Coarse-to-fine attention neural network for vehicle light signal detection

Shenao Zhang[1] , **Xin He**[1] , **Lei Kuang**[2] , **Delu Zeng**[1] and **Bo Wu**[2]
[1]South China University of Technology, China
[2]Columbia University, USA
eeshenaozhang@mail.scut.edu.cn, msxinhe@mail.scut.edu.cn, lk2807@columbia.edu
dlzeng@scut.edu.cn, bo.wu@columbia.edu

Abstract

Attention mechanism has proven useful to boost the discriminative power to obtain expressive models. In this paper, we propose a coarse-to-fine attention mechanism for environment-interfering vehicle rear-light signal detection tasks to ease the procedure of generating proposals and dynamically localize the precise regions even without high-quality proposals. We provide proof of the effectiveness of our approach and develop it as an independent module that can be used to be an effective add-on to the network while keeping end-to-end. We then validate our conclusion by exquisitely designing experiments on the CIFAR-10 dataset. We also publish our Vehicle Light Signal (VLS) dataset, which is a challenging detection dataset due to the similar pattern features of the rear part of cars as well as the confusion of light signals of different cars and the interference of the lights in the surrounding. Comprehensive ablation studies and empirical comparisons with state-of-the-art models show that our model outperforms the counterparts in various scenarios.

1 Introduction

Autonomous vehicles need HD maps, sensors, cameras to collect information and make decisions. Understanding complex driving behaviors scenarios is both important and challenging with computer vision techniques given raw data of images or videos collected in real driving situations using cameras. BDD100K [Xu *et al.*, 2017] and Cityscapes [Cordts *et al.*, 2016] are great attempts for perception of road objects, drivable areas with manual annotations of bounding boxes, lane markings, and full-frame instance segmentation. In this paper, we provide a new view to better understand the road situations, i.e. vehical light signal perception. Apart from detecting what objects are around us, including other cars, people and trees, perception of surrounding cars' behaviours are also important. For example, if one car knows the front car is braking or changing lanes as its tail lights show, it will take actions to deceleration or acceleration, respectively, rather than computing the distance between them and wait until the distance is smaller or larger than the predefined thresholds. With this motivation, we publish our Vehicle Light Signal (VLS) dataset, which contains bounding boxes of surrounding cars and labels of their behavior, according to their tail light signals, collected from various car DVRs in various scenarios, as the complex real situations cars might encounter. We will publish our VLS dataset and codes at *https://github.com/scutDACIM/CFA*.

Our dataset is a challenge one for the following reasons. Firstly, lights of the environment or lights of different vehicles might confuse traditional detectors. Secondly, various cars with occlusions will cause low recall rate and precision rate. Thirdly, with different ambient lighting conditions, the intensity of vehicle tail lights will vary. For low-light scenarios, the localization of surrounding cars is difficult, with only salient lights in view. For strong-light scenarios, the vehicle tail lights are difficult to observe. Details and demonstrations can be seen at section 4. For these challenging problems, we propose our fine-to-coarse attention (CFA) mechanism that can be inserted into traditional two-stage object detectors for boosting the performance. With CFA, even if we didn't obtain high-quality proposals, e.g. proposals generated by RPN in Faster RCNN [Ren *et al.*, 2015], we can still gain excellent performance by first localizing coarse patterns by feature maps of different channels, and then localizing precise subparts agreed by experts. We provide our full approach and proof at 3.

The contributions of our work can be divided into three folds: **(I)** We publish our Vehicle Light Signal (VLS) dataset for perception of the behaviours of surrounding vehicles with their light signals. **(II)** We provide proof and conduct experiments on CIFAR-10 of our coarse attention mechanism to validate its efficiency on classification tasks. **(III)** We propose a coarse-to-fine attention module that can insert directly into Faster RCNN or other two-stage detectors for our vehicle light signal detection task.

