Computational Visual Media
https://doi.org/10.1007/s41095-018-0116-x Vol. 4, No. 3, September 2018, 253–266
Research Article

Traffic signal detection and classification in street views using an attention model

Yifan Lu[1], Jiaming Lu[1] (✉), Songhai Zhang[1], and Peter Hall[2]

Abstract Detecting small objects is a challenging task. We focus on a special case: the detection and classification of traffic signals in street views. We present a novel framework that utilizes a visual attention model to make detection more efficient, without loss of accuracy, and which generalizes. The attention model is designed to generate a small set of candidate regions at a suitable scale so that small targets can be better located and classified. In order to evaluate our method in the context of traffic signal detection, we have built a traffic light benchmark with over 15,000 traffic light instances, based on Tencent street view panoramas. We have tested our method both on the dataset we have built and the Tsinghua–Tencent 100K (TT100K) traffic sign benchmark. Experiments show that our method has superior detection performance and is quicker than the general faster RCNN object detection framework on both datasets. It is competitive with state-of-the-art specialist traffic sign detectors on TT100K, but is an order of magnitude faster. To show generality, we tested it on the LISA dataset without tuning, and obtained an average precision in excess of 90%.

Keywords traffic light detection; traffic light benchmark; small object detection; CNN

1 Introduction

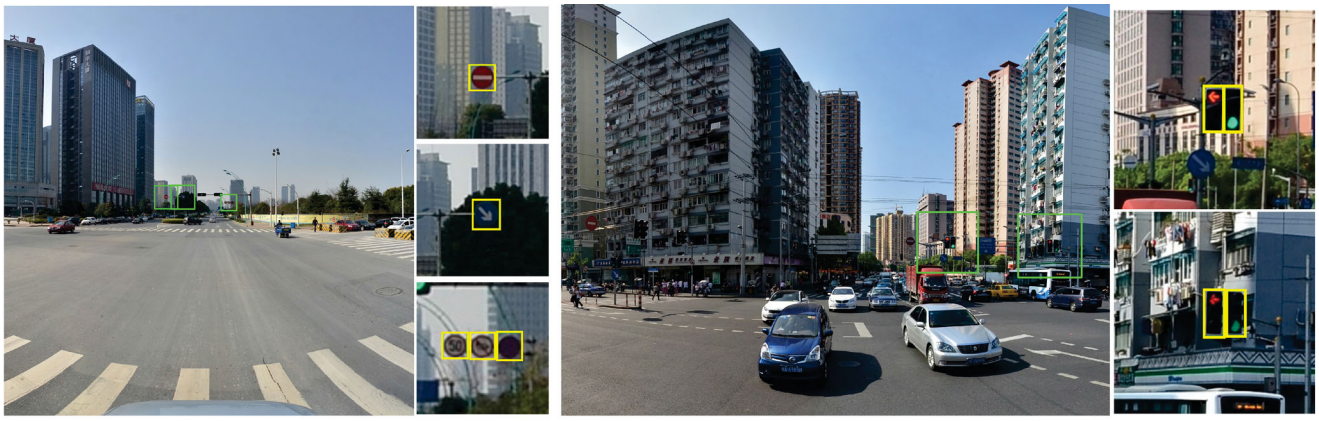
Object detection and classification are important tasks in computer vision. The task is especially challenging when target objects are relatively small and are surrounded by a high degree of background clutter, as is the case for traffic signals in street views. For example, images of a street can be captured at high resolution (e.g., 2048 × 2048 pixels), but vital information, such as traffic signs and traffic lights, are often contained in very small regions (e.g., 50 × 50). With recent developments in autonomous driving, advanced driver assistance systems, and intelligent vehicles, visual content captured by vehicle mounted equipment plays an increasingly important role in perception of the environment and navigation guidance, as you can see in Fig. 1.

Traffic sign and traffic light detection and classification have attracted much study. Any algorithm for road sign detection must be reliable, fast, and general, and should produce results early. It must be reliable so that signs are robustly detected, fast so that other decisions such as sign recognition can take place, general to account for differences in signs between countries, and early so that signs are determined sufficiently distant from the vehicle to allow a safe response time.

Most previous works utilize color, texture, and geometric features as input to machine learning methods such as SVM and tree classifiers to distinguish either targets from background, or different classes of targets. As convolutional neural networks (CNNs) have been found to have superior performance for object classification and detection, they are extensively used in this area. CNN-based methods (e.g., faster RCNN [1] and SSD [2]) with state-of-the-art performance on PASCAL VOC and ImageNet ILSVRC datasets focus on large scale objects in images, whereas traffic signs should be detected early, i.e., at small scale.

CNNs have been built specifically for traffic sign recognition and detection [3]. Jin et al. [4] achieved a recognition rate of 99.65% on the German Traffic

1 TNList, Tsinghua University, Beijing 100084, China. E-mail: Y. Lu, luyifanfrank@foxmail.com; J. Lu, loyaveforever@gmail.com (✉); S. Zhang, shz@tsinghua.edu.cn.
2 Department of Computer Science, University of Bath, United Kingdom. E-mail: P.M.Hall@bath.ac.uk.
Manuscript received: 2018-03-09; accepted: 2018-04-07



(a) Traffic signs in a street view (b) Traffic lights in a street view

Fig. 1 Examples of traffic signs and traffic lights in street views. Traffic signs and traffic lights are detected separately: (a) shows traffic signs only while (b) shows traffic lights only. Green rectangles on the left of each subfigure indicate attention regions; rectangles on the right are corresponding cropped attention regions and bounding boxes of targets within them.

Sign Recognition Benchmark with CNNs. Zhu et al. [5] published a more practical traffic sign dataset, the Tsinghua–Tencent 100K (TT100K) dataset, and proposed a CNN-based method that performs better than fast RCNN [6]. Although the method proposed by Zhu et al. achieves high recall and accuracy, it uses a CNN to scan the whole high resolution image at different scales, which is time-consuming. We propose a more efficient system that utilizes a visual attention model to reduce computation in background areas. We show results of testing it on traffic sign and traffic light detection and classification tasks.

Our CNN approach exhibits all four of the properties needed for traffic signal detection: it is reliable, fast, able to detect signals at a wide variety of scales, and generalises to new datasets obtained in a country different to the training set. Its key technical contribution is use of a visual attention model. Studies in neuroscience [7] suggest that instead of forming a coherent, richly detailed representation of all the objects in the scene, the human visual system tends to focus attention on one object at a time. The perception that all objects are represented in detail simultaneously is a subjective construction enabled by coordinating attention in a few areas deemed to be salient. In this way, perception requires far less processing and memory resources. Similarly, when detecting objects in images, we can design algorithms that focus only on certain regions instead of processing the whole image at high resolution. We use a two-stage framework to accomplish the process. The first stage, based on an attention proposal model, is trained on a low resolution version of the scene to propose attention regions: those regions which should be examined in detail. The second stage, the accurate localization and recognition network, detects and classifies targets within the attention regions, at full resolution. Our framework greatly reduces computational and memory resources since it avoids computing detailed representations of most of the background. Moreover, it avoids processing at multiple scales since the regions proposed by the first stage implicitly contain scale information.

