# ROI-wise Reverse Reweighting Network for Road Marking Detection

Xiaoqiang Zhang[1]
xqzhangnwpu@gmail.com
Yuan Yuan[1]
y.yuan1.ieee@gmail.com
Qi Wang*[12]
crabwq@gmail.com

[1] School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL) Northwestern Polytechnical University Xi'an, P. R. China

[2] Unmanned System Research Institute Northwestern Polytechnical University Xi'an, P. R. China

**Abstract**

Road markings provide essential navigation information for automated driving and driver assistant systems. As road markings vary dramatically in scale and shape, different markings in various traffic scenes may be sensitive to different level of deep convolutional features. Practically, it is difficult to accurately define which layers are more useful for detecting a marking. It is also inadvisable to extract the same distributed multi-layer features for all ROIs like previous methods (ROI means the region of interest that probably contains markings), which ignores the differences of different markings. To remedy this problem, we propose a novel ROI-wise Reverse Reweighting Network ($R^3Net$) to adaptively combine multi-layer features for different markings. It consists of a multi-layer pooling operation and a ROI-wise reverse reweighting module, which independently generates a specific distribution over multi-layer features for each ROI to model different ROI's unique properties. To evaluate our method, we construct a large dataset that contains about 10000 images and 13 categories. Experimental results demonstrate that our proposed network is more effective than others for using multi-layer features.

## 1 Introduction

As shown in Figure 1, road markings refer to the signs painted on the road, which are different from the traffic signs locating beside or above the road. Many categories of road markings can provide more accurate traffic information than traffic signs. For example, an arrow drawn in the lane can indicate the navigation information of the specific lane. Therefore, the detection of road markings is also an important component of Intelligent Transportation System (ITS). They contribute to automated driving and driver assistant systems that require precise navigation information.

However, in the last decades, there are very few literatures on road markings. Recently application of deep convolutional neural networks has made significant progress in general object detection, but no progress has been made in road marking detection. Indeed, it faces more challenges in some respects than general objects which can be summarized as follows:

Figure 1: Examples of road markings in our dataset. The first two lines show 13 categories of road markings. From left to right and top to bottom, they are 60, 70, car_people, center_ring, cross_hatch, diamond, forward_left, forward_right, forward, left, and right. The last line shows some images describing different traffic scenes.

- In the real traffic environments, the visibility of road marking is easily reduced by many factors, such as the occlusion of vehicles and pedestrians, the influence of weather and light, and the wear of markings' appearances.
- The scales and shapes of road markings vary dramatically due to the camera's perspective change, distortion, and projection transformation.
- There are local similarities for different categories of road markings.

Obviously, image processing and traditional machine learning methods cannot effectively deal with these difficulties. It has been well recognized in the vision community for years that deep features perform much better than handcrafted features and shallow features. Moreover, RCNN-based detection framework is currently the mainstream for general object detection [7, 8, 13]. However, the direct application of this method to road markings does not yield satisfactory results.

Road markings are artificially designed signs with regular lines and shapes. In human vision, colors and outlines may be very important features that distinguish road markings. These features correspond exactly to low-level features of deep convolutional network. Therefore, it is a natural idea to simultaneously use the low-level features and high-level semantic information to improve the generalization ability of features. Furthermore, since road markings vary dramatically in scale and shape, different markings may be sensitive to different level of deep convolutional features. For example, when detecting a small marking, lower-level features are necessary to obtain the missing details. While detecting an occluded marking, robust high-level features are needful to exclude the effects of occlusion. In other words, the distributions over multi-layer features for different markings are affected by their current states. Disappointingly, previous methods usually extract the same distributed multi-layer features by a convolution layer before or after the ROI pooling for all ROIs, which ignore the differences of different ROIs [7, 11, 12].