2 Related work

2.1 Attention mechanism

Deep neural network models are highly benefited from attention mechanism, which is an effective way for capacitating the ability of dynamically extracting informative features. Squeeze-and-excitation block [Hu *et al.*, 2018] is efficient by

assigning each channel of feature map a weight, and can be inserted directly into any deep neural network, optimized in an end to end manner. However, except through backpropagating loss to the block, we cannot know the exact metric SE block uses for deciding the importance weights. In our work, we propose a novel Coarse Attention (CA) mechanism with detailed proof, straightforward intuition and convincing experimental results. Spatial transformer networks (STN) [Jaderberg *et al.*, 2015] is designed to learn transformation parameters in the spatial domain. We also provide a Fine Attention (FA) module to localize the precise discriminative region in a much simpler way. Other kinds of attention mechanisms are also widely adopted for better extracting features in both computer vision and NLP fields. [Vaswani *et al.*, 2017] shed light on attention mechanism to solve the problem of lacking long-range relationships in machine translation problems. [Bello *et al.*, 2019; Li *et al.*, 2019; Khandelwal and Sigal, 2019] also provide evidence that attention can facilitate different tasks.

2.2 Object detection

Popular object detection tasks mainly include detecting general objects, like COCO [Lin *et al.*, 2014] and PASCAL VOC [Everingham *et al.*, 2010] challenges. Traditional detection approaches can be classified into two categories, i.e. two-stage detector and one-stage detector. Two-stage detectors adopt a intuitive idea to first roughly localize where objects we are interested are, i.e. proposals. Then a simple classifier is followed to regress the final object category and location. [Girshick *et al.*, 2014] adopts selective search to generate proposals, and is the first work combining region proposals with CNNs. However, R-CNN is slow due to that it is a multi-stage pipeline and needs to learn within every object proposal. [Girshick, 2015] accelerates it by sharing computation. [Ren *et al.*, 2015] proposes RPN for efficiently generating high-quality proposals, and achieved excellent performance on many datasets. However, for our dataset, Faster R-CNN performs relatively poorly due to the fine-grained property between classes. Details can be found at experiment part. One-stage detector [Redmon *et al.*, 2016; Lin *et al.*, 2017] can do quick training and inference by combining processes of generating proposals and regression together, but will cause precision loss. Recent works focusing on efficient anchor localizing [Wang *et al.*, 2019] are efficient for obtaining region anchors, rather than via a predefined manner as before. Research based on neural architecture search (NAS) [Xu *et al.*, 2019] is to search for an optimal network architecture expecting for highest mAP or trade-off between performance and speed. Our VLS dataset can also be regarded as an object detection dataset, but with more challenges. More specifically, the tail lights of cars will easily be wrongly detected with the light of other vehicles and the street lights, especially at the low-light or crowded scenarios. So previous methods will perform poorly at the vehicle light signal perception task. More dataset details and experiment results can be found at section 4.

3 Approach

For popular two stage object detector, thousands of proposals are generated for bounding box regression and object category prediction. Take Faster R-CNN [Ren *et al.*, 2015] for example, Region Proposal Network (RPN) is introduced after extracting features with backbone networks to simultaneously predict rough bounds of objects and objectness scores at each position. Then two fully connected (FC) layers are followed to classify the exact category of object in positive proposals and regress to generate the final bounding box. However, as discussed previously, for similar objects or objects that appear near the same place, the detectors will have difficulties to decide which parts in the rough larger proposals are responsible for dominating the following accurate bounding box regression and classification processes, especially in our vehicle light signal dataset. In this section, we will introduce our coarse-to-fine attention scheme, including Coarse Attention and Fine Attention to solve fine-grained patterns and co-occurrence objects problems in our task, where these problems are more urgent to be solved.

3.1 Coarse Attention

Attention is believed to play an important role in computer vision tasks for capturing extra clues in both spatial domain [Jaderberg *et al.*, 2015] and feature domain [Hu *et al.*, 2018]. We propose a Coarse Attention (CA) mechanism to first decide dimensions of feature maps which depict rough regions that may facilitate the whole process, then Fine Attention is followed to precisely generate a spatial attention mask on feature maps benefited from the CA. Then parameters of feature maps with attention can be optimized in an end to end manner. [Zhou *et al.*, 2016] has shown that deep neural network have the ability to localize region of interests despite being trained only on image-level labels. Adopting global average pooling (GAP), discriminative image regions are depicted by generating class activation map (CAM) for classifying particular categories. More specifically, the feature after GAP is:

$$F_d = \sum_x \sum_y f_d(x, y), \tag{1}$$

where d denotes the index of dimension, i.e. channel and x, y are for the size of feature map at this channel. Then FC layer is followed to compute the classification scores:

$$S_{ori} = \sum_c \sum_i \sum_d \mathbb{1}(c = label_i) Softmax_c(W_d^c F_d^i), \tag{2}$$

where S_{ori} is the performance score of the batch, i is the index in batch, d is the index of channel, c is the class number, W_d^c is the weight of FC layer with class c and dimension d, $label_i$ is the true label of sample i. $Softmax_c$ is the prediction score at class c of each sample. The idea behind Eq. 2 is straightforward: if we have high scores for the true classes, we can say the classifier is of good performance.

Now consider a case that we have features of samples after GAP in the whole batch, and calculate the score with the

following formula:

$$S = \sum_c \sum_i \sum_d \mathbb{1}(c = label_i)$$
$$Softmax_c[W_d^c F_d^i(-(\frac{\sum_{i'} F_d^{i'} \mathbb{1}(i' = c)}{\sum_{i'} \mathbb{1}(i' = c)} - F_d^i)^2 + \alpha \sum_{c'} \mathbb{1}(i \neq c')(\frac{\sum_{i'} F_d^{i'} \mathbb{1}(i' = c')}{\sum_{i'} \mathbb{1}(i' = c')} - F_d^i)^2)] \quad (3)$$

Here, α controls the weight of contribution for samples with different labels. Generally, we can set $\alpha = \frac{1}{C-1}$, where C is the total number of classes.

For simplicity, in the following, we will only analyze the binary classification case, i.e. the value of c changes between 0 and 1. Tasks with more classes can be analyzed in a very similar way. And for now, we only analyze our mechanism when the primary network has roughly converged and only analyze the FC layers in the network. We will later introduce our approach that can be flexibly inserted to the network and trained end to end with deep supervision. Denote:

$$Avg_0 = \frac{\sum_{i'} F_d^{i'} \mathbb{1}(i' = 0)}{\sum_{i'} \mathbb{1}(i' = 0)} \quad (4)$$

$$Avg_1 = \frac{\sum_{i'} F_d^{i'} \mathbb{1}(i' = 1)}{\sum_{i'} \mathbb{1}(i' = 1)} \quad (5)$$

Here, Avg_0 and Avg_1 represent the average feature after GAP of label 0 and label 1 respectively. So Eq. 3 can be rewritten as:

$$S = Softmax_{0,1}^0(S_0) + Softmax_{0,1}^1(S_1), \quad (6)$$

$$\begin{aligned} S_0 &= \sum_{i \in Neg} \sum_d W_d^0 \cdot F_d^i \cdot \\ &\quad (-(Avg_0^2 - 2F_d^i Avg_0 + (F_d^i)^2) \\ &\quad + (Avg_1^2 - 2F_d^i Avg_1 + (F_d^i)^2)) \\ &= \sum_{i \in Neg} \sum_d W_d^0 \cdot F_d^i \cdot \\ &\quad (Avg_1^2 - Avg_0^2 + 2(Avg_0 - Avg_1)(F_d^i)) \end{aligned} \quad (7)$$

$$\begin{aligned} S_1 &= \sum_{i \in Pos} \sum_d W_d^1 \cdot F_d^i \cdot \\ &\quad (Avg_0^2 - Avg_1^2 + 2(Avg_1 - Avg_0)(F_d^i)) \end{aligned} \quad (8)$$

Here, Neg and Pos represents negative set and positive, i.e. $label_i = 0$ and $label_i = 1$ respectively. Then we obtain:

$$S_0 = \sum_d W_d^0 \sum_{i \in Neg} F_d^i (b + k \cdot F_d^i) \quad (9)$$

$$S_1 = \sum_d W_d^1 \sum_{i \in Pos} F_d^i (-b - k \cdot F_d^i) \quad (10)$$

Here b denotes $Avg_0^2 - Avg_1^2$, k denotes $2(Avg_1 - Avg_0)$.

