StripNet: Towards Topology Consistent Strip Structure Segmentation

Paper ID 247

ABSTRACT

In this work, we propose to study a special semantic segmentation problem where the targets are long and continuous strip patterns. Strip patterns widely exist in medical images and natural photos, such as retinal layers in OCT images and lanes on the roads, and segmentation of them has practical significance. Traditional pixel-level segmentation methods largely ignore the structure prior of stripped patterns and thus easily suffer from the topological inconformity problem, such as holes and isolated islands in segmentation results. To tackle this problem, we design a novel deep framework, StripNet, that leverages the strong end-to-end learning ability of CNNs to predict the structured outputs as a sequence of boundary locations of the target strips. Specifically, StripNet decomposes the original segmentation problem into more easily solved local boundary-regression problems, while putting the topological constraints on the predicted boundaries. Moreover, our framework adopts a coarse-to-fine strategy and uses carefully designed heatmaps for training the boundary localization network. We examine StripNet on two challenging strip pattern segmentation tasks, retinal layer segmentation and lane detection. Extensive experiments demonstrate that StripNet achieves excellent results and outperforms state-of-the-art methods in both tasks.

CCS CONCEPTS

• **Theory of computation → Models of learning**; *Structured prediction*; • **Computing methodologies → Neural networks**; Instance-based learning;

KEYWORDS

Strip Segmentation, Lane Detection, Retinal Layer Segmentation

ACM Reference Format:
Paper ID 247. 2018. StripNet: Towards Topology Consistent Strip Structure Segmentation. In *Proceedings of ACM Multimedia conference (ACM Multimedia 2018).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In this paper we target at segmenting certain long and continuous strip structures from input images. Strip structures widely exist in real life scenarios, e.g., retinal layers in OCT images and lanes on the roads, as shown in Fig. 1 (a). Understanding these structures from images is an important computer vision task. For example, segmentation of retinal layers in OCT images is the key step for the diagnosis of some eye diseases, while lane detection plays an important role in traffic scenario understanding, which helps guiding vehicles in autonomous driving.

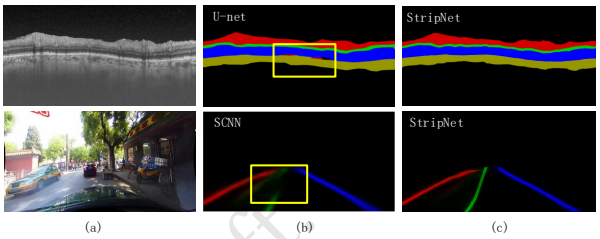

Figure 1: Examples of two strip structures and their segmentation results by previous methods and our proposed method StripNet. (a) retinal layers in OCT images and lanes on the road. (b) Results of previous methods (U-net [32] and SCNN [42]. (c) Results of StripNet. Note conventional FCN based methods exhibit topological errors (highlighted by rectangles), while the proposed StripNet could avoid topological inconformity problems.

The strip structures distribute contiguously as a connected component with no holes or isolated islands, which forms strict topology priors. In other words, there should be no more than one connected segmentation component in any column/row of the image. However, most previous segmentation methods do not specifically distinguish between this kind of stripped patterns and other targets. Currently popular paradigm [5, 6, 23, 34] may segment the image is to classify each pixel independently into one of the predefined categories. These pixel-level segmentation methods naturally encounter the topological inconformity problem, these various topological errors includes isolated islands or holes in the segmentation results, as shown in Fig. 1 (b), which require extra post-processing after neural networks [5, 6, 25].

To address this challenging problem and fulfill the topological constraint, we propose a novel deep architecture, called StripNet, for segmentation of strip structures. We design a structured output by decomposing the strips into a sequence of connected regions, which solves the problem of inconsistent topology as shown in Fig. 1 (c). More specially, StripNet uniformly divide the whole image into columns or rows with fixed width or height, and predicts the existence and boundaries (if exists) of the strip in each column or row. By doing so, at most one connected component of a strip will be obtained, and the strip can be constrained by the boundary, as shown in Fig. 2.

The strip structures only occupy a small portion of the image and are difficult to directly predict its locations from whole divided

ACM Multimedia 2018, October 2018, Seoul, Korea

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

column. Therefore, we design a coarse-to-fine approach to solve this problem. Firstly, we roughly predict the location of strip in each column. Since the width of the region is fixed, only the up and down boundaries are needed to be predicted to get the rough Region of Interest (RoI). This step does not give the exact prediction of the strip location, but helps clean out many other unrelated areas that may distract the prediction, and this helps a lot for the precise prediction in next stage. Then we use RoIAlign [14] to extract the feature in the RoI extracted from CNN, and predict the location of the strip precisely, that is, *precise boundary regression*. We design StripNet to predict the location in the form of heatmap regression in both two stages, because we find that strip structures still count little in RoI, thus directly predict one coordinate could cause deviation easily, but heatmap can reflect the distribution of objects in regions more directly and precisely. And this has been proved to be more stable and accurate than directly predicting coordinates in [9, 27] or using the anchor mechanism [31]. After that, we connect the points obtained in *precise boundary regression* that belong to the same boundary, and arrange areas between same boundaries to the same layers, which prevent us from topological errors such as holes and isolated areas.