We have evaluated our system for traffic sign and traffic light detection and classification tasks using three datasets: the Tsinghua–Tencent 100K (TT100K) dataset, our own purpose-built dataset, the Tsinghua–Tencent traffic light (TTTL) dataset, which is based on Tencent street views, and the LISA dataset [8] to test generalization. The TTTL dataset contains over 16,000 high resolution images covering various driving scenes. Our system achieved 86.6% mAP (mean average precision) on the dataset, performing better and more reliably than faster RCNN. For the TT100K dataset, an mAP of 87.0% is achieved, which is close to the state-of-the-art method proposed by Zhu et al. [5], but our algorithm is an order of magnitude faster than theirs. It generalizes well, without tuning, to the LISA dataset that was constructed in USA.

Our main contributions are as follows:

- A novel attention-model based two-stage detection framework. The framework is designed to detect small targets in large high resolution images. The

attention model makes our method efficient as it processes images at low resolution and generates a small set of candidate regions. It also makes detection more accurate since the small targets occupy a large portion of the area of attention regions.

- A new street view traffic light benchmark TTTL. To the best of our knowledge, there are no readily available benchmarks for traffic light detection other than the LISA traffic light dataset [8]. We annotated around 10K Tencent street view panorama images and built a dataset with more than 15K traffic light instances. They cover a wide range of street scenes and lighting conditions, and exhibit traffic lights in forms not present in LISA.

The remainder of the paper explains our contribution in greater detail.

2 Related work

2.1 Traffic light and traffic sign detection and classification

Traffic signals such as traffic lights and traffic signs provide important information for driving, and many algorithms have been developed to detect and recognize them. Diaz et al. [9] presented an exhaustive survey of current techniques for such purposes.

Traffic signal detection and recognition methods since about 2007 have been based on color segmentation, shape, and texture features in conjunction with SVM classifiers [10]. Illumination conditions have been studied by Jang et al. [11] and De Charette and Nashashibi [12], whose detector was shape based. Slightly later, shape was also used by Cai et al. [13] to recognize arrow traffic lights. More recently color segmentation has been used, e.g., by Sooksatra and Kondo [14]. Ji et al. [15] proposed a visual selective attention model based on color to construct salience maps; traffic lights are then detected using an SVM classier with HOG features. This is similar in spirit but different in practice to our work.

Some prior art utilizes digital maps and GPS information to improve the efficiency and accuracy of detection [16, 17]. However, prior information is not always accessible and is not in principle necessary (humans make no use of such data).

The widespread adoption of convolutional neural networks (CNNs) has seen their application to traffic signal detection. John et al. [17, 18] showed that CNNs are effective as classifiers of traffic lights, but they used traditional methods for saliency map generation. In 2016, Zhu et al. [5] published the Tsinghua–Tencent 100K dataset for traffic sign benchmark, and developed an end-to-end CNN for both detection and classification. Their approach provides excellent reliability and operates at a range of scales, but has to scan high resolution images at different scales, which impedes its efficiency.

2.2 CNN-based detection

In the past few years, a wide variety of CNN-based approaches have been developed for object detection. Sermanet et al. [19] proposed the Overfeat framework by sliding a fully convolutional network over an input image to produce classification and bounding box regression results. Zhu et al. [5] adopted this framework for traffic sign detection and classification.

Girshick et al. [20] combined region proposal algorithms and CNN as a feature extractor to perform the detection task, in an approach known as RCNN. To avoid redundant computations on overlapping region proposals in RCNN, spatial pyramid pooling (SPPNet) [21] and ROI pooling (fast RCNN) [6] have since been developed. The input image is fed forward into convolutional layers only once and ROI pooling extracts fixed length feature vectors from the feature maps. This greatly accelerates training and testing. Faster RCNN, proposed by Ren et al. [1], further improves the framework by replacing the selective search [22] by a region proposal network (RPN), which shares its convolutional layers with the classification and regression network. The RPN generates fewer proposals yet achieves state-of-the-art results on PASCAL VOC 2007, 2012, and MS COCO datasets. Single-shot approaches have also been proposed, such as YOLO [23] and SSD [2]. They leave out region proposal production and ROI pooling, and directly conduct box regression on feature maps. While this makes them even faster, YOLO's accuracy falls below that of fast RCNN and faster RCNN.

All of these CNN-based detection frameworks were designed for large scale objects. Directly applying them to detect extremely small objects in high resolution images gives results that are inefficient, inaccurate, or both.

2.3 Visual attention model

Visual attention models are inspired by studies of attention mechanisms of human visual system [7], and have found their way into CNNs.

Mnih et al. [24] developed a recurrent neural network (RNN) that selectively processes a sequence of regions of an input image at high resolution. The network takes a region of the image (called a glimpse) and its location as input and determines the location of the next region to be processed. The computational expense is independent of image size, since only a few glimpses are taken and they contain many fewer pixels than the original image. The system outperforms CNN on image classification on the MNIST dataset.

Working to recognise house numbers in the SVHN dataset, Ba et al. [25] extended the idea to multiple object recognition by fixing the number of glimpses for each target in the object label sequence, and adding an *end-of-sequence* label. They added a contextual network that takes a low resolution version of the original image as input and provides initial state for the RNN. Their problem is formulated as classification rather than detection, but nonetheless their model is close in design to our attention proposal model for target detection. Their deep recurrent attention model (DRAM) is trained using reinforcement learning to find attention regions implicitly without the localization loss that we use (see Section 3.2).

Huang et al. [26] utilized the recurrent attention model to detect arbitrary oriented text in the wild and achieved state-of-the-art accuracy on ICDAR 2013 and MSRA-TD500. However, their detection pipeline depends on extremal regions to generate initial attention proposals and on CNN classifiers to filter non-text proposals. In contrast, we directly utilize an attention model to propose target regions.

Gidaris et al. [27] proposed the AttractioNet approach, which generates box proposals by iterative attention and refinement processes. Their approach surpasses all other box proposal methods. It differs from our method in that our attention model proposes (wide) context regions rather than (tight) object bounding boxes. In the case of multi-region detection, Gidaris et al. [28] improved results by making use of context information integrated from the immediate area surrounding each box; this points to the importance of context, a lesson echoed in our results.

3 Method

3.1 Concepts

In this section, the framework of our system for small target detection and classification is presented. The system is composed of two parts: the *attention proposal modeller* (APM) and the *accurate locator and recognizer* (ALR). The two parts are designed to accomplish different tasks: the APM proposes attention regions that are likely to contain targets, telling ALR where to look; the ALR then localizes and classifies targets in these attention regions. Both tasks can be formulated as taking raw image pixels as input and performing regression on the coordinates of certain boxes. Since faster RCNN performs impressively at such a task, we adopt its structure as the basis for both parts. The difference is that the APM performs regression on the bounding box of an attention region while the ALR performs regression on the bounding box of a real object. Figure 2 provides an overview of the framework. We next discuss the design of the two parts.