In this paper, we propose a novel ROI-wise Reverse Reweighting Network ($R^3Net$) to adaptively combine multi-layer features for road marking detection. As illustrated in Figure 2, it consists of two key components: multi-layer pooling (MLPool) operation and a ROI-wise reverse reweighting ($R^3$) module, which is built upon Faster-RCNN [13]. MLPool operation obtains multi-layer features through ROI-pooling for each ROI. $R^3$ module is derived from an attention model which has achieved great success in natural language processing [22, 23]. It can adaptively generate a specific distribution over multi-layer features for each ROI by combining information from current features and reverse feed feature vector. This

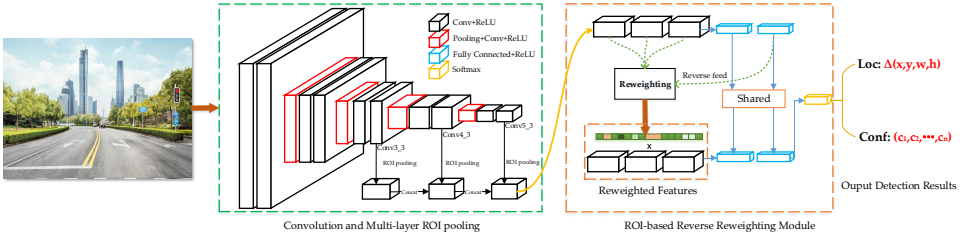



Figure 2: $R^3Net$ consists of a multi-layer pooling (MLPool) operation and a ROI-wise reverse reweighting ($R^3$) module. MLPool operation obtains multi-layer features through ROI-pooling. $R^3$ module can adaptively generate a specific distribution over multi-layer features for each ROI by combining information from current features and reverse feed feature vector.

distribution describes which channels of multi-layer features are more conducive to the detection for each ROI. Finally, the reweighted multi-layer features can be obtained based on this distribution. The most important difference compared to other methods for reweighting features such as SENet [9] is that it additionally accepts a reverse feed feature vector as input which helps to generate better distributions. The main contributions of this work are summarized as follows:

- A novel ROI-wise Reverse Reweighting Network ($R^3Net$) for road marking detection is proposed, which achieves a 7.24% mAP improvement compared to baseline on road marking dataset.
- Comprehensive ablation studies and experimental analysis are conducted, which prove that our proposed reweighting module is effective for using multi-layer features.
- A new large dataset is constructed to evaluate our method, which partially compensates the lack of dataset in road marking.

## 2 Related Works

This section introduces some works related to our proposed method.

**Road marking detection methods.** The previous methods mainly use image processing and traditional machine learning algorithms for road marking detection. Many researchers usually use Inverse Perspective Mapping (IPM) transformation [3] to preprocess the original image, which can reduce the distortions and variations of road markings due to 3D space projection [1, 6, 10, 14, 16, 17, 24]. Then, some image processing methods are used to generate candidate regions (ROIs) from the bird's-eye image of the road, such as hough transform [17], MSER features [1, 24], geometric feature extraction [10], and high brightness slice filtering [14]. Finally, the ROIs are recognized using some classification methods with extracted features, such as KNN classifier with Fourier descriptors [17], template matching with Histogram of Oriented Gradients (HOG) features [24], Adaboost classifier with Haar-like features and ELM classifier with BW-HOG features [14], softmax with shallow CNN features [1]. The IPM-based methods depend on camera-calibrated parameters that are easily changed or difficult to obtain in some cases. So some methods for extracting ROIs from the original image are proposed [4, 21]. Specifically, Chen *et al*. proposes a general framework which firstly extracts ROIs using binarized normed gradient (BING) method and then

applies the PCA network (PCANet) for classification [4]. In summary, they are all two-step methods including ROI generation and classification. Although well-performing CNNs have been adopted in classification, the generation of ROIs mostly employs image processing methods, resulting in bad generalization. To our knowledge, the method we propose is the first application of an end-to-end CNN-based detection method in road marking.

**Multi-layer features.** In last few years, the Region-based CNN methods have made a great achievement in general object detection [7, 8, 18]. In particular, following RCNN, many methods using multi-layer features are proposed to improve performance for multi-scale object detection [2, 11, 12, 13, 25, 26]. There are two main ways to process multi-layer features. One is to fuse features from different layers [2, 11, 12, 26], and the other is to individually employ features of each layer [13, 25]. The former generally concatenates multi-layer features before ROI pooling [11, 12] or after ROI pooling [2, 26], and then fuses them by adding a convolution operation. This method assumes that the multi-layer features required by all objects obey the same distribution, ignoring the differences of objects under different circumstances. The latter usually makes independent predictions by mapping objects of different sizes onto feature maps of corresponding resolutions [13, 25]. Although this approach takes into account the individual differences of objects, it is merely a heuristic idea that small objects needs low-level features while large objects needs high-level features. Besides, it requires abundant training samples for various scale objects. Therefore, as far as we know, our proposed $R^3Net$ is the first time to focus on learning a distribution over multi-layer features for each object.