To summarize, our contributions are three folds:

1) As far as we know, this work is the first attempt to develop a deep architecture for strip segmentation which effectively integrates the topological priors of strip patterns and the end-to-end learning ability of CNNs. We elaborately design a structured output as a sequence of proposals to guarantee the topology consistency.

2) To tackle the imbalance problem between the strip structures to be segmented and the backgrounds, our StripNet performs segmentation in a coarse-to-fine manner. In the coarse stage the region of the strips is roughly localized in each column of the image and in the fine stage its score and precise locations are predicted. Locations of the strips are generated using a carefully designed heatmap, with Gaussian kernels indicating the boundaries of the strips.

3) We evaluated the proposed framework on two distinct tasks, i.e., retinal layer segmentation in OCT images and lane detection in road images. Extensive experiments on publicly available dataset (for lane detection) and self-collected dataset (for retinal layer segmentation) show that our method outperforms the state-of-the-art approaches, without having topological errors.

2 RELATED WORK

2.1 Semantic Segmentation by Deep Learning

The task of semantic segmentation is to assign a predefined label to each pixel on a given image. As one of the basic problems in computer vision, extensive research efforts have been devoted in this field [1, 5, 6, 22, 29]. In recent years, deep learning based methods have dramatically improved the performance of semantic segmentation. Farabet et al [10] proposed a multi-scale convolutional neural network (CNN) to predict the label of each image patch densely sampled from the image, and applied superpixel voting or Conditional Random Field (CRF) for improving the smoothness of the prediction. Pinheiro et al [28] introduced a Recurrent Neural Network (RNN) to recurrently refine its predictions by concatenating the RGB image with its predicted masks as input. Both [10, 28] are patch-based deep models, which are redundant in computation and time-consuming. In 2015, [23] proposed the Fully Convolutional Network (FCN) which takes the whole image as input and outputs the prediction in the same resolution, which is achieved by replacing the fully connected layers with convolution layers and adding deconvolution layers for upsampling. The design of FCN makes semantic segmentation an end-to-end trainable problem and largely improved the efficiency. Therefore, a lot of FCN-based works [6, 32, 35, 43] are proposed that further boost the performance of semantic segmentation. [32] used skip-connections between lower layers and higher layers to add more detailed information for the fine resolution prediction. [6] propose to refine the segmentation results of CNN by post-processing with CRF, as the raw output of CNN might contain isolated islands or hole errors.

Our tasks, retinal layer segmentation and lane detection, differ from general semantic segmentation as the targets to be segmented are long and thin regions. Moreover, each category (e.g., certain lane or retinal layer) usually has at most one connected component. Directly applying general semantic segmentation to these tasks ignores the high-level structure priors and may lead to topology errors such as isolated islands and holes. Different from the FCN-based methods which are based on pixel-level predictions, we integrate the high-level structure priors with the powerful expressive ability of deep models to overcome this disadvantage. We replace the pixel-level prediction with a structured output, which can easily eliminate topological errors.

2.2 Retinal Layer Segmentation

Automated methods for layer segmentation and measuring layer thicknesses in OCT images have been widely studied [11, 20, 26, 26, 30]. [20, 26] exploited random forest and level set to produce accurate boundaries of retinal layers in B-scan OCT images. For the segmentation of 3-D spectral OCT images, a graph-theoretic method is proposed by [11], and [30] presents a novel probabilistic approach and achieves impressive results. In recent years, some deep learning approaches [16, 33] apply FCN-based networks for retinal layer segmentation. These methods leverage the strong representation ability of deep models and perform better than conventional methods. However, they still suffer from topological errors. [16] proposed the topology correction network for refining the topologically incorrect images. However, there is no mathematical guarantee of the result to be topology consistent and it costs extra time for post-processing.

2.3 Lane Detection

One commonly used approach for lane detection is to detect edges by various kinds of filters and then use Hough transform [7, 18, 37, 38] to fit lines to these edges. However, as these methods are based on low level features, they are very sensitive to illumination variations or road condition changes. Inspired by the success of deep learning methods in image classification [8, 15, 36] and segmentation [6, 23], neural networks were introduced to tackle the lane detection problem [12, 13, 17, 19, 21]. At first the CNN was used as feature extractor [12] or for image enhancement [19]. Then end-to-end CNN frameworks for lane detection and classification is proposed [13, 17]. However, these aforementioned networks use CNNs that are designed for general purpose without leveraging the

high level structure priors. Recently, [21] combined lane detection with vanishing point prediction task to enhance the learning of context information. [42] propose Spatial CNN (SCNN) to learn the spatial relationship in such long structure. Our network shows the end-to-end leaning abilities of the previous CNN models, with the distinction that we explicitly design a structured output to address the topological errors problems.