3.2 Attention proposal modeller

The aim of the APM is not to precisely locate the targets, but to provide candidate regions with high confidence at low computational cost. This task depends more on global information about the whole image and less on details of the targets. Thus, the original high resolution image may be down-sampled to lower resolution for this purpose. We formulate the task as follows: given a high resolution image I_{H}, the APM takes the corresponding down-sampled image I_{L} as input, and outputs a set of at most K attention regions $A = A_1, A_2, \cdots, A_K$, as well as their corresponding confidences $\theta = \theta_1, \theta_2, \cdots, \theta_K$. These attention regions are cropped from I_{H} for use as input to the ALR which accurately locates and classifies targets within them.

Our approach to producing attention regions is based on faster RCNN, which solves the following problem: given an input image, output a set of region proposals with their locations (x, y) and size (w, h). Each region proposed has an associated value that measures the confidence that the region contains an object of interest.

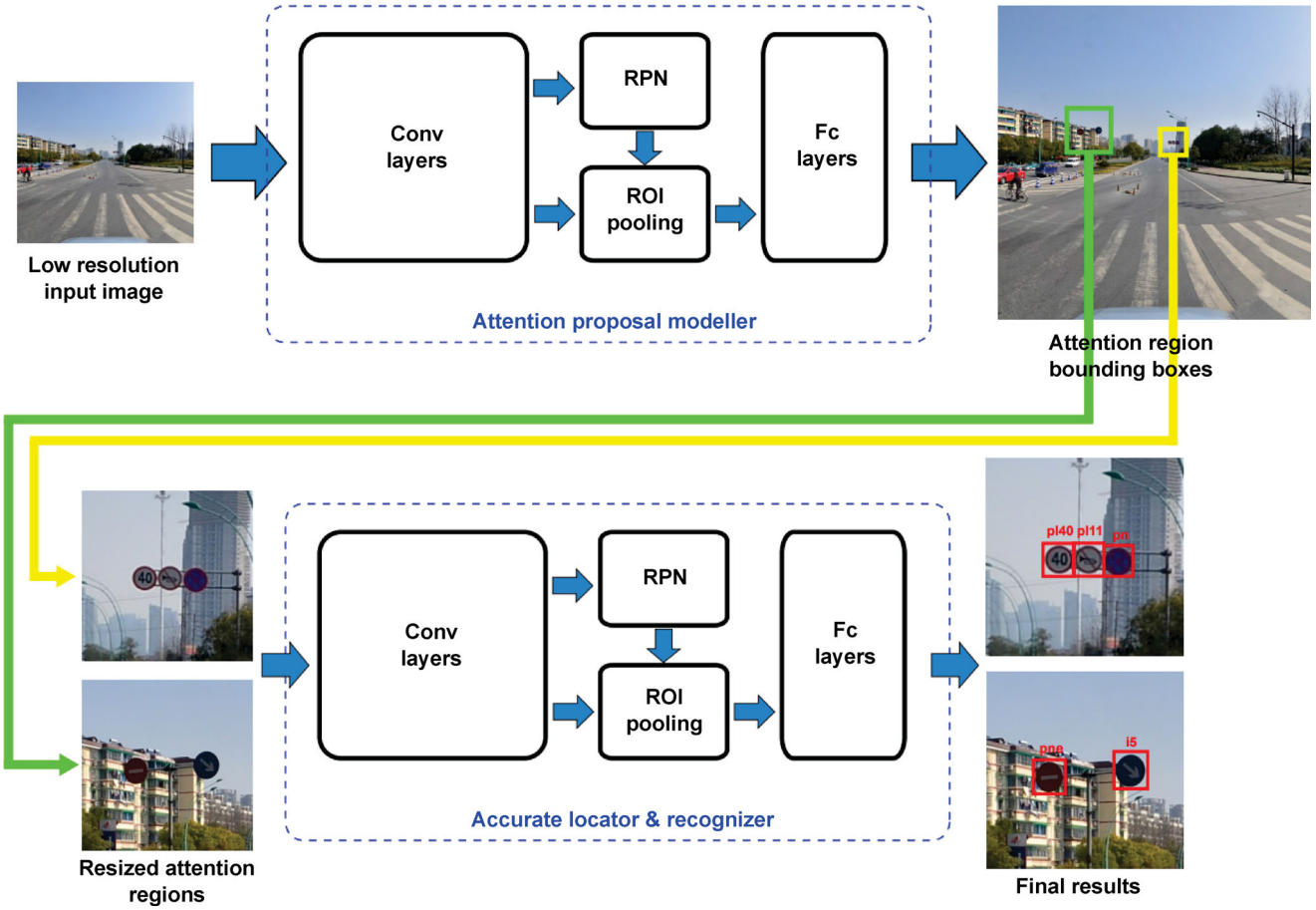


Fig. 2 System overview. Our attention proposal model has a similar architecture to faster RCNN but outputs bounding boxes of attention regions. The accurate locator and recognizer also uses a faster-RCNN-like model. It takes as input cropped and resized attention regions generated by the attention proposal model, and predicts bounding boxes and classes of the targets in attention regions.

Following faster RCNN, the APM comprises a region proposal network (RPN) and a fast RCNN. They share a convolutional sub-network that outputs a feature map of spatial size $W \times H$. The RPN generates region proposals based on anchor boxes at each position of the feature map. High confidence proposals are then processed by fast RCNN through ROI pooling and fully connected layers. Both RPN and fast RCNN output box regression results and confidence scores; they are trained with ground truth boxes. In our case, we only have ground truth bounding boxes for traffic signals, but we can define the bounding boxes for attention regions. The attention region should enclose the traffic signal and its size should be proportional to the object size (for the reason stated below): see Fig. 3. Thus, the attention box (x^*, y^*, w^*, h^*) is defined as follows:

$$x^* = x_0, \quad y^* = y_0, \quad w^* = h^* = \alpha \max(w_0, h_0)$$

where (x_0, y_0, w_0, h_0) is the bounding box of a traffic light or sign. The scale α is set so that traffic lights or signs are contained within the proposal, but not so large as to slow a detailed search for the object at the next stage; we find $\alpha = 5$ to be a suitable choice.

