_n\right)\right)\right)}{\sum\limits_{z \in \mathbb{N}_n} \exp\left(\sigma\left(f_\alpha\left(E_{zn} \parallel N_n\right)\right)\right)}, \tag{16}$$

where $f_\alpha : R^{2d} \to R$; $\mathbb{N}_n$ is the neighborhood of e_n; σ is the activation function. Then we aggregate the in-flow edge features into N_n.

$$N'_n = \sigma\left(f_{N1}\left(\left(\sum_{z \in \mathbb{N}_n} \alpha_{zn} E_{zn}\right) \parallel N_n\right)\right), \tag{17}$$

where $f_{N1} : R^{2d} \to R^d$. The edge feature is updated by aggregating features of its two endpoints with bidirectional attention coefficients.

$$E_{mn} = \sigma\left(f_E\left(\frac{\alpha_{nm} N_m + \alpha_{mn} N_n}{\alpha_{nm} + \alpha_{mn}}\right) \parallel E_{mn}\right), \tag{18}$$

where $f_{E1} : R^{2d} \to R^d$. In this way, the edge can integrate features of both source and destination nodes.

CLAtt (for cross-level links). CLAtt fuses the source node features into the destination nodes through the attention mechanism, whose motivation is to achieve cross-level communication. Here we denote the source node by p_m and the destination node by q_n.

The feature of the source node is first processed by a linear transformation $f_{N2} : R^d \to R^d$.

$$N'_{p_m} = \sigma\left(f_{N2}\left(N_{p_m}\right)\right). \tag{19}$$

Then we calculate the attention coefficient between p_m and q_n, which can be formulated as:

$$\beta_{mn} = \frac{\exp\left(\sigma\left(f_\beta\left(N'_{p_m} \parallel N_{q_n}\right)\right)\right)}{\sum\limits_{p_z \in \mathbb{N}_{q_n}} \exp\left(\sigma\left(f_\beta\left(N'_{p_z} \parallel N_{q_n}\right)\right)\right)}, \tag{20}$$

where $f_\beta : R^{2d} \to R$. β_{mn} is directly set to 1.0 when applied to cross-level links between S-S and C-C subgraphs. Finally, all source node features are fused into their destination nodes through $f_{N3} : R^{2d} \to R^d$ and a residual link.

$$N'_n = N_n + \sigma\left(f_{N3}\left(\left(\sum_{p_z \in \mathbb{N}_n} \beta_{zn} N'_{p_z}\right) \parallel N_{q_n}\right)\right). \tag{21}$$

BiMat (for A-T subgraph). BiMat performs A-T edge feature updation for the final matching between arrow and traffic elements.

To make the model perceive the affiliation and relative position between traffic elements, we first apply BiAtt to T-T links, where, concretely, $\mathbb{N}_n$ is the traffic element nodes in the neighborhood of t_n. Then we denote the updated traffic element node feature N'_{t_n}, arrow element feature N_{a_m}, and edge feature between a_m and t_n E_{mn}. Therefore, the updated edge feature E'_{mn} is calculated by:

$$E'_{mn} = \sigma\left(f_{E2}\left(N_{a_m} \parallel E_{mn} \parallel N'_{t_n}\right)\right), \tag{22}$$

where $f_{E2} : R^{3d} \to R^d$.

Method	Detection	Relation Reasoning					VTKGG			FPS
		S-S	C-C	A-T	Overall	Mean	TFPM-B	TFPM-T	TFPM-BT	
FCOS [44]	0.853	-	-	-	-	-	-	-	-	-
GCN [18]	0.893	0.859	0.789	0.802	0.800	0.816	0.791	0.774	0.762	**5.4**
GAT [31]	0.903	0.848	0.807	0.813	0.810	0.822	0.802	0.784	0.773	4.4
aGCN [37]	0.902	0.849	0.806	0.812	0.809	0.822	0.803	0.785	0.774	4.5
HL-Net [25]	0.895	0.861	0.806	0.806	0.808	0.825	0.799	0.780	0.769	3.4
HGAT-U	0.900	0.819	0.792	0.810	0.799	0.807	0.796	0.779	0.767	4.2
HGAT-S	0.901	0.867	0.812	0.813	0.814	0.831	0.808	0.788	0.776	5.1
HGAT	**0.903**	**0.878**	**0.821**	**0.823**	**0.823**	**0.84**1	**0.812**	**0.792**	**0.781**	4.7