$$\begin{aligned} S &= Softmax_{0,1}^0(S_0) + Softmax_{0,1}^1(S_1) \\ &= \sum_d (\sum_{i \in Neg} Softmax_{0,1}^0(W_d^0 F_d^i (b + k \cdot F_d^i)) \\ &\quad + \sum_{i \in Pos} Softmax_{0,1}^1(W_d^1 F_d^i (-b - k \cdot F_d^i))) \end{aligned} \quad (11)$$

The original score function Eq. 2 can be rewritten as:

$$\begin{aligned} S_{ori} &= \sum_d (\sum_{i \in Neg} Softmax_{0,1}^0(W_d^0 F_d^i) \\ &\quad + \sum_{i \in Pos} Softmax_{0,1}^1(W_d^1 F_d^i)) \end{aligned} \quad (12)$$

We then compare these two score functions. For each sample, the contribution it contributes to our score function, i.e. Eq. 11 is:

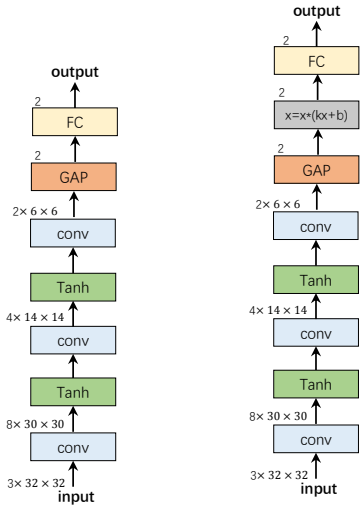
$$\begin{aligned} S_i &= \sum_d Softmax_c(W_d^c F_d^i (b + k \cdot F_d^i)) \\ &= \sum_d Softmax_c(W_d^c F_d^i b + W_d^c (F_d^i)^2 k)), \end{aligned} \quad (13)$$

and the contribution it contributes to the original score function, i.e. Eq. 12 is:

$$S_i' = \sum_d Softmax_c(W_d^c F_d^i) \quad (14)$$

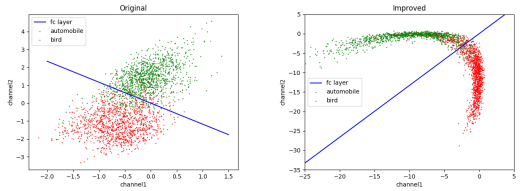
Comparing Eq. 13 and Eq. 14, we can notice that since we analyze when primary network has roughly converged, our approach benefits the training process by not only projecting the feature tensor to a boundary, i.e. the plane decided by the cluster center, which is nearly the normal vector of the optimal hyperplane that splits different samples, but also adapt to scale according to the center position of samples in the feature domain. Both these two properties make our mechanism efficient.

To shed light on this attention mechanism, we now describe how we use it in our method. We first need to get the average feature after GAP in the batch. However, we cannot feed all our data to GPU due to the memory limitations. One more drawback is that we don't have good features that hyperplanes can optimally split at the beginning of our training process. To solve these two problems, we introduce our Coarse Attention mechanism module, as shown in figure 3. We use two FC layers to regress what the average features after GAP would be by providing labels according to samples that have fed in. More specifically, according to Eq. 3, we save the average feature of each class in memory, then generating pseudo-labels which are expected attention scores of each feature map. Then the Euclidean Loss calculated between predicted scores and pseudo-labels is added to the original loss during the training process. In this way, the coarse attention module can be optimized to decide the importance of feature maps which depict rough regions that may facilitate our training. When at inference stage, the pseudo-labels are removed since we already have



(a) Original (b) Ours

Figure 1: Original network architecture and ours for training automobile and bird classes of CIFAR-10.



(a) Original (b) Ours

Figure 2: Features before FC layer

a subnetwork that can dynamically generate importance weights. We can then have an efficient and robust module during the whole process. Intuitively, we can also have the idea why of our attention mechanism works by the fact that the average feature after GAP is likely to cluster within the same class, while can be separated between different classes. We can reinforce this information to the following layers by paying more attention to channels which are discriminative between different classes and are similar within the same class.