2.4 Linear Structure Detection

Some methods have been proposed for linear structure detection, such as roads in an aerial image and cell membranes in an electron microscopy image. These problems differ from ours as the linear structures generally have amorphous spatial extent, while both lane detection and retinal layer segmentation tasks target at instance-level segmentation. For linear structure detection, [40] uses a CRF formulation whose priors are computed on higher-order cliques of connected superpixels likely to be part of road-like structures. Another approach to model higher-level statistics is to represent linear structures as a sequence of short linear segments, which can be accomplished using a Marked Point Process [2]. However, it requires computationally expensive inference formulates as Reversible Jump Markov Chain Monte Carlo. [24] designs a topology loss that is aware of the higher-order topological features of linear structures. It encourages topology coherent prediction results but does not guarantee it, as all other pixel-based segmentation methods do.

3 METHOD

We propose a novel deep convolutional network, StripNet, for segmentation of long and continuous strip patterns. It decomposes the original segmentation problem into more easily solved local boundary prediction problems, while preserving topology consistency by the structured outputs. Our network follows a coarse-to-fine philosophy, which consists of two stages: *rough strip localization* and *precise boundary regression*.

Specifically, *rough strip localization* separates the whole image into segments vertically or horizontally, and locate the strip structure in each segment roughly. Then *precise boundary regression* regress the boundary of strip in each segment precisely. The main architecture of our models are shown in Fig. 2.

For illustration, we first introduce how StripNet works for the retinal layer segmentation task and then tell the difference of two tasks and adapt StripNet for lane detection. Sec. 3.1 describes the procedure and settings of *rough strip localization*, and a detailed description is given in Sec. 3.2to introduce *precise boundary regression*. The post processing is mentioned in Sec. 3.3. And Sec. 3.4 tells the difference and specific adaption for lane detection task.

3.1 Rough Strip Localization

Rough strip localization aims to locate the whole strip region in a coarse way. It separates the whole image into segments vertically or horizontally, and locate the strip structure in each segment roughly, that is, to locate the boundary of RoI that could cover the whole retinal layer for each segment. For a specific task, we need to identify a direction (vertical or horizontal) for predicting the structured output, depending on the overall orientation of strip structures. As the retinal layer distributes horizontally, StripNet predicts the sequence of outputs in a horizontal direction. The whole image is thus uniformly partitioned into fixed-width (e.g., 16 pixels in this paper) segments, and the predictions will be made per segment.

In this stage, StripNet only predicts the up and down boundaries of all the retinal layers as a whole. This is based on the observation and experiments before that directly predicting the precise location of each retinal layer is prone to errors, as these layers only occupy a small portion of the slice and may be affected by distracting noise. Therefore, we adopt the heatmap for training the network, which is inspired by the deep pose estimation methods [9, 27, 41]. For pose estimation, the network is trained to predict the location of body joints on specifically designed heatmaps as shown in Fig. 3. We generate the ground truth maps G_u and G_d by convolving a vertical Gaussian kernel g with the up and down binary boundary map B_u and B_d of the image, respectively

$$G_u(v) = g * B_u(v), v = 1, 2, ..., V \tag{1}$$

$$G_d(v) = g * B_d(v), v = 1, 2, ..., V \tag{2}$$

$$g(p) = \exp\left(-\frac{p^2}{2\sigma^2}\right) \tag{3}$$

where σ is the variance of the Gaussian kernel, and we fix $\sigma = 8$ in our experiments. Both G and B are 16× down-sampled.

We take the celebrated deep model VGG16 [3] as our backbone network, and place two 1×1 convolution layers on top of the 16× down-sampled $conv5\_3$ maps to generate the score maps for regression. Batch normalization and ReLU units are adopted and placed after each convolution layer. A sigmoid layer is applied to transform the scores to the range of 0 to 1. And we adopt the L_2 loss for training.

With the predicted heatmap, we can obtain a RoI for each column on the heatmap that contains the retinal layers without too much background noise. Let the i-th RoI be defined by its top and bottom coordinates (h_i^0, h_i^1). We first identify the locations of the highest response at each column in the predicted heatmaps P_u and P_d,

$$\begin{cases} r_u(i) = \arg\max_j \{P_u(j,i)\}, \\ r_d(i) = \arg\max_j \{P_d(j,i)\}. \end{cases} \tag{4}$$

To compensate for the inaccuracies of the heatmaps and ensure that all the retinal layers are included in the RoI, we enlarge the search region by a constant η.

$$\begin{cases} h_i^0 = r_u(i) - \eta, \\ h_i^1 = r_d(i) + \eta. \end{cases} \tag{5}$$

After obtaining the above boundaries for each column, useful features in RoI are extracted for *precise boundary regression* to regress precise boundary between retinal layers.