Table 3. Performance of some methods on RS10K. Except for FPS, all results are F1-measures.

Relation Classification. The output edge features of the S-S, C-C, and A-T subgraphs are employed for the relation classification through three linear classifiers respectively. Then the output probabilities are filtered by three thresholds to get the final relations. When testing, S-S relations are leveraged to filter C-C relations between unrelated signs to improve the precision. During training, the focal loss [24] is adopted as the final loss function for S-S relation loss L_{SSrel}, C-C relation loss L_{CCrel}, and A-T relation loss L_{ATrel}. Therefore, the total loss can be expressed by:

$$L_{\mathrm{total}} = L_{\mathrm{FCOS}} + L_{\mathrm{rel}}, \tag{23}$$

$$L_{\mathrm{rel}} = \lambda_1 L_{\mathrm{SSrel}} + \lambda_2 L_{\mathrm{CCrel}} + \lambda_3 L_{\mathrm{ATrel}}, \tag{24}$$

where λ_1, λ_2, and λ_3 are hyperparameters.

4.4. Traffic Knowledge Graph Generation

To generate the traffic knowledge graph, a crucial step is to recognize the sign texts and their attributes. As shown in Figure 3, after relation reasoning, we gather all sign texts and input them into an external text recognizer and an attribute recognizer respectively. The text recognizer derives from [7], which is a model for scene text recognition trained on RS10K along with other closed-source data. The attribute recognizer is a visual classifier trained on RS10K. At last, after we obtain the contents and attributes of the sign texts, the traffic knowledge graph is generated by arranging all components and traffic elements into a predefined format of graph structure through postprocessing.

5. Experiments

To evaluate the performance of the proposed method, we conduct some comparative and ablation experiments, all of which are carried out under the same conditions.

5.1. Implementation Details

Our implementation is built on PyTorch [27] and MMRotation [44] framework. The stem network adopts ResNet-50 [15], which inherits parameters trained on ImageNet dataset [19]. HGAT is composed of two stacked layers with 512 as its feature vector dimension d. In training and testing, we scale the long edge of input images to 1920 pixels while keeping the aspect ratio. The model is trained on the training set for 32 epochs with a batch size of 8. The learning rate is set to 0.01 at the beginning and decays with a rate of 0.1 at epoch 24 and epoch 30. Hyperparameters λ_1, λ_2, and λ_3 are all set to 1.0.

5.2. Comparison with Other Methods

As shown in Table 3, to prove the superiority of our framework, we compare it with several classical and state-of-the-art methods.

We first evaluate the detection results of the Rotated FCOS [44], which is directly applied to the whole scene image. For fairness, invalid detection outside traffic signs will be removed. As shown in Table 3, our improved detector outperforms the origin by 0.05 in sign component detection (0.903 vs 0.853).

Several relation reasoning methods are applied to RS10K, including classical graph neural network GCN [18] and GAT [31], and visual relation detection methods aGCN [37] and HL-Net [25]. Notably, we only replace the message propagation mechanism of our HGAT with these methods, while other parts, including the sign component detector, the input graph and recognizers, remain unchanged. As shown in Table 3, our HGAT outperforms other methods when evaluating the S-S, C-C, and A-T relations. HGAT surpasses the best method HL-Net by 0.016 on the mean F1-measure (0.841 vs 0.825).