Now that we have the ground truth boxes of attention regions, we parametrize the coordinates of boxes as in Ref. [1]. For the RPN, the parametrized coordinates are calculated using:

$$t_x^* = (x^* - x_a)/w_a, \qquad t_y^* = (y^* - y_a)/h_a$$

$$t_w^* = \log(w^*/w_a), \qquad t_h^* = \log(h^*/h_a)$$

$$t_x = (x - x_a)/w_a, \qquad t_y = (y - y_a)/h_a$$

$$t_w = \log(w/w_a), \qquad t_h = \log(h/h_a)$$

where $(t_x^*, t_y^*, t_w^*, t_h^*)$ are coordinates of the ground truth and (t_x, t_y, t_w, t_h) are the RPN prediction, (x_a, y_a, w_a, h_a) is the anchor box and (x, y, w, h) is the predicted box. Smooth L_1 loss [6] is used for

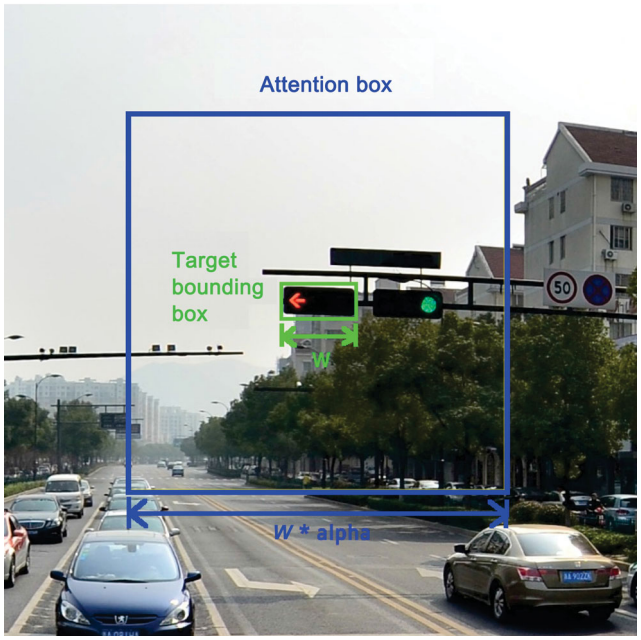


Fig. 3 The attention box is a square region enclosing the target bounding box, with side length α times that of the longer side of the bounding box.

regression in the RPN, which seeks to minimise the function:

$$L_{\text{loc}}^{\text{RPN}} = \Sigma_{i\in x,y,w,h}\text{Smooth}_{L_1}(t_i - t_i^*)$$

where

$$\text{Smooth}L_1(x) = \begin{cases} |x| - 0.5, & |x| \geqslant 1 \\ 0.5x^2, & |x| < 1 \end{cases} \quad (1)$$

Here, $L_{\text{loc}}^{\text{RPN}}$ is the regression loss of RPN. The classification loss is a cross-entropy loss of softmax output: $L_{\text{loc}}^{\text{RPN}} = (1/N)\Sigma_{i=1}^{N}(\log(p_{n,l_n}))$. N is the number of samples and p_{n,l_n} is the predicted softmax probability of the nth sample belonging to the ground truth class l_n. There are two classes in the APM, one for the attention region and one for the background. The ground truth is based on the anchor box matching strategy proposed in Ref. [1]. For the fast RCNN sub-network, the ground truth is also parametrized, but with respect to proposals from the RPN rather than anchor boxes:

$$u_x^* = (x^* - x_p)/w_p, \qquad u_y^* = (y^* - y_p)/h_p$$
$$u_w^* = \log(w^*/w_p), \qquad u_h^* = \log(h^*/h_p)$$
$$u_x = (x_f - x_p)/w_p, \qquad u_y = (y_f - y_p)/h_p$$
$$u_w = \log(w_f/w_p), \qquad u_h = \log(h_f/h_p)$$

where u_i^* and u_i $(i \in x, y, w, h)$ are the parametrized coordinates of the ground truth and the fast RCNN prediction respectively. (x_p, y_p, w_p, h_p) is the box proposed by RPN and (x_f, y_f, w_f, h_f) is the predicted box from fast RCNN. Smooth L_1 loss is also used for regression in fast RCNN:

$$L_{\text{loc}} = \Sigma_{i\in x,y,w,h}\text{Smooth}_{L_1}(u_i - u_i^*)$$

The classification loss L_{cls} is also a 2-class softmax loss, and is defined as $L_{\text{cls}} = -(1/N)\Sigma_{i=1}^{N}(\log(p_{n,l_n}))$.

To sum up, the overall loss function of the APM is

$$L_{\text{APM}} = L_{\text{loc}}^{\text{RPN}} + \lambda_1 L_{\text{cls}}^{\text{RPN}} + \lambda_2 L_{\text{loc}} + \lambda_3 L_{\text{cls}}$$

We set $\lambda_1, \lambda_2, \lambda_3$ to 1, as Ren et al. [1] found training is insensitive to their values over a wide range. The number of proposals generated by the APM determines the computational cost of the second stage, so this number should not be too large while guaranteeing high recall. We apply non-maximum suppression (NMS) and filter the proposals with a confidence threshold T to reduce proposal number. The maximum number of proposals is set to K. If more than K proposals are generated, only the K highest scored proposals are considered by the second stage. We empirically choose $K = 8$. The architecture of the network is shown in Fig. 2 and Table 1. The shareable convolutional layers are similar to the Zeiler and Fergus model [29] and there are 3 fully connected layers after ROI pooling. The RPN parameters such as anchor numbers, NMS threshold, and proposal numbers are set to the same values as in Ref. [1]. It is worth noting that other detection algorithms could also be used as the attention proposal modeller, as long as they can generate attention regions of a similar kind and can produce a reasonably small set of results. With the definition of the attention box, we are able to train a faster RCNN to propose a small set of regions for further examination. This reduces the computational cost in two ways: the APM only needs a low resolution image input for attention proposals, and only a few regions need to be processed at high resolution in the second detection stage.

3.3 Accurate localization and recognition

The APM output is a set of regions, and only those regions need be examined for targets during the second stage. This brings two advantages. Firstly, it saves considerable computation, since only a small part of the original high resolution image, rather than the whole, is taken as input. Secondly, the proportion of object area to the attention region

Table 1 Network structure of the attention proposal model

Layer	Conv1	Conv2	Conv3	Conv4	Conv5	RPN conv	RPN output		Fc6	Fc7	Predictions	
							Scores	Boxes			Scores	Boxes
Channels	64	128	256	256	256	256	18	36	4096	4096	2	8
Kernel size	7	5	3	3	3	3	3	3	—	—	—	—
Stride	2	2	1	1	1	1	1	1	—	—	—	—
Pooling	Max(3,2)	Max(3,2)	—	—	ROI pooling	—	—	—	—	—	—	—

area is much larger than to the original image area, making the localization and recognition task easier. Detection algorithms such as fast RCNN and faster RCNN are usually poor at detecting small objects, but if attention regions are resized to reasonable scale, such algorithms would be suitable for detecting the originally small objects. The ALR takes attention regions proposed by the APM as input and scales them all to the same size, chosen based on performance on the validation set. Since the APM is supposed to predict regions whose sizes are α times as large as the target size, the target sizes in the scaled inputs lie within a small range, further simplifying the task. As we will see in the next section, the target sizes in the rescaled attention regions are concentrated in a narrow range, which helps achieve better performance for originally small targets.