3.2 Precise Boundary Regression

This stage precisely regress the boundaries between retinal layers. In order to achieve that, we concentrate in the RoI obtained from the former stage, where much noise has been thrown before. In the same manner described in Sec. 3.1, we predict the heatmaps to identify the boundaries between any two neighboring retinal layers. Suppose there are N retinal layers to be segmented, then we have $N-1$ internal boundaries and two up and down boundaries, which

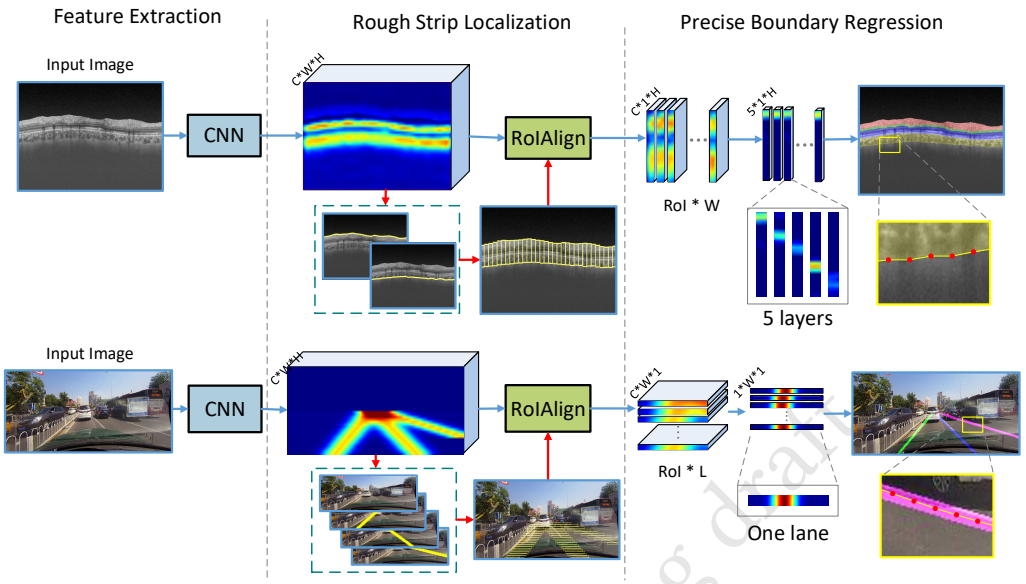

Figure 2: Framework of the proposed StripNet. It first regress heatmap to predict the RoI that fully cover the strip structures in each columns or rows in *rough strip localization* stage, then precisely regress the boundary for each strip in *precise boundary regression* stage.

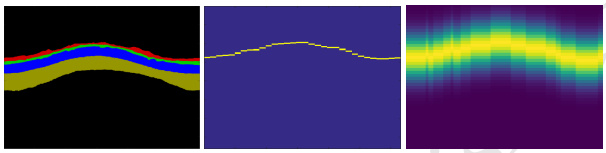

Figure 3: Comparisons between the label map, up boundary map and up heatmap.

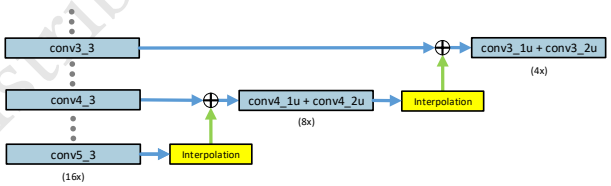

Figure 4: Upsampling architecture for encoding more detailed information.

sums up to $N+1$ boundaries in total. We generate the ground truth R_t for the t-th map as

$$R_n = g * B_n, n = 1, 2, ..., N+1, \quad (6)$$

where B_t denotes the t-th boundary map.

For each column in feature maps, RoIAlign [14] is used to extract and resize the features to feature vectors in a fixed height. Then we adopt the same architecture as the one regressing the up and down boundaries. Two $1 * 1$ convolution layers with batch normalization and ReLU Units and a sigmoid layer are placed on top of RoIAlign layer to generate the score maps.