# 3 ROI-wise Reverse Reweighting Network

As shown in Figure 2, $R^3Net$ is built upon Faster-RCNN [18] and consists of two key components: a multi-layer pooling operation and a ROI-wise reverse reweighting module. For the remainder of this section, we firstly review the Faster-RCNN baseline and then describe these two components in detail respectively.

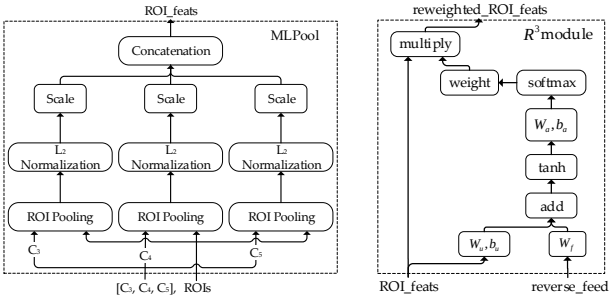
## 3.1 Baseline:Faster-RCNN

Faster-RCNN is composed of a region proposal network and a region recognition network, both of which share the same deep convolutional features. As a two-stage detector, its backbone network firstly takes a whole image as input to get the shared features, and then it generates ROIs that probably contain objects through region proposal network. Finally region recognition are carried out with region features extracted by ROI pooling. The entire Faster-RCNN network can be end-to-end trained with cross entropy loss and $Smooth_{L_1}$ loss. For the sake of readability, we use CNN, RPN, ROIPool and RCNN to represent shared deep convolutional network, region proposal network, ROI pooing and region recognition network respectively. The entire detection architecture can be expressed simply as:

$$C_i(i=1,\cdots,5) = CNN(img), ROIs = RPN(C_5), \tag{1}$$

$$ROI\_feats = ROIPool(C_5, ROIs), predictions = RCNN(ROI\_feats, ROIs). \tag{2}$$

In Eq. (1), $img$ denotes the input image, $C_i$ represents the convolutional features of the $i$th layer (the same scale feature maps are treated as the same layer) and $i$ is in the range of 1 to




(a) *MLPool* (b) $R^3$ module

Figure 3: The details of *MLPool* and $R^3$ module.

5 in this paper, because VGG16 [20] is applied as a backbone network. From Eq. (2), we can see that $ROI\_feats$ only use the features of $C_5$.

## 3.2 Multi-layer Pooling Operation

Different from general objects, road markings are designed with lines and arrows. Thus low-level gradient features are as important as high-level semantic features. To elegantly combine low-level features and high-level semantic information, we use multi-layer pooling operation to generate $ROI\_feats$. At this point, the first formula in Eq. (2) can be written as:

$$ROI\_feats = MLPool([C_5, C_4, C_3], ROIs). \tag{3}$$

where *MLPool* denotes the multi-layer pooling operation. The details of *MLPool* is illustrated in Figure 3(a). It takes $C_5, C_4, C_3$ and ROIs as input. For each ROI, it gets pooled features from $C_5, C_4, C_3$ through ROI pooling. Then, the pooled features are normalized by $L_2$ normalization and scaled by $\alpha$ before concatenation.

## 3.3 ROI-wise Reverse Reweighting Module

Since road markings vary dramatically in scale and shape, our proposed ROI-wise reverse reweighting ($R^3$) module can adaptively learn a specific distribution over multi-layer features for each ROI to model different ROI's unique properties. The main idea is that the final feature vector contains important decision information and can promote the learning of distribution. Generally, this module combines information from the $ROI\_feats$ and reverse feed feature vector to generate the reweighted features, which is written as:

$$reweighted\_ROI\_feats = R^3(ROI\_feats, reverse\_feed). \tag{4}$$

Correspondingly, we define the *predictions* in Eq. (2) as:

$$predictions = RCNN(reweighted\_ROI\_feats, ROIs). \tag{5}$$

where the $ROI\_feats$ denotes the features generated by multi-layer pooling operation and the $reverse\_feed$ denotes the $fc7$ features of VGG16, which can be obtained by adding two fully connected layers on $ROI\_feats$. The reweighted features go through the same two fully connected layers again and finally connect to the network output. As illustrated in Figure 2,

*reverse* means $fc7$ is not directly connected to the output of detection but is instead input to the $R^3$ module, which can transfer important decision information. During training, this shortens the gradient propagation path between $R^3$ module and network losses, which can accelerate the flow of information and more effectively supervise the train of $R^3$ module to get better multi-layer features.