Many detection algorithms can be used as the second stage localizer and recognizer (ALR). We use the faster RCNN framework as it provides state-of-the-art results for many detection tasks. The architecture is the same as for the APM shown in Table 1 except that the number of class score outputs is adjusted to match the number of label classes. For the traffic sign dataset, the model is trained to recognize 45 classes of signs and to predict their bounding boxes, while for the traffic light dataset, lights need to be classified into 6 categories, and the light housing bounding boxes need to be regressed. For all other settings of the framework, we just follow faster RCNN [1].

At testing time, all proposed regions are fed forward in a batch and the output bounding boxes are transformed to their original position in I_{H}. Then NMS is applied to yield the final localization results.

4 Experiment

We performed experiments on detection and classification of two kinds of small targets in street views: traffic signs (see Section 4.1) and traffic lights (see Section 4.2). The experiments on traffic signs used the Tsinghua–Tencent 100K dataset [5], and we compare our method to the method in that paper. The traffic light detection and classification experiments used our TTTL dataset, as well as the LISA traffic light dataset to test generality.

4.1 Traffic sign detection and classification

To make a fair comparison with the method in Ref. [5], we used their training and testing data. There are 45 classes of traffic signs; each class has more than 100 instances. We did not follow their data augmentation protocol, in which they blend traffic sign templates with background street views to generate more data. To diminish the imbalance in number of samples between classes, we oversampled classes with fewer than 1000 instances to ensure that each class had over 1000 samples in each epoch. No other data augmentation was conducted. For attention model training, we set the enlargment ratio α of target bounding boxes to 5. The attention region boxes were not class specific, i.e., there were only two classes, attention region and background, in the attention model.

4.1.1 Training

When training the APM, we resized the original 2048×2048 high resolution images to 480×480 lower resolution images, and trained the network with a single image per batch for 100,000 iterations, with approximately 15 epochs over the training data.

For the ALR, we trained the network on the attention regions generated by the attention model. There were about 47,000 images per epoch and the network was trained for 500,000 iterations with batch size 1. The input images were resized to 360×360.

For both APM and ALR, we used SGD with initial learning rate 10^{-3} and momentum 0.9. The learning rate was set to 5×10^{-4} after 300,000 iterations for ALR. We set the dropout ratio to 0.5 for the fc6 and fc7 layers. Both networks were trained from scratch, after initialization using the method of He

et al. [30]. When testing the system, the input size of both networks was the same as for training, and the maximum number of attention proposals K was set to 8.

4.1.2 Evaluation

We evaluated our method on the Tsinghua–Tencent 100K test dataset. It achieved 87.0% mAP at a Jaccard similarity coefficient of 0.5 and the average recall and precision at highest F1 score were 83.4% and 91.7% respectively. The performance is close to the state-of-the-art method due to Zhu et al. [5], which has an mAP of 87.5%, an average recall of 86.0%, and an accuracy of 88.3%. However, our method is an order faster than their Overfeat-based method, as we avoid scanning the whole high resolution image and detecting at multiple scales; they process input images at scales 0.5, 1, 2, and 4. The original images are of size 2048×2048 so that Zhu et al.'s largest input image has size 8192×8192, which incurs a tremendous computational cost. Our approach takes only 0.3 s to process the same image.

We also evaluated faster RCNN [1] on the dataset as a baseline method. We used ALR alone to detect and classify targets on the original high resolution image. Both its performance and efficiency are lower than those of our method. Table 2 gives a detailed comparison of the three methods. All methods were benchmarked on an nVidia GTX980 GPU.

To demonstrate that the attention model can propose regions at a suitable scale, we examined the statistics of the target sizes in attention regions resized to 360×360, and in the original images. All targets in the TT100K testing set were considered. We used the square root of the area of the target bounding boxes as a measure of target sizes. As shown in Fig. 4(a), in attention regions, more than 80% of the targets were in the size range [32, 96] pixels, while the original size of over 40% of the targets was smaller than 32 pixels. In other words, the targets originally had widely differeing sizes, but in attention regions they were concentrated at medium sizes. This was to be expected since the APM was trained to propose attention region boxes that are α times as large as the target bounding box.

Table 2 Performance of three methods on the TT100K dataset

Method		Faster RCNN	Zhu et al.	Ours
AP	mean	0.684	0.875	0.870
	std	0.137	0.087	0.085
Recall	mean	0.650	0.860	0.834
	std	0.108	0.083	0.075
Precision	mean	0.757	0.883	0.917
	std	0.132	0.122	0.092
Run time	mean	0.33 s	5.83 s	0.26 s
	std	0.02 s	0.07 s	0.03 s

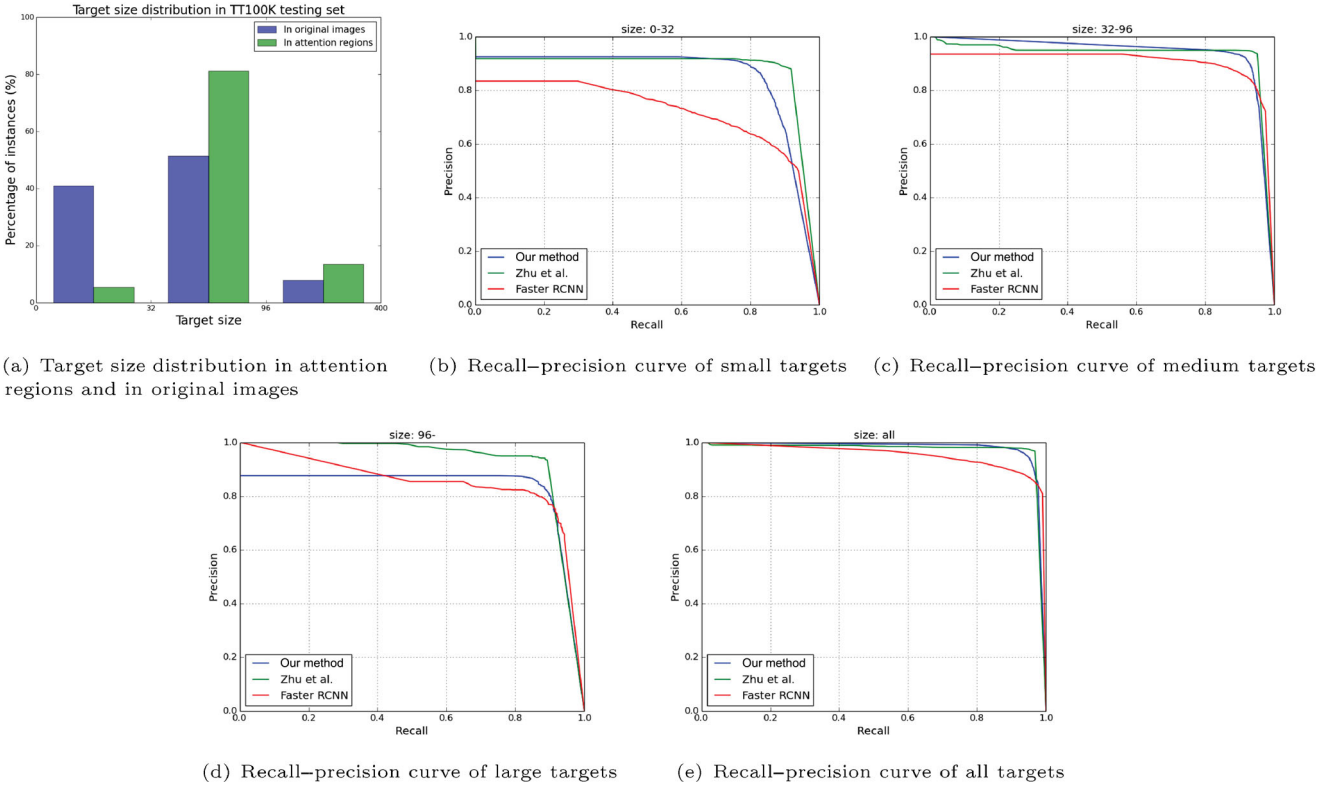
Our APM inherently solves the problem of adjusting scale, making it easier for the ALR to accurately locate and classify targets. Therefore, our method performs just as well on small (area < 322), medium ($322 <$ area < 962), and large (area > 962) targets, as shown in Figs. 4(b)–4(d). In contrast with faster RCNN, which has poor performance on small targets, our method and the Overfeat method proposed by Ref. [5] both have high recall and precision on small targets. Our method can furthermore detect small targets that Overfeat fails to detect, as shown in Fig. 6.