We observe that in precise boundary regression, since we have to map the boundary to a fixed vector, if the target length is too short, we may not get precise boundary results. Therefore, we enlarge the height of feature vector to 200 pixels, so that we can get a dense regression result. RoIAlign layer is adopted for this purpose by adapting bilinear interpolation on the connected feature map. Moreover, the sampling ratio determines the up limit of quantization errors. So in order to decrease the sampling ratio of the feature map, we adopt an extra upsampling architecture which is inspired by U-net structure as shown in Fig. 4. We apply bilinear interpolation to *conv*5_3 feature maps and concatenate it with *conv*4_3, then two 3×3 convolution layers are exploited to fuse the feature maps. This architecture results in twice resolution, and furthermore provides multiscale features. We apply the same operation to *conv*3_3 and get 4× sampling ratio finally.

3.3 Post Processing

In a standard one-way Gaussian peak, the center value may be extremely close to its 2 direct neighbors. Such small differences raise the difficulty for locating the center, thus leading to a minor deviation if we locate the max value of the score map directly. Moreover, the *L*2 loss function only guarantees the similarity between the prediction and the ground truth map, which is a Gaussian peak, but makes little supervision on the position of the top value. Inspired by the geometric characteristic of the loss function, we propose a method to choose the target position accurately by Sliding Gaussian Peaks (SGP). We slide a one-way Gaussian peak with the same σ as

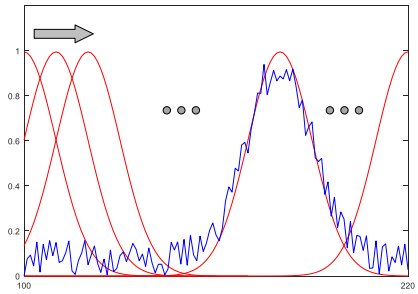

Figure 5: Method of Sliding Gaussian peaks. The red lines are standard Gaussian peaks and the blue line is the predicted score vector. Note how sliding Gaussian kernel generates more robust results avoids picking the local maximum response locations.

training process along the score map and calculate the $L2$ distance between the two vectors as shown in Fig. 5.

The final predicted coordinate is where the $L2$ distance reaches to the minimum:

$$R_N = \underset{t=\rho,\rho+1,\ldots,T}{\arg\min} \{||s - g_t||\} \tag{7}$$

where s is the $T \times 1$ score vector output from the network and R is the final regression coordinate. g_t is a vector with a one-way Gaussian kernel placed in t. ρ is a parameter which aims to prevent boundary disorder, and is set to 1 for $N = 1$ and R_{N-1} for $N \geq 2$. In this way, the supervision of the loss function is exploited completely, and the final prediction is determined by the whole score map.

To obtain the final results, we first compute the exact locations of the boundaries regressed in *precise boundary regression*, which gives us n equidistant points discrete in y-axis. We connect these points in order, and 5 fold line are generated from up to down as the prediction of the 5 boundaries and the final segmentation results of 4 tissues are obtained. By doing so, we ensure that neither holes nor isolated areas could appear in the segmentation results, since we assign areas between the same boundaries belong to the same layer. Because one boundary only correspond to one connected line, no fault could appear, and the topological constraint is fulfilled.

3.4 StripNet for Lane Detection

As shown in Fig. 1, each lane only covers a small range of the horizontal direction, while they almost appear at the same rows. So it is natural to adapt StripNet to predict the sequence of output in a vertical direction. Besides, since the lanes distribute with large gaps between each other, we adapt StripNet to localize each lane separately. To be more specific, we predict two heatmaps of the left and right boundaries for each lane, respectively, instead of two heatmaps for all lanes together. The ground truth maps are generated in the same way as mentioned in Sec. 3.1.

Unlike OCT images, there can be no lane markings in the picture due to occlusion or any other reasons, although lanes may still exist and need to be predicted, with number varies from zero to four, (for example in Fig. 1 is three), while retinal layers exist in a confirmed number. Unlike OCT, lanes do not always go through from one side to the other. Therefore, in lane detection, for each RoI, a score is also predicted to suggest whether there is a lane segment. Since the lane has a fixed width, we change to predict the centers of the lanes. With these subtle modifications, we can get the slope and location of the line easily and precisely.

Only when the score of a RoI is greater than a threshold, e.g., 0.5, will the heatmap be further processed. Otherwise the RoI is treated as a non-lane area. By the argmax operation for each row of one heatmap, the exact relative locations of the left and right boundaries can be acquired. Given the left and right location, using the topology prior, the lane segments in each bounding box are treated as a straight line, due to the small height of the bounding box and that lane segments usually lie through the box from up to down. These centers are connected directly to get the final output.

4 EXPERIMENTS

In this section, we conduct experiments on both OCT segmentation and Lane detection task to evaluate the proposed StripNet.