We now describe $R^3$ module in detail. In Figure 3(b), the input consists of $ROI\_feats$ and $reverse\_feed$, which have the shape of $N \times C \times H \times W$ and $N \times D_f$. Here, $N$ represents the batch size (ROI number), $C$ is the total number of channels, $H, W$ is the height, width of feature maps, and $D_f$ is the dimension of $reverse\_feed$. Since $R^3$ module has the same computation for all ROIs, we only consider one for simplicity. For each ROI, we first reshape $ROI\_feats \in R^{C\times H\times W}$ to $U = [u_1, u_2, \cdots, u_C]$, where $u_i \in R^{H\times W}$ represents the $i$-th channel features and use $F \in R^{D_f}$ to represent $reverse\_feed$. Then, the distribution $weight \in R^C$ over multi-layer features can be generated through simple computation followed by a softmax function. Finally, the $reweighted\_ROI\_feats$ can be obtained by the multiplication of $ROI\_feats$, $weight$, and scale factor $\beta$. The complete definitions are as follows:

$$\begin{aligned} a &= tanh((W_u U + b_u) \oplus W_f F), \\ weight &= softmax(W_a a + b_a), \\ reweighted\_ROI\_feats &= U \otimes weight \otimes \beta. \end{aligned} \tag{6}$$

where $W_u \in R^{K\times(H\times W)}$, $W_f \in R^{K\times D_f}$, $W_a \in R^K$ ($K$ is the dimension of inner vector) are module weights, and $b_u$, $b_a$ are bias terms. They are all learnable variables. $\oplus$ and $\otimes$ represent element-wise addition and multiplication with broadcasting. $\beta$ is a scalar to recovery the response amplitude before reweighting.

# 4 Experiments

We validate the effectiveness of the proposed $R^3Net$ for road marking detection by comparing with some popular detection methods and conducting comprehensive ablation studies on our road marking dataset.

## 4.1 Dataset

To our knowledge, there are no large public datasets with manual annotations that allow us to train the deep convolutional network, so we only evaluate on our dataset. Our road marking dataset consists of RM *train* set and RM *test* set. They come from 55 videos and 30 videos, respectively. These videos are from a lot of driving videos in real traffic scenes in China. Because of the wide variety of road markings, we select 13 categories that have appeared most in videos to be manually annotated. We represent them as 60, 70, car_people, center_ring, cross_hatch, diamond, forward_left, forward_right, forward, left, right. Note that they are only used as category identifiers and some of them do not have specific meanings. Some image examples of each category are shown in Figure 1. Our dataset contains a total of 9394 images with 6,472 in *train* set and 2,922 in *test* set. The resolution of all images is $1280 \times 720$. One main challenge of our dataset is that it contains a variety of weather conditions such as rainy, cloudy, and various traffic scenes such as crowded urban roads, highways and so on. Another important challenge is the imbalance of samples in each category, as shown in Figure 4.

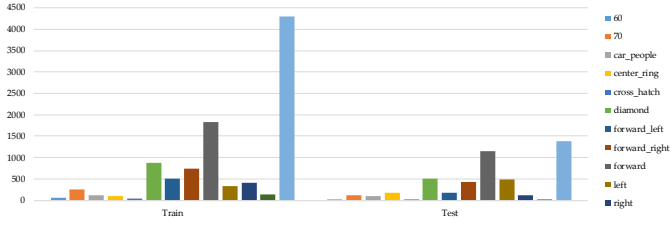



Figure 4: Distribution of samples in each category in our dataset.