To validate our coarse attention mechanism, we first conduct experiment on CIFAR-10 [Krizhevsky *et al.*, 2009] dataset. For better visualization and avoid dimensionality reduction, we use an extremely simple network as shown in figure 1 that ends up with two $6x6$ feature maps before GAP and FC layers for training two categories of images. We load data from CIFAR-10 batch files, and save new batch files with labels equal to 1 and 2, i.e. automobile and bird. We train

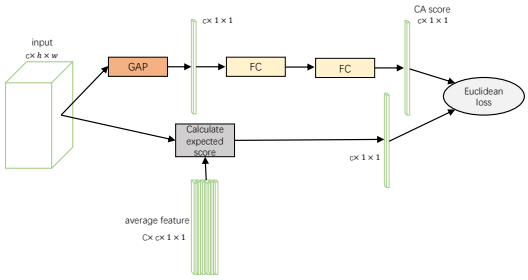


Figure 3: Coarse attention module

our network for 120 epochs and get a 93.67% final accuracy. Then we visualize the features after GAP as in figure 2 using the final model. We can notice that as described previously, average features after GAP cluster to two groups. For comparison, we train a model whose the only difference lies in that the features after GAP are multiplied by a vector with average feature of the two classes as start point and end point, as in Eq. 13. While achieving similar accuracy, 93.79%, the splitting bound is obviously better for classification. We also notice that the features are sparse at the channel dimension, i.e. the features for these two classes are mainly encoded by channel 0 and 1 respectively. When the case extends to high dimensions, it will highly benefit the training process due to better properties of sparse systems: efficient, compact and the ability to avoid overfitting. We will validate our approach on larger datasets in the experiment part.

3.2 Fine Attention

Although coarse attention mechanism has proven to be effective, we notice that it cares more about big patterns or the overall contour. We use the toolbox provided by [Yosinski *et al.*, 2015] to visualize where coarse attention pays attention. Some examples are provided in Figure 4. With only coarse attention module inserted in the original Faster R-CNN, we train our model end to end on our challenging dataset, which outperforms Faster R-CNN baseline. Then we visualize what coarse attention has learned using Deconvnet proposed in [Zeiler and Fergus, 2014]. Now we get two pairs by choosing the same channel whose attention scores are within top2, predicted by the coarse attention module, as a pair. We can see in Figure 4 that what coarse attention pays more attention corresponds to patterns like the center part of the whole car tile or the right part of the car.

Though the result further validates our idea, we need a fine attention module to precisely localize the distinguishable regions in the spatial domain. Since we have already obtained the ranking of importance of feature dimensions according to if the dimension is likely to extract discriminative regions. So intuitively, if most of them are reliable, we can assign the attention score to each channel, and adopt a similar routing-

Figure 4: Two pairs of examples that with top2 coarse attention scores in proposal regions. **Top:** original image; **Mid:** the feature map of channels whose coarse attention scores are top2; **Bottom:** deconvolution network for visualizing patterns the feature map depicts [Yosinski *et al.*, 2015].

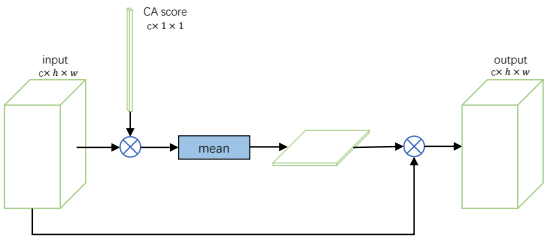


Figure 5: Fine attention module

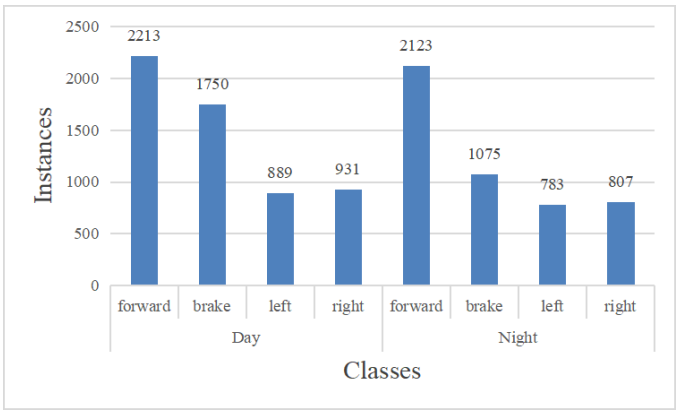


Figure 6: Data distribution of Vehicle Light Signal (VLS) dataset.