4.1 Data and implementation details

We first evaluate StripNet on our self-collected OCT dataset. The dataset includes a total of 1,202 DRI-OCT (Atlantis, Topcon, Tokyo, Japan) with 579 normal people and 605 glaucoma patients and 202 Spectralis (Heidelberg, Germany) glaucoma patients. These circular scans targeted at the center of optic disc with diameters of 3.5mm and 3.4mm respectively. These images are all manually delineated by three doctors, and each image is at least labeled by two doctors. We ask a senior doctor to visually inspect the labeling results and choose the better ones as the final label maps. A total of 4 retinal layers are labeled, including RNFL, GCC, Retina and Choroid. We split all DRI-OCT scans into 1051/151 for training and testing respectively, and no patient is included simultaneously in both sets. The StripNet is only trained on DRI-OCT images while tested on both DRI-OCT and Heidelberg Spectralis images.

The training process is divided into two phases. We first feed the ground truth of the rough prediction as the input of the RoIAlign layer and the precise regression block for training the later part of the network. Then we use the prediction of the rough prediction in replace of ground truth for joint training of both stages. We adopt the stochastic gradient descent for optimization with batch size 1. We train the whole network for 50 epochs, using a decreased learning rate from 10^{-5} to 10^{-7} by reducing learning rate by 0.1 when training for 30, 40 and 45 epochs. The VGG16 model is pre-trained on a large-scale dataset ImageNet for image classification. The whole framework is implemented with the caffe library.

4.2 Ablation Study

In this chapter, extensive experiments are conducted verify the effectiveness of each component in StripNet.

4.2.1 Evaluation of Rough Strip Localization. The first part of StripNet aims at giving a rough while robust location of the retinal layer, which needs to cover the total retina layers roughly. So we conduct an experiment to assess its performance by comparing it

with a widely used regression strategy which regresses the normalized coordinates directly. We implement the coordinate regression method by placing two convolution layers after $conv5_3$ feature map to produce 2 coordinates of the up and down boundaries in each column. The first convolution layers are set to 7×1 kernel with 5×1 stride padding 1, and 9×1 kernel with stride 1 padding 0. The architecture outputs a $2 \times 1 \times 79$ score map which denotes the normalized coordinate offset of the up and down boundaries in 79 columns. We compare these methods by referring to the performance of precise regression architecture. For fair comparison, these two experiments are trained in common and shares parameters from $conv1_1$ to $conv5_3$. The rough localization architecture is set with 16× sampling ratio and RoIAlign layer extracts feature vector with 120 in height. For testing, to ascertain the up limit of the rough localization, a ground truth group is also added into comparison, where we set the ground truth as the input of precise regression. Moreover, in order to deduct the error caused by rough localization, we utilize a method of expand the selection area of RoI to 16 pixels higher and this method is applied in the above experiments. All these comparisons are reported in Table. 1, which shows that our method outperforms the traditional coordinate regression method especially in RNFL and Choroid layer, while the performance in GCC and Regina differs a little between all 3 experiments. This is because that RNFL and Choroid are at the top and bottom of the total layer, thus they are more sensitive to the error of boundary prediction than the other 2 inboard layers.

Table 1: Comparison between two methods of rough localization and the ground truth. The '+' marked group has expand their selection area to 16 pixels higher.

Method	RNFL	GCC	Retina	Choroid	mean
Coordinate	77.94	71.75	91.92	82.52	81.03
Gaussian Map	82.33	71.12	92.27	85.81	82.88
Ground Truth	87.16	73.63	92.72	88.58	85.52
Coordinate+	84.89	73.35	92.70	85.91	84.21
Gaussian Map+	85.76	73.68	92.83	86.68	84.74
Ground Truth+	86.48	73.70	92.70	87.67	85.14

We also observe that the application of selection expanding lifts the performance of our method but reduces the performance of the ground truth group. Because this operation improves the recall of the total layer, but leads to more background noise at the same time. So it also can be inferred that we should enlarge the selecting area of RoI to a right degree, neither too small nor too large. A set of experiments shows that 8 pixel is one of the compromised choices, and we use this setting in all the subsequent experiments.

4.2.2 Evaluation of Precise Boundary Regression. In this step, we target at regressing the position of the boundary in each RoI precisely. Firstly, we adapt RoI pooling to extract feature vector in each RoI. RoI pooling layer is designed for extracting features in various aspect ratios into a fix-sized rectangular. Specifically, it works by dividing the RoI into $\alpha \times \beta$ sub-windows and then max pool the features in each sub-window. However, the performance of RoI pooling is not good, because the rounding operation to the coordinate introduces misalignments between the RoI and the extracted feature maps, which leads to unexpected deviations especially in our tasks. Therefore, we adopt RoIAlign layer in replace of RoI pooling. RoIAlign layer uses bilinear interpolation to compute the exact values of the input features at four regularly sampled locations. It guarantees the spatial correspondence between the features and the images and is of vital importance in our tasks because both retinal layers and lanes are sensitive to small misalignments. The change from RoI pooling to RoIAlign brings large improvements as shown in Table. 2. Both experiments are positioned in 16× sampling ratio and extract feature map with 120 in height.