Our method may miss targets in unusual contexts since the APM is unlikely to propose such regions for further detection. Such failures are shown in Fig. 6.

4.2 Traffic light detection and classification

We also tested our method on traffic light detection and classification. Although many methods have been proposed for these tasks, there is no readily available specific dataset with high resolution street view images. We thus built a dataset specifically for traffic light detection and classification, in order to evaluate our method and to provide a benchmark for other studies. We also used the LISA traffic light database [8] to test the generalization capability of our model: the training set, TTTL, and the LISA test set originate in different countries and therefore exhibit differences in traffic lights.

We chose to build this dataset using Tencent street view data, as street views are closer to driving scenarios than photos taken by pedestrians with cameras or cell phones. Furthermore, there is sufficient Tencent street view data to cover diverse scenes and lighting conditions, providing a good test of the method robustness. While the LISA dataset contains continuous frames from video sequences, we picked street views from different places to ensure diversity. The dataset consists of more than 16,000 images; about 8300 of them contain traffic lights. We call it the Tsinghua–Tencent traffic light (TTTL) dataset. We trained our networks on a training set of over 6700 images and evaluated them on the



(a) Target size distribution in attention regions and in original images

(b) Recall–precision curve of small targets

(c) Recall–precision curve of medium targets

(d) Recall–precision curve of large targets

(e) Recall–precision curve of all targets

Fig. 4 The attention model improves small target detection performance for different sizes in the Tsinghua–Tencent 100K testing set. (a) The attention proposal model tends to propose attention regions at reasonable scales so that target sizes in the resized regions are concentrated in the range [32, 96] while in the original images they are more widely distributed. (b)–(e) Recall–precision curves for three methods on targets of different sizes. Our method outperforms faster RCNN and is competitive with that of Zhu et al.

testing set of about 1600 images. We also tested the trained model on 6 clips from the LISA dataset. All experiments used an nVidia GTX 980 GPU.

4.2.1 Tsinghua–Tencent traffic light dataset

The Tencent street view images were captured by vehicle or shoulder mounted cameras, and post-processed to form 8192 × 4096 pixel high resolution panoramas. Since the upper and bottom parts of the panoramas are mainly sky and ground, the images are cropped to between 25% and 62.5% of their height, and then split into 4 pieces horizontally. This yielded 16,313 images of size 2048×1536 pixels. Those images were annotated with bounding boxes of the traffic light surrounds, bounding boxes of the lit bulbs, and the kinds of lights. We have 15 classes of lights, including an *other* class and an *unrecognisable* class. Some classes have very few instances, so we just considered 6 major classes: *green*, *red*, *red left turn*, *green forward*, *red pedestrian*, and *other*. Ignoring those images without traffic lights, we randomly split the dataset into training and testing set in the ratio of 4:1, yielding 6709 training images and 1656 testing images. Table 3 gives the number of instances and example images for each of the 6 classes.

4.2.2 Training

As for the traffic sign task, we resized the original 2048 × 1536 images to 480 × 360 lower resolution images as inputs to the APM. The network was trained for 75,000 iterations, and about 11 epochs, with batch size 1. The trained APM was used to generate about 13,000 attention region images over the training data. The ALR was then trained with those images for 500,000 iterations with a single image per batch. The attention region images were also resized to 360 × 360 as input. As in the traffic sign task, SGD was used and the learning rate scheduler was the same; the dropout ratio of fc6 and fc7 was

Table 3 Number of instances for each major class in Tsinghua–Tencent traffic light dataset

Class	Red	Green	Red left turn	Green forward	Red pedestrian	Other
Instance number	4558	3873	2748	908	1549	2539

0.5. Both networks were initialized with the method proposed by He et al. [30]. When testing the system, the input size of both networks was the same as for training, and the maximum number of attention proposals K was set to 8.