5.3. Ablation Studies

Why Hierarchical. The difference between the three relations is obvious. The S-S relation mainly focuses on the appearance of signs and the distance between them, the C-C relation concerns the spatial position inside the sign and the type of components more, and the A-T relation lay more stress on the relative position between signs and traffic

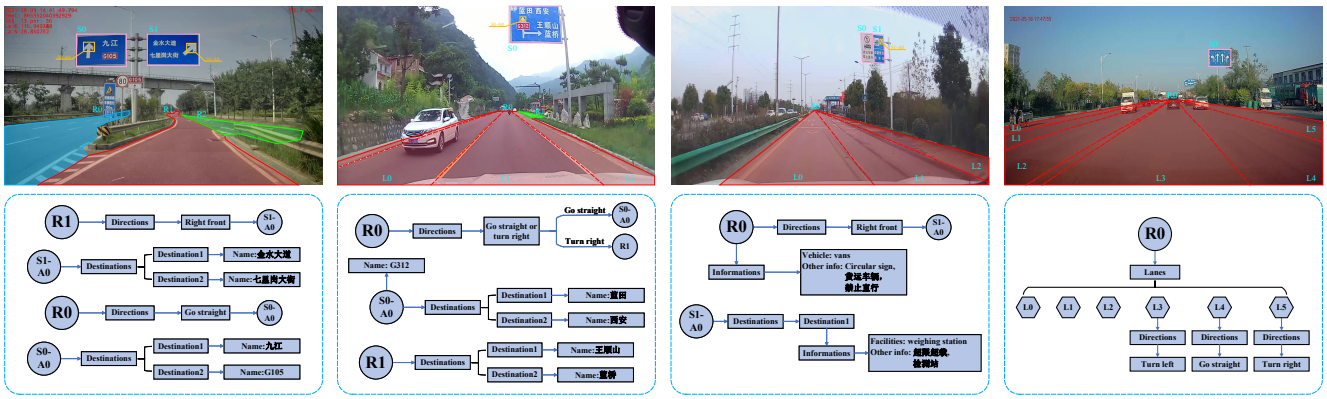

Figure 4. Visualization results of our approach. The first row displays the scene images and the second shows the generated knowledge graphs, in which the roads or virtual roads are denoted by circles, and lanes are hexagons. The current road is in red, the left road in blue, the right road in green. The lane dividing lines are highlighted by red dashed lines inside the road areas. For better visualization, we only show the information provided by the traffic panels inside pink bounding boxes.

HGAT w/o	S-S	C-C	A-T	Overall	Mean
S-S	0.878	0.800	0.823	0.810	0.834
T-T	0.860	0.805	0.815	0.810	0.827
SeF	0.853	0.797	0.809	0.803	0.820
SpF	0.873	0.803	0.811	0.808	0.829
-	0.878	0.821	0.823	0.823	0.841

Table 4. Ablation studies. "S-S" denotes S-S relations; "T-T" stands for T-T links; "SeF" means the semantic node feature Y_{e_*}. "SpF" represents the spatial node feature P_{e_*}. All results are the F1-measures for relation reasoning.

elements and layout of components. Therefore, it is natural and reasonable to employ three levels to reason these three relations. To prove this, we first convert the input graph into a homogeneous graph, and then refine the graph with HGAT-U, a GNN containing only BiAtt layers. As shown in Table 3, HGAT outperforms HGAT-U by 0.024 (0.823 vs 0.799) on the overall F1-measure and 0.034 (0.841 vs 0.807) on the mean F1-measure over all relations.

Cross-Level Links. These three sorts of relations are not independent. As mentioned earlier, the relations of the upper level are helpful to the below. Therefore, cross-level links and CLAtt are introduced to realize cross-level interaction. To verify their effectiveness, we suggest HGAT-S by removing cross-level links and CLAtt of HGAT. As shown in Table 3, the experiment on RS10K shows that CLAtt increases the overall F1-measure by 0.009 (0.823 vs 0.814) and the mean F1-measure by 0.01 (0.841 vs 0.831).