Table 2: The performance comparison of RoIAlign layer and RoI pooling layer.

Method	RNFL	GCC	Retina	Choroid	mean
RoI pooling	72.43	58.87	86.21	80.11	74.16
RoIAlign	84.91	73.20	92.89	86.89	84.47

As is illustrated in Sec. 3.3 , the final segmentation result benefits from the density degree of RoI and the height of extracted feature vectors. The density of RoI is fixed and determined by the size of the input image and the sampling ratio. So we change the sampling ratio for 3 stages, 16×, 8×, and 4×. The heights of feature vectors are selected as 40 ,120, 160. In this experiment, we adapt RoIAlign layer to extract feature maps in RoI. The results shown in Table. 3 confirms the point that in the same sampling ratio, the performance of the StripNet shows an overall upward trend as the height of extracted feature vector increases or the sampling ratio decreases. Moreover, for those in height 40, we observe that the 4× and 8× sampling group obtains similar performances. This is because in the two groups, the bottlenecks of raising performance are the height of the feature vectors.

Table 3: Experimental results of various sampling ratio and the target length of the feature vector after RoIAlign.

Rate	RoI height	RNFL	GCC	Retina	Choroid	mean
16	40	84.40	72.08	92.30	86.4	83.80
16	120	84.91	73.2	92.89	86.89	84.47
16	200	85.08	73.88	93.03	86.86	84.71
8	40	86.78	74.91	93.22	87.92	85.71
8	120	87.36	75.85	93.69	88.42	86.33
8	200	87.89	76.83	93.92	88.44	86.77
4	40	86.99	74.4	93.07	87.65	85.53
4	120	88.59	75.19	93.79	88.31	86.47
4	200	**89.65**	**76.91**	**94.05**	**89.12**	**87.43**

4.2.3 Evaluation of Sliding Gaussian Peak. This method slides the standard Gaussian peak along the score map and calculates $L2$ loss directly for each position as the final score, and then locate the position where has the minimum $L2$ loss as the final prediction. SGP is proposed as a more precise reprocessing step in replacement of the traditional method that locates the max value of the score map

roughly. Table. 4 shows that this method improves the performance of all layers in StripNet.

Table 4: Comparison between argmax and sliding Gaussian peaks for precise boundary prediction.

Method	RNFL	GCC	Retina	Choroid	mean
arg max	89.65	76.91	94.05	89.12	87.43
Sliding peak	90.02	78.17	94.46	89.25	87.98

4.2.4 Comparisons with Existing Methods. In this section, we compare StripNet with three deep models including Deeplab-v3 [4], U-net [32] and S-net [16]. All the three models are pretrained on COCO dataset. For fair comparison, all models are trained for 50 epochs with batch size 1 in Topcon DRI-OCT training set without any data augmentation. Moreover, the state-of-the-art Random Forest (RF) + graph method [39] is also added for comparison. It makes use of manual-crafted features and performs excellent on both Heidelberg Spectralis and Zeiss Cirrus images. We trained RF using randomly selected 56 images in training set with 60 trees and 10 subjects for each tree. As RF + graph is not designed for recognizing the bottom boundary Choroid in our dataset, we leave it unlabeled in Table. 5. Some examples are shown in Fig. 6.

Table 5: Comparison with state-of-art models on Topcon DRI-OCT test set.

Method	RNFL	GCC	Retina	Choroid	mean
Deeplab-v3 [4]	89.28	76.71	93.89	86.53	86.60
U-net [32]	88.06	77.24	94.15	87.55	86.75
S-net [16]	88.54	77.19	94.04	85.68	86.36
RF+graph [39]	86.19	69.72	90.75	-	-
StripNet,4×,200	**90.02**	**78.17**	**94.46**	**89.25**	**87.98**

Table. 5 shows that our StripNet outperforms all these deep models as well as the traditional RF+Graph method. Traditional deep models have no restriction on the topology structure of the final precision, which may lead to boundary disorders, isolated areas or even holes, while the proposed StripNet ensures the consistency of topology. The RF+Graph method also avoids topology errors, but its manually-crafted features are largely effected by the variation of the input images, which leads to a deterioration in performance.

Then, we test these models on the Heidelberg Spectralis images to compare their generalization abilities between different manufacturers. The Spectralis OCT differs from DRI-OCT in many aspects such as scale, noise level and length-width ratio. The test results are shown in Table. 6 and visualizations are given in Fig. 6.