| Method | mAP | 60 | 70 | car_people | center_ring | cross_hatch | diamond | forward_left | forward_right | forward | left | right | u_turn | zebra_crossing |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Faster-RCNN[9] | 62.71 | 27.99 | 64.82 | 64.68 | 38.50 | **38.57** | 79.62 | 74.23 | 71.22 | 75.24 | 62.51 | 67.62 | 85.79 | 64.42 |
| SSD300[9] | 54.14 | 6.67 | 56.28 | 57.06 | 35.97 | 28.78 | 77.37 | 68.16 | 62.51 | 69.80 | 44.22 | 59.65 | 77.50 | 59.81 |
| SDD512[9] | 61.27 | 18.03 | 67.55 | 60.64 | 37.40 | 11.58 | **81.60** | 73.79 | 75.47 | 77.19 | 63.83 | 72.10 | **93.77** | 63.58 |
| $R^3Net$ (ours) | **69.95** | **60.14** | **73.84** | **72.23** | **38.52** | 35.84 | 81.20 | **80.27** | **75.51** | **81.22** | **77.21** | **73.87** | 93.75 | **65.80** |

Table 1: The road marking detection results on our dataset. Our proposed $R^3Net$ achieves a 7.24% mAP improvement compared to Faster-RCNN baseline.

## 4.2 Implementation Details

We use VGG16 as the backbone network in all experiments. Following Faster-RCNN, three anchor scales $\{128, 256, 512\}$ and three aspect ratios $\{1:2, 1:1, 2:1\}$ are set. The $R^3Net$ is end-to-end trained with stochastic gradient descent (SGD) and the parameters are initialized with a pre-trained model on ImageNet [19]. All models based on Faster-RCNN are trained for 10 epochs with an initial learning rate of 0.001, which is then divided by 10 at 7 epochs. Weight decay of 0.0005 and momentum of 0.9 are used. In practice, the scaling factors $\alpha$, $\beta$ are 1000 and $K$ is set to 512 in section 3.2 and 3.3. Note that all models are trained on RM *train* set without image flipping, because many categories in flipped image have wrong labels, for example, left arrow in flipped image will be right arrow. And all models are evaluated on RM *test* set.

## 4.3 Evaluation Results and Ablation Studies

We firstly train Faster-RCNN and SSD [15] as baselines on our road marking dataset. SSD is a one-stage detection architecture, which is very popular as well as Faster-RCNN. Then we evaluate our proposed ROI-wise Reverse Reweighting Network ($R^3Net$). Following the evaluation criterion of Pascal VOC [5], detection accuracy is measured by mean Average Precision (mAP) with IoU threshold of 0.5. The results are shown on Table 1. It is worth noting that our proposed $R^3Net$ achieves a 7.24% (69.95% vs. 62.71%) mAP improvement compared to Faster-RCNN baseline. Our method is superior to Faster-RCNN and SSD on most categories. This demonstrates that our proposed method is highly effective for road marking by adding a *MLPool* operation and a $R^3$ module on Faster-RCNN. Some detection examples are shown in Figure 6. We also see that the results of SSD are quite affected by the size of input image (54.14% vs. 61.27%) and Faster-RCNN performs a little better than SSD (62.71% vs. 61.27%). Besides, it is observed that some categories with small samples usually have lower performance than others such as 60, center_ring and corss_hatch. Next we mainly analyze the role of each component.

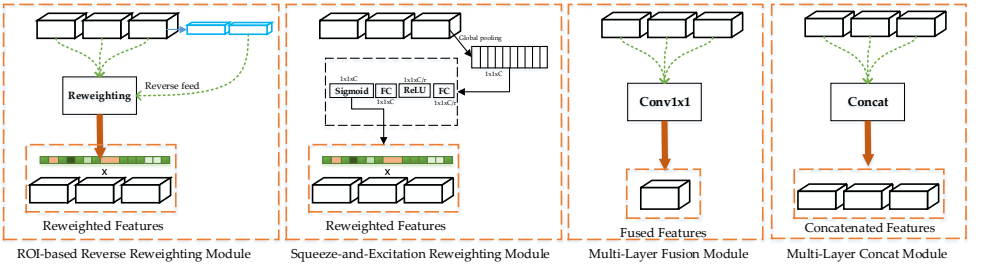



Figure 5: Illustration of four methods using multi-layer features. From left to right, they are (a) ROI-wise Reverse Reweighting ($R^3$) module, (b) Squeeze-and-Excitation Reweighting (SER) module, (c) Multi-Layer Fusion (MLF) module, (d) Multi-Layer Concat (MLC) module respectively.