S-S Relations. When testing, S-S relations are used for filtering invalid C-C relations across two unrelated signs to improve the precision of C-C relations. To prove the effectiveness of S-S relations, we adopt the heuristic rules to replace the S-S relations predicted, as shown in Table 4, which causes a reduction of 0.021 on the F1-measure of the C-C relations (0.800 vs 0.821).

T-T Links. The affiliation and the relative position between traffic elements are also critical. To verify their significance, we remove T-T links and conduct the experiment. As shown in Table 4, T-T links improve the model by 0.014 on the mean F1-measure (0.841 vs 0.827).

Semantic and Spatial Node Features. Semantic and spatial features are as crucial as visual features to the model. We conduct experiments by removing the two features and compare them with the complete model. As shown in Table 4, the two features improve the model by 0.021 (0.841 vs 0.820) and 0.012 (0.841 vs 0.829) on mean F1-measure.

Visualization Analysis. As shown in Table 3, our approach achieves a good performance (0.786 on TFBM-BT). Some visualization results are displayed in Figure 4, showing that our framework can generate traffic knowledge graphs effectively for various traffic scenes. For more comparisons of visualization results, please refer to the supplementary material.

6. Conclusion

In this paper, we introduce VTKGG, a new task that aims at extracting traffic knowledge from scene images and formatting it into a graph. To achieve this, we create a large dataset with comprehensive annotations to support VTKGG. Meanwhile, a novel framework is introduced by us to generate traffic knowledge graphs through sign component detection, relation reasoning, and text and attribute recognition, in which our HGAT performs relation reasoning in a top-down manner to utilize the correlation in different types of relations. Experiments show that our framework achieves good performance and generates traffic knowledge graphs effectively.

Acknowledgments This work has been supported by the National Key Research and Development Program Grant 2018AAA0100400, and the National Natural Science Foundation of China (NSFC) Grants U20A20223 and 61721004.

References

[1] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *European Conference on Computer Vision (ECCV)*, pages 44–57, 2008.

[2] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[3] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15661–15670, 2021.

[4] Yajie Chen and Linlin Huang. Chinese traffic panels detection and recognition from street-level images. In *MATEC Web of Conferences*, volume 42, page 06001, 2016.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[6] Bo Dai, Yuqi Zhang, and Dahua Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3076–3086, 2017.

[7] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7098–7107, 2021.

[8] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17062–17070, 2022.

[9] Farouk Ghallabi, Fawzi Nashashibi, Ghayath El-Haj-Shhade, and Marie-Anne Mittet. Lidar-based lane marking detection for vehicle positioning in an hd map. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2209–2214, 2018.

[10] Ross Greer, Jason Isa, Nachiket Deo, Akshay Rangesh, and Mohan M Trivedi. On salience-sensitive sign classification in autonomous vehicle path planning: Experimental explorations with a novel dataset. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 636–644, 2022.

[11] Shuo Gu, Yigong Zhang, Jinhui Tang, Jian Yang, and Hui Kong. Road detection through crf based lidar-camera fusion. In *International Conference on Robotics and Automation (ICRA)*, pages 3832–3838, 2019.

[12] Yunfei Guo, Wei Feng, Fei Yin, Tao Xue, Shuqi Mei, and Cheng-Lin Liu. Learning to understand traffic signs. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pages 2076–2084, 2021.

[13] Jianhua Han, Xiajun Deng, Xinyue Cai, Zhen Yang, Hang Xu, Chunjing Xu, and Xiaodan Liang. Laneformer: Object-aware row-column transformers for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 799–807, 2022.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[17] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2013.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems (NIPS)*, 25:1097–1105, 2012.

[20] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *International Conference on Robotics and Automation (ICRA)*, pages 4628–4634, 2022.