From the results we can see that StripNet has strong generalization abilities and obtains the best performance. This is because between two brands of OCT image, there are certain differences in the internal gray scale, but their topology structure is guaranteed and the characteristics of each tissue vary little. so the method of regression boundaries is more effective for segmentation. The utilization of the heat map and rough prediction architecture gives more position-sensitive guidances to the network, and decreases the influence of background noises. Moreover, the structured output guarantees the topology consistency, thus leading to the excellent performance in Spectralis OCT images. However, The traditional FCN-based deep segmentation models are more sensitive to the variation of the gray scale, thus causing a sharp decrease on final performance. For Deeplab-V3, the specially designed atrous convolution layer enhances the ability of segment objects at multiple scale. But in this test set, the atrous convolution introduces more noise and got the worst performance. The RF+Graph method also shows the topology-correct segmentation results, but still performs unsatisfactory. However, StripNet aims at regressing the boundaries and pays more attention to the differences in characteristics of the adjacent organizations, thus offsets the noise from the variations of inputs. Then, the structured output confirms the consistency of the topology, and thus performs excellent in Spectralis images. .

Table 6: Comparison with state-of-art models on Heidelberg Spectralis images.

Method	RNFL	GCC	Retina	Choroid	mean
Deeplab-v3 [4]	15.89	23.96	48.85	40.88	32.39
U-net [32]	86.79	60.22	87.31	75.80	77.53
S-net [16] [16]	84.36	59.56	77.98	70.64	73.14
RF+graph [39]	83.53	68.09	77.39	-	-
StripNet,4×,200	**91.46**	**79.65**	**93.51**	**87.48**	**88.03**

4.3 Lane detection

4.3.1 Data and Implementation Details. The precise regression architecture is pretrained firstly on CULane dataset for segmentation, using standard SGD with batch size 12, basic learning rate 0.01, momentum 0.9, and weight decay 0.0001. The policy for learning rate is 'poly' with power and iteration number set to 0.9 and 60k, respectively. Then the rough prediction architecture is added for joint end-to-end training using the same strategy with iteration number of 30k.

We evaluate StripNet on CULane dataset [42], which is currently the biggest lane detection dataset including 8 challenging scenarios. And these scenarios account for 72.3 % of the dataset. For evaluation, the lane markings are viewed as lines with widths of 30 pixels and the intersection-over-union (IoU) is calculated between the ground truth and the prediction. Predictions whose IoUs are larger than certain threshold are viewed as true positives (TP), and the threshold is 0.5 for strict evaluations. Then F1-measure is employed to evaluate methods' performance on CULane datasets.

4.3.2 Comparison with state-of-art methods. To verify the effectiveness of StripNet in lane detection, we compare it with several methods: ResNet-50 (Baseline) model, SCNN, SCNN++, SCNN-StripNet and Res50-StripNet. Our ResNet-50 (baseline) model is modified based on the LargeFOV model [6]. We modify the stride in '*conv*4_1' of ResNet-50 [15] to 1 to change the resolution of the feature map to be 8× downsampled. The SCNN is released by [42], which performs best in CULane dataset. To verify whether the improvement of StripNet is brought by simply adding more model

Table 7: Comparison with other methods, with IoU threshold=0.5. For crossroad, only False Positive (FP) are shown

Category	Normal	Crowded	Dazzle light	Shadow	No line	Arrow	Curve	Crossroad	Night	Total
ResNet-50 [15]	86.1	64.2	53.5	59.7	36.9	78.1	62.3	2092	59.7	66.2
Res50-StripNet	86.7	65.3	55.5	66.6	39.2	79.7	63.9	2468	61.4	67.4
SCNN [42]	90.6	69.7	58.5	66.9	43.4	84.1	64.4	1990	66.1	71.6
SCNN++	90.7	69.7	58.9	69.7	44.1	84.9	64.9	**1891**	65.9	71.9
SCNN-StripNet	**90.8**	**69.9**	**60.0**	**69.7**	**44.5**	**85.3**	**66.1**	2020	**66.9**	**72.2**

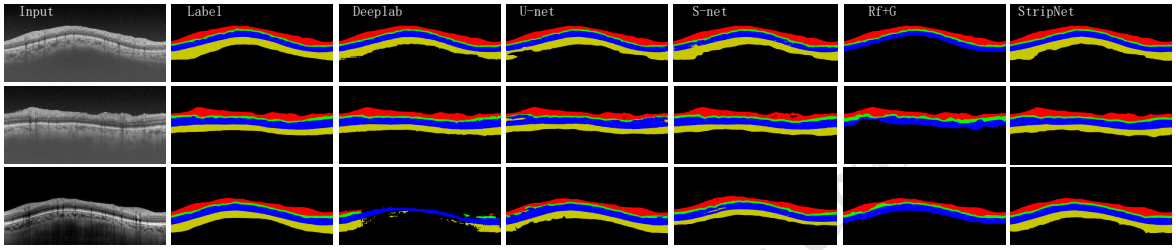

Figure 6: Comparisons between results of FCN-based models and StripNet. The first two rows are DRI-OCT images, and the third row is Spectralis image.