### 4.3.1 The Effects of Multi-layer Features

In order to observe the effects of multi-layer features on road marking detection, we remove the reweighting module in our proposed method and add a $1 \times 1$ convolution layer after *MLPool*. We expect a $1 \times 1$ convolution to fuse multi-layer features by setting the number of output channels to 512, which is less than the total number (1280) of input channels. This new structure is called Multi-Layer Fusion (MLF) module, as shown in Figure 5(c). The comparison results of Faster-RCNN with MLF and Faster-RCNN are shown in Table 2. We see that there is a 3.81% mAP improvement by only adding a MLF module. This proves that low-level features and high-level semantic information included in multi-layer features are contributed to the detection of road markings with varying scales and shapes.

### 4.3.2 The Effects of Reweighting Module

We set up three structures to verify the importance of reweighting module, which are $R^3$, MLF, MLC, corresponding to (a), (c), (d) in Figure 5, respectively. Multi-Layer Concat (MLC) is set to exclude the effect of the number of feature channels, which does nothing for *MLPool*'s output. Their results are shown in Table 2. We see that $R^3$ module is 3.43% better than MLF and 3.71% better than MLC. The fact is reasonable that reweighting module performs better than others. We think that the distribution over multi-layer features is affected by many factors such as size for each ROI. Our proposed $R^3$ module has the ability to learn a specific distribution over multi-layer features for different ROI. However, MLC module regards the features of different layers as equally important for all ROIs at any case. Although MLF module can fuse multi-layer features, its parameters learned by training are fixed. It means that MLF module learns the same distributed multi-layer features for all ROIs, which ignores the differences for different ROI. We also observe that MLF and MLC have almost the same results, which again proves that not all channels are equally important, and that $1 \times 1$ convolution is too weak to effectively mine valuable information from multi-layer features.

### 4.3.3 The Effects of Reverse Feed

The most important difference between our method and others for reweighting features is that it additionally accepts a reverse feed feature vector as input. To confirm the need for

| Method | mAP |
|---|---|
| Faster-RCNN[13] | 62.71 |
| Faster-RCNN + MLF | 66.52 |
| Faster-RCNN + MLC | 66.24 |
| Faster-RCNN + SER | 67.25 |
| Faster-RCNN + $R^3$ | 69.95 |

Table 2: The road marking detection results of ablation studies on our dataset.

reverse feed, we compare our $R^3$ module to SER module that is shown in Figure 5(b). The SER module is designed based on the idea of the SENet [9]. Firstly, it gets a feature vector by global pooling on multi-layer features for each ROI. Then, the weights for each channel are generated by adding two fully connected layers. The first one has $C/r$ units with ReLU activation and the other has $C$ units with Sigmoid activation, where $r$ is set to 16 in our experiment. Finally, the reweighted features are acquired by multiplying multi-layer features by weights with broadcasting. From the comparison results in table 2, we see that $R^3$ module with reverse feed is 2.7% better than SER module with merely current features as input. We think this is because the reverse feed (fc7), which is close to the network output, contains very important decision information and can guide the reweighting module to learn useful weights easier.

### 4.4 Discussion

We have validated the effectiveness of the proposed $R^3Net$ on road marking dataset. It is proven that the reweighting module is effective in dealing with objects that vary in scale and shape. We want to verify whether this module is also suitable for general object detection, so we do an experiment on Pascal VOC that has 20 object categories. We train the Faster-RCNN with MLF and $R^3$ respectively on VOC 2007 *trainval* set and evaluate on VOC 2007 *test* set. From the results (70.33% vs. 68.06%), we see that $R^3$ module is 2.27% better than *MLF* for mAP. To a certain extent, it turns out that the reweighting module also performs better than $1 \times 1$ convolution fusion in the detection of other objects.