[21] Yikang Li, Wanli Ouyang, Bolei Zhou, Jianping Shi, Chao Zhang, and Xiaogang Wang. Factorizable net: an efficient subgraph-based framework for scene graph generation. In *European Conference on Computer Vision (ECCV)*, pages 335–351, 2018.

[22] Tianjiao Liang, Hong Bao, Weiguo Pan, and Feng Pan. Traffic sign detection via improved sparse r-cnn for autonomous vehicles. *Journal of Advanced Transportation (JAT)*, 2022:1–16, 2022.

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[25] Xin Lin, Changxing Ding, Yibing Zhan, Zijian Li, and Dacheng Tao. Hl-net: Heterophily learning network for

scene graph generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19476–19485, 2022.

[26] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 7276–7283, 2018.

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. volume 32, pages 8026–8037, 2019.

[28] Mengshi Qi, Weijian Li, Zhengyuan Yang, Yunhong Wang, and Jiebo Luo. Attentive relational networks for mapping images to scene graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3957–3966, 2019.

[29] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized hd-map construction with piecewise bezier curve. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13218–13228, 2023.

[30] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9627–9636, 2019.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[32] Junfan Wang, Yi Chen, Zhekang Dong, and Mingyu Gao. Improved yolov5 network for real-time multi-scale traffic sign detection. *Neural Computing and Applications (NCA)*, pages 1–13, 2022.

[33] Sanghyun Woo, Dahun Kim, Donghyeon Cho, and In-So Kweon. Linknet: Relational embedding for scene graph. pages 560–570, 2018.

[34] Dong Wu, Man-Wen Liao, Wei-Tian Zhang, Xing-Gang Wang, Xiang Bai, Wen-Qing Cheng, and Wen-Yu Liu. Yolop: You only look once for panoptic driving perception. *Machine Intelligence Research (MIR)*, 19(6):550–562, 2022.

[35] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5419, 2017.

[36] Shenghua Xu, Xinyue Cai, Bin Zhao, Li Zhang, Hang Xu, Yanwei Fu, and Xiangyang Xue. Rclane: Relay chain prediction for lane detection. In *European Conference on Computer Vision (ECCV)*, pages 461–477, 2022.

[37] Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph r-cnn for scene graph generation. In *European Conference on Computer Vision (ECCV)*, pages 670–685, 2018.

[38] Yi Yang, Hengliang Luo, Huarong Xu, and Fuchao Wu. Towards real-time traffic sign detection and classification. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 17(7):2022–2031, 2015.

[39] Rowan Zellers, Mark Yatskar, Sam Thomson, and Yejin Choi. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5831–5840, 2018.

[40] Ji Zhang, Mohamed Elhoseiny, Scott Cohen, Walter Chang, and Ahmed Elgammal. Relationship proposal networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5678–5686, 2017.

[41] Jianming Zhang, Manting Huang, Xiaokang Jin, and Xudong Li. A real-time chinese traffic sign detection algorithm based on modified yolov2. *Algorithms*, 10(4):127, 2017.

[42] Yuxiao Zhang, Haiqiang Chen, Yiran He, Mao Ye, Xi Cai, and Dan Zhang. Road segmentation for all-day outdoor robot navigation. *Neurocomputing*, 314:316–325, 2018.

[43] Tu Zheng, Yifei Huang, Yang Liu, Wenjian Tang, Zheng Yang, Deng Cai, and Xiaofei He. Clrnet: Cross layer refinement network for lane detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 898–907, 2022.

[44] Yue Zhou, Xue Yang, Gefan Zhang, Jiabao Wang, Yanyi Liu, Liping Hou, Xue Jiang, Xingzhao Liu, Junchi Yan, Chengqi Lyu, Wenwei Zhang, and Kai Chen. Mmrotate: A rotated object detection benchmark using pytorch. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7331–7334, 2022.

[45] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2110–2118, 2016.