4.2.3 Evaluation

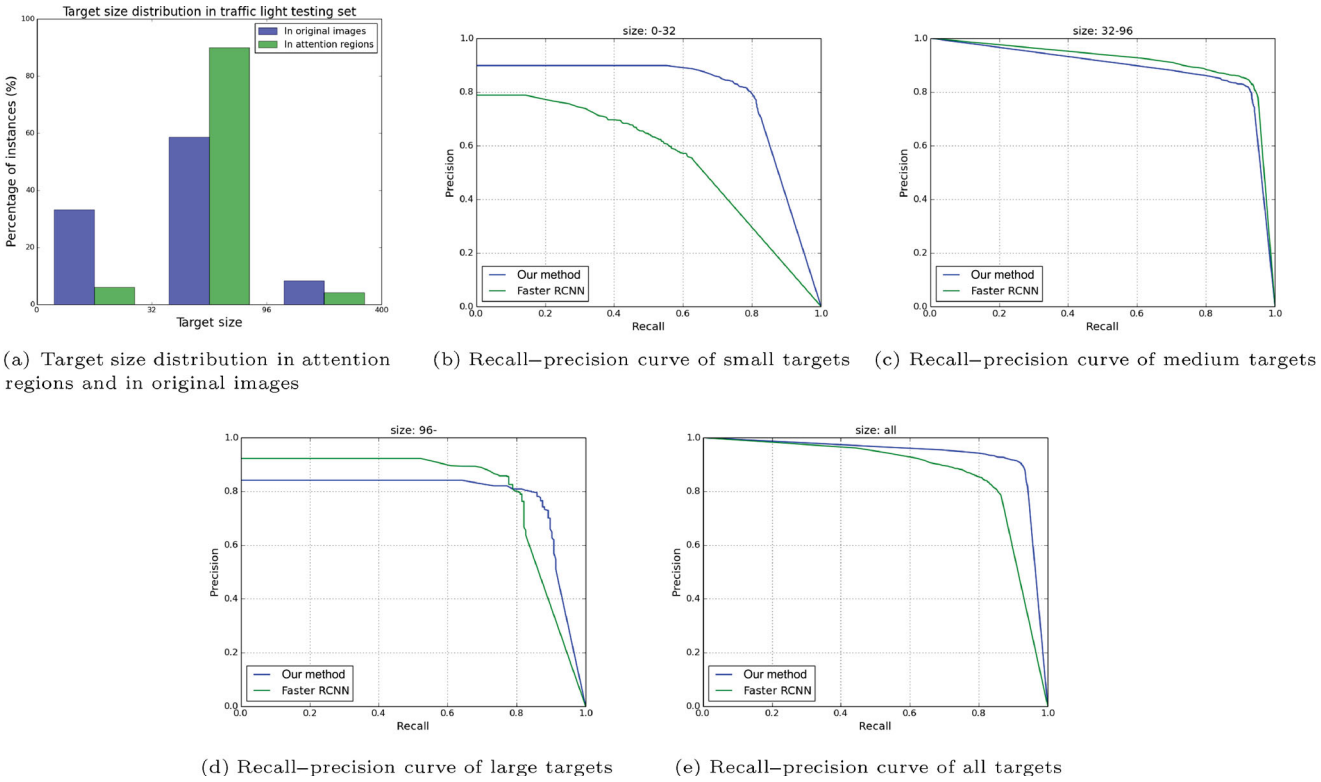
We tested our method on the TTTL testing set, achieving an mAP of 86.2%, an average recall of 83.6%, and an average precision of 84.7%, without considering the *other* class. As shown in Table 4, our method performs better and runs faster than the baseline faster RCNN. The performance on targets of different sizes are shown in Fig. 5. Similarly to the results found for traffic signs, although over 33% of targets original sizes are smaller than 32 pixels, in the resized attention regions proposed by APM, nearly 90% of them are concentrated in the size range [32, 96]. The recall and precision for small, medium, and large targets are close to each other. In comparison to the baseline faster RCNN, the performance for medium and large targets is similar, but our method has much higher recall and precision for small targets, due to the relatively larger proportion of those small targets to the attention regions. Figure 7 shows some examples of our results in various challenging cases. Our method is robust to variations in lighting conditions such as overexposure and underexposure, and different environmental contexts such as underneath bridges and the entrances of tunnels.

Table 4 Performance of our method and faster RCNN on the Tsinghua–Tencent traffic light dataset

Method		Faster RCNN	Our method
AP	mean	0.760	0.866
	std	0.056	0.025
Recall	mean	0.712	0.836
	std	0.049	0.011
Precision	mean	0.744	0.847
	std	0.052	0.061
Run time	mean	0.28 s	0.20 s
	std	0.01 s	0.04 s

4.2.4 Generalization

To evaluate the generalization capability of our method trained on the TTTL dataset, we tested



(a) Target size distribution in attention regions and in original images

(b) Recall–precision curve of small targets

(c) Recall–precision curve of medium targets

(d) Recall–precision curve of large targets

(e) Recall–precision curve of all targets

Fig. 5 The attention model improves small target detection performance on the Tsinghua–Tencent traffic light testing set. (a) As for traffic sign results, target sizes in the resized attention regions are more concentrated around medium sizes than those in the original images. (b)–(e) show the recall–precision curves of our method and faster RCNN for different target sizes; our method performs much better than faster RCNN on small targets.

Fig. 6 Example results for challenging cases in the Tsinghua–Tencent 100K dataset. Above: our results. Below: Overfeat results. The bottom right of each image shows a close-up of the region of interest. In the first two cases, our method detected small targets that Overfeat missed. In the last two cases, our method missed some targets, as the APM failed to propose the those regions.

Fig. 7 Results for some challenging cases in the Tsinghua–Tencent traffic light dataset. Above: images. Below: close-ups. Our method is robust to different lighting conditions, e.g., extremely dark regions under a bridge and bright regions under strong sun light.

it directly on the LISA dataset. As annotations of the LISA test set are unavailable, we tested it on the training set. There are 13 clips of video taken during daytime; we took 6 of them for evaluation. There are 6 traffic light classes: *go*, *go left*, *stop*, *stop left*, *warning*, *warning left*. These classes are not entirely the same as those in the TTTL dataset, so our model may not properly classify some classes such as *warning left*. Thus, we only evaluated the performance on the classes *go* and *stop*, which correspond to our *green* and *red* classes in TTTL. The original image size in LISA is 1280×960. We resized them to 480×360 images for APM input. The attention proposals were cropped and resized to 360×360 for ALR input. All other settings were the same as when testing on TTTL.

Recall, precision, and AP are shown in Table 5. Our model has high overall recall and precision on the *go* and *stop* classes in the LISA dataset, even though it was not trained on any data in LISA. It

Table 5 Performance of our TTTL-trained model on the LISA traffic light dataset

Clip	Class	Recall	Precision	AP
DayClip1	go	0.959	0.994	0.981
	stop	0.874	0.890	0.945
DayClip2	go	0.908	0.898	0.925
	stop	0.883	0.931	0.953
DayClip5	go	0.908	0.897	0.915
	stop	0.909	0.398	0.410
DayClip7	go	0.972	0.897	0.945
	stop	0.810	0.798	0.78
DayClip8	go	0.857	0.605	0.706
	stop	0.738	0.734	0.735
DayClip9	go	0.998	0.995	0.996
	stop	0.910	0.918	0.907
Average	go	0.934	0.881	0.911
	stop	0.854	0.778	0.788

demonstrates that our model has good generalization capability. While the data from TTTL are all street views in China, the videos in LISA are all captured from US roads and have different lighting and weather conditions. These results show the robustness of our model with respect to varying scenes and natural conditions. We note that for DayClip5, the precision of *stop* is low. This may be because our model classifies *stop left* lights as *stop* lights. In video sequences, there are many very similar frames, so any mistakes made by the method are repeated many times in DayClip5. Similarly, in DayClip8 the mistake that the model confuses *go* with go left is repeated. Also, there are very small traffic lights that are not annotated, but are detected by our model, explaining the relatively low precision. Hopefully, the classification performance could be improved by fine tuning our model on the LISA dataset.

5 Conclusions

In this paper, we have presented an attention model based detection framework to tackle the problem of detecting small objects in large high resolution images. We applied our method to traffic signal detection in street view images. As a complement to the TT100K benchmark, we have built the Tsinghua–Tencent traffic light dataset for training and testing. Our framework outperforms the baseline faster RCNN on both datasets, especially when detecting small targets with area less than 322 pixels. Furthermore, our system runs an order of magnitude faster than the state-of-the-art on TT100K, while having similar recall and precision. Experiments show that the attention proposal model can generate a small set of candidate regions whose area as a proportion of target size lies in a narrow range, making the second stage localization and classification more accurate. Our model trained on the TTTL dataset also shows good generalization capability, achieving high recall and precision on the LISA dataset without any training on it.