## 5 Conclusion and Future Work

We propose a new ROI-wise Reverse Reweighting Network for road marking detection, which achieves a 7.24% mAP improvement compared to Faster-RCNN baseline. Experimental results demonstrate that multi-layer pooling operation and ROI-wise reverse reweighting module both play important roles in our framework. It is worth noting that reweighting module with reverse feed can learn a better distribution over multi-layer features. Recently, it is believed that modeling relations between objects is helpful for detection. However, it is difficult to obtain valuable relation information. We will try to apply this reweighting module to explore the appropriate relation information for road marking detection in the future.

More importantly, we construct a large dataset to evaluate different methods. As far as we know, our approach is the first end-to-end CNN-based detection method for road marking. Our dataset will be available later. We hope our method is a baseline, and welcome more researchers to study road marking using our dataset.


Figure 6: Detection examples on RM *test* with our proposed $R^3Net$. We show detections with scores higher than 0.6. Each color corresponds to an object category.

## 6 Acknowledgment

This work was supported by the National Key R&D Program of China under Grant 2017YFB 1002202, National Natural Science Foundation of China under Grant 61773316, Natural Science Foundation of Shaanxi Province under Grant 2018KJXX-024, Fundamental Research Funds for the Central Universities under Grant 3102017AX010, and the Open Research Fund of Key Laboratory of Spectral Imaging Technology, Chinese Academy of Sciences.

## References

[1] Oleksandr Bailo, Seokju Lee, Francois Rameau, Jae Shin Yoon, and In So Kweon. Robust road marking detection and recognition using density-based grouping and machine learning techniques. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 760–768. IEEE, 2017.

[2] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2874–2883, 2016.

[3] Massimo Bertozz, Alberto Broggi, and Alessandra Fascioli. Stereo inverse perspective mapping: theory and applications. *Image and vision computing*, 16(8):585–590, 1998.

[4] Tairui Chen, Zhilu Chen, Quan Shi, and Xinming Huang. Road marking detection and classification using machine learning algorithms. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 617–621. IEEE, 2015.

[5] 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.

[6] Philippe Foucher, Yazid Sebsadji, Jean-Philippe Tarel, Pierre Charbonnier, and Philippe Nicolle. Detection and recognition of urban road markings using images. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pages 1747–1752. IEEE, 2011.

[7] Ross Girshick. Fast r-cnn. *arXiv preprint arXiv:1504.08083*, 2015.

[8] 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.

[9] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.

[10] Hiroyuki Ishida, Kiyosumi Kidono, Yoshiko Kojima, and Takashi Naito. Road marking recognition for map generation using sparse tensor voting. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1132–1135. IEEE, 2012.

[11] Kye-Hyeon Kim, Sanghoon Hong, Byungseok Roh, Yeongjae Cheon, and Minje Park. Pvanet: deep but lightweight neural networks for real-time object detection. *arXiv preprint arXiv:1608.08021*, 2016.

[12] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 845–853, 2016.

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017.

[14] Wei Liu, Jin Lv, Bing Yu, Weidong Shang, and Huai Yuan. Multi-type road marking recognition using adaboost detection and extreme learning machine classification. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 41–46. IEEE, 2015.

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[16] Ziqiong Liu, Shengjin Wang, and Xiaoqing Ding. Roi perspective transform based road marking detection and recognition. In *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, pages 841–846. IEEE, 2012.

[17] J Rebut, A Bensrhair, and G Toulminet. Image segmentation and pattern recognition for road marking analysis. In *Industrial Electronics, 2004 IEEE International Symposium on*, volume 1, pages 727–732. IEEE, 2004.

[18] 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.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115 (3):211–252, 2015.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Mohak Sukhwani, Suriya Singh, Anirudh Goyal, Aseem Behl, Pritish Mohapatra, Brijendra Kumar Bharti, and CV Jawahar. Monocular vision based road marking recognition for driver assistance and safety. In *Vehicular Electronics and Safety (ICVES), 2014 IEEE International Conference on*, pages 11–16. IEEE, 2014.

[22] 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 6000–6010, 2017.

[23] Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781, 2015.

[24] Tao Wu and Ananth Ranganathan. A practical system for road marking detection and recognition. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 25–30. IEEE, 2012.

[25] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2016.

[26] Sergey Zagoruyko, Adam Lerer, Tsung-Yi Lin, Pedro O Pinheiro, Sam Gross, Soumith Chintala, and Piotr Dollár. A multipath network for object detection. *arXiv preprint arXiv:1604.02135*, 2016.