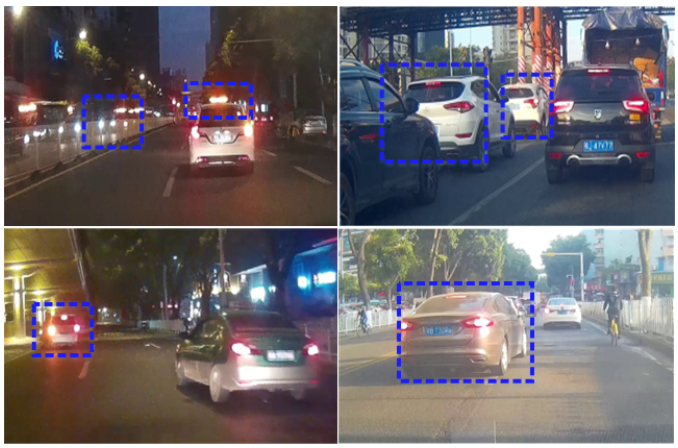


Figure 7: Difficult examples in Vehicle Light Signal (VLS) dataset.

by-agreement mechanism proposed in [Sabour *et al.*, 2017] to generate a spatial attention mask that informative channels agree. The architecture of our fine attention module is illuminated in Figure 5. In this way, we can pay more attention to the fine-grained subparts.

4 Experiments

4.1 VLS Datasets

Our Vehicle Light Signal (VLS) dataset contains 4 common behaviours of vehicles: driving forward, braking, turning left, turning right. Each behaviour signal contains 2 classes: day and night, since the lighting signals are not the same when during the day and night. We collect our data from the driving recorder by uniform sampling 15 frames from one video, to avoid very similar images. One video is 15 minutes, containing continuous real-world road scenarios. The overall vehicle category is: L2, L6, L7, M1, M2, M3 according to the UNECE categories. The road videos are mainly taken in Guangdong Province, China. Since we only focus on the signals of other vehicles, the impact of location distribution should be trivial. We hired five experienced annotators to annotate the bounding box of the obvious vehicles, whose number is up to 10 in one image, with the above 8 signal categories. Our dataset includes 7720 images, 8 categories, and 10571 instances totally. The data distribution statistics can be viewed at histogram in Figure 6. We randomly choose 60% of samples as the training data, 20% as validation and 20% for testing on our VLS dataset, and fix them for different models for fair comparison in experiments.

As discussed at part 1, the four difficulties of VLS dataset are: **(I)** Interference from environment like street lamp and other cars' front or rear lighting when detecting a particular car. **(II)** Occlusions of vehicles. **(III)** Low-light scenarios that distinguishing surrounding cars is difficult, with only salient lights in view. **(IV)** Strong-light scenarios that the tail lights are difficult to observe with reflection. Examples are in Figure 7, the blue dotted box reflects the four difficult-to-detect scenarios that mentioned above.

4.2 Implementation Details

We implement our CFA using Caffe [Jia *et al.*, 2014]. The models are trained on one NVIDIA GTX 1080Ti. For all our models, the initial learning rate is set to 10^{-3}, the momentum is set to 0.9 and the weight decay is 5×10^{-4}. We use the

Methods	backbone	C-A	F-A	day_no	day_break	day_left	day_right	night_no	night_break	night_left	night_right
Faster RCNN	VGG16			67.17	84.31	39.65	53.46	84.33	80.79	39.53	39.16
Ours	VGG16	√		78.42	86.32	42.02	45.75	85.97	81.91	43.16	47.22
	VGG16	√	√	79.88	86.51	45.69	55.21	86.14	83.50	49.28	51.67

Table 1: Quantitative ablation experimental results on VLS dataset.