In future, we hope to improve the recall of our framework by exploring better attention proposal methods. As our framework is intended for detection in still images, we would like to develop a method for video sequences that utilizes previous detection results to further reduce computational cost. We are also planning to apply our method to other problems such as detecting small targets in remote sensing images. There, the ratio between target size and image size can be even smaller, so it is more challenging to accurately locate targets with relatively low computational cost. Generalization is also interesting: all of the datasets we used are from countries that drive on the right, and there is no database we know of from countries that drive on the left.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61772298), Research Grant of Beijing Higher Institution Engineering Research Center, and the Tsinghua–Tencent Joint Laboratory for Internet Innovation Technology.

References

[1] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Proceedings of the Advances in Neural Information Processing Systems, 91–99, 2015.

[2] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. SSD: Single shot multibox detector. In: *Computer Vision–ECCV 2016. Lecture Notes in Computer Science, Vol. 9905.* Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 21–37, 2016.

[3] Chen, C.; Liu, M.-Y.; Tuzel, O.; Xiao, J. R-CNN for small object detection. In: *Computer Vision–ACCV*

2016. Lecture Notes in Computer Science, Vol. 10115. Lai, S. H.; Lepetit, V.; Nishino, K.; Sato, Y. Eds. Springer Cham, 214–230, 2016.

[4] Jin, J.; Fu, K.; Zhang, C. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems* Vol. 15, No. 5, 1991–2000, 2014.

[5] Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2110–2118, 2016.

[6] Girshick, R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 1440–1448, 2015.

[7] Rensink, R. A. The dynamic representation of scenes. *Visual Cognition* Vol. 7, Nos. 1–3, 17–42, 2000.

[8] Jensen, M. B.; Philipsen, M. P.; Møgelmose, A.; Moeslund, T. B.; Trivedi., M. M. Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems* Vol. 17, No. 7, 1800–1815, 2016.

[9] Diaz, M.; Cerri, P.; Pirlo, G.; Ferrer, M. A.; Impedovo, D. A survey on traffic light detection. In: *New Trends in Image Analysis and Processing–ICIAP 2015 Workshops. Lecture Notes in Computer Science, Vol. 9281.* Murino, V.; Puppo, E.; Sona, D.; Cristani, M.; Sansone, C. Eds. Springer Cham, 201–208, 2015.

[10] Maldonado-Bascon, S.; Lafuente-Arroyo, S.; Gil-Jimenez, P.; Gomez-Moreno, H.; Lopez-Ferreras, F. Road-sign detection and recognition based on support vector machines. *IEEE Transactions on Intelligent Transportation Systems* Vol. 8, No. 2, 264–278, 2007.

[11] Jang, C.; Kim, C.; Kim, D.; Lee, M.; Sunwoo, M. Multiple exposure images based traffic light recognition. In: Proceedings of the IEEE Intelligent Vehicles Symposium, 1313–1318, 2014.

[12] De Charette, R.; Nashashibi, F. Real time visual traffic lights recognition based on spot light detection and adaptive traffic lights templates. In: Proceedings of the IEEE Intelligent Vehicles Symposium, 358–363, 2009.

[13] Cai, Z.; Gu, M.; Li, Y. Real-time arrow traffic light recognition system for intelligent vehicle. In: Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition, 1, 2012.

[14] Sooksatra, S.; Kondo, T. Red traffic light detection using fast radial symmetry transform. In: Proceedings of the 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 1–6, 2014.

[15] Ji, Y.; Yang, M.; Lu, Z.; Wang, C. Integrating visual selective attention model with HOG features for traffic light detection and recognition. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 280–285, 2015.

[16] Fairfield, N.; Urmson, C. Traffic light mapping and detection. In: Proceedings of the IEEE International Conference on Robotics and Automation, 5421–5426, 2011.

[17] John, V.; Yoneda, K.; Qi, B.; Liu, Z.; Mita, S. Traffic light recognition in varying illumination using deep learning and saliency map. In: Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems, 2286–2291, 2014.

[18] John, V.; Yoneda, K.; Liu, Z.; Mita, S. Saliency map generation by the convolutional neural network for real-time traffic light detection using template matching. *IEEE Transactions on Computational Imaging* Vol. 1, No. 3, 159–173, 2015.

[19] Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint* arXiv:1312.6229, 2013.

[20] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580–587, 2014.

[21] He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *Computer Vision–ECCV 2014. Lecture Notes in Computer Science, Vol. 8691.* Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 346–361, 2014.

[22] Uijlings, J. R.; van de Sande, K. E. A.; Gevers, T.; Smeulders, A. W. Selective search for object recognition. *International Journal of Computer Vision* Vol. 104, No. 2, 154–171, 2013.

[23] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 779–788, 2016.

[24] Mnih, V.; Heess, N.; Graves, A.; kavukcuoglu, k. Recurrent models of visual attention. In: Proceedings of the Advances in Neural Information Processing Systems, 2204–2212, 2014.

[25] Ba, J.; Mnih, V.; Kavukcuoglu, K. Multiple object recognition with visual attention. *arXiv preprint* arXiv:1412.7755, 2014.

[26] Huang, W.; He, D.; Yang, X.; Zhou, Z.; Kifer, D.; Giles, C. L. Detecting arbitrary oriented text in the wild with a visual attention model. In: Proceedings of the ACM on Multimedia Conference, 551–555, 2016.

[27] Gidaris, S.; Komodakis, N. Attend refine repeat: Active box proposal generation via in-out localization. *arXiv preprint* arXiv:1606.04446, 2016.

[28] Gidaris, S.; Komodakis, N. Object detection via a multi-region and semantic segmentation-aware CNN model. In: Proceedings of the IEEE International Conference on Computer Vision, 1134–1142, 2015.

[29] Zeiler, M. D.; Fergus, R. Visualizing and understanding convolutional networks. In: *Computer Vision–ECCV 2014. Lecture Notes in Computer Science, Vol. 8689.* Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 818–833, 2014.

[30] He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE International Conference on Computer Vision, 1026–1034, 2015.

Yifan Lu is currently a master student of the Graphics and Geometric Computing Group in the Department of Computer Science and Technology, Tsinghua University. He received his B.S. degree in biology from Wuhan University in 2013. His main research interests include computer vision and deep learning.

Jiaming Lu is a Ph.D. student in the Department of Computer Science and Technology, Tsinghua University. His research interests are in computer vision and computer graphics.

Songhai Zhang received his Ph.D. degree from Tsinghua University, China, in 2007. He is currently an associate professor in the Department of Computer Science and Technology, Tsinghua University. His research interests include image and video processing, and geometric computing.

Peter Hall is leader of the Visual Computing Research Group in the Department of Computer Science at the University of Bath. He is the director of the Centre for Digital Entertainment doctoral training centre. His total grant income totals over $15 million. He regularly publishes in tier one conferences and leading journals. He is on the Editorial Boards of *Computer Graphics Forum* and *Computational Visual Media.*