Figure 8: Examples of successful rear light detection. These are results of driving forward, braking and turning respectively from top to bottom, each with day and night situations.

average precision (AP) of each category and the mean average precision (mAP) in object detection as our performance evaluation criteria.

4.3 Results and Analysis

Table 1 shows that our coarse to fine attention mechanism consistently outperforms baselines in terms of average precision on VLS datasets. We also conduct ablation experiments on the coarse attention and fine attention components. Through a comparative analysis of the experimental results in Table 1, we found that using only coarse attention can improve the mAP by $**\%$, and when using coarse attention and fine attention in combination, mAP can increase by $**\%$, which performs best.

Some visualizations of where each attention pays attention to are provided in figure 9. We can see that our coarse attention focuses on coarse patterns of automobiles, e.g. the right part of the tail, while with fine attention, the network focuses on the fine parts, e.g. the rightmost light of the car. However, for Faster RCNN, the attention is inconspicuous.

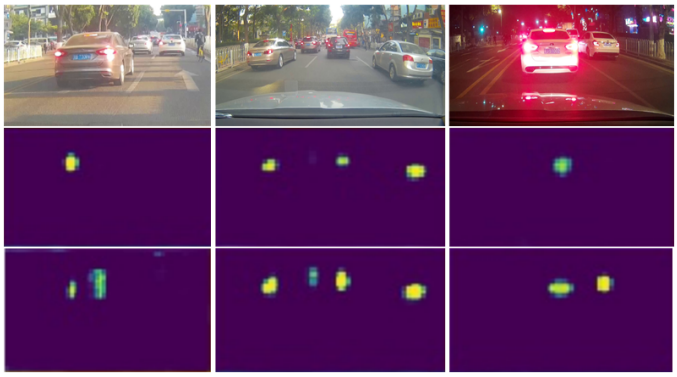


Figure 9: There examples with classification probability of RPN output that also indicates coarse attention scores. Top: original image; Mid: the feature map of channels in base network; Bottom: the feature map of channels in our network.

In Figure **??**, we provide the statistics of proposals generated from the first stage of two-stage detectors, e.g. in Faster RCNN, the proposals generated by RPN. We can notice that ....

5 Conclusion and Future Works

In this paper, we have introduced the coarse-to-fine attention mechanism for vehicle light signal detection, which can be used as an effective add-on to detection network while keeping trained end-to-end. We provide reliable theoretical derivation and experimental verification for our proposed network. We also publish a challenging real scene vehicle light signal (VLS) dataset. For future works, We will combine continuous information in time to improve the accuracy of turning light detection, and try to model the change process of the signal state of the rear lights to accurately capture the characteristics of the states changement.

Acknowledgments

We would like to give our thanks for the support in part by grants from National Science Foundation of China (No.61571005, No.61811530271), the China Scholarship Council (CSC NO.201806155037), the Science and Technology Research Program of Guangzhou, China (No.201804010429), the Fundamental Research Funds for the Central Universities, SCUT (No.2018MS57).

References

[Bello *et al.*, 2019] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Cordts *et al.*, 2016] 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*, pages 3213–3223, 2016.

[Everingham *et al.*, 2010] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[Hu *et al.*, 2018] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[Jaderberg *et al.*, 2015] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[Jia *et al.*, 2014] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014.

[Khandelwal and Sigal, 2019] Siddhesh Khandelwal and Leonid Sigal. Attentionrnn: A structured spatial attention mechanism. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[Li *et al.*, 2019] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[Lin *et al.*, 2017] 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*, pages 2980–2988, 2017.

[Redmon *et al.*, 2016] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[Ren *et al.*, 2015] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[Sabour *et al.*, 2017] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[Wang *et al.*, 2019] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2965–2974, 2019.

[Xu *et al.*, 2017] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2174–2182, 2017.

[Xu *et al.*, 2019] Hang Xu, Lewei Yao, Wei Zhang, Xiaodan Liang, and Zhenguo Li. Auto-fpn: Automatic network architecture adaptation for object detection beyond classification. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[Yosinski *et al.*, 2015] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[Zeiler and Fergus, 2014] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[Zhou *et al.*, 2016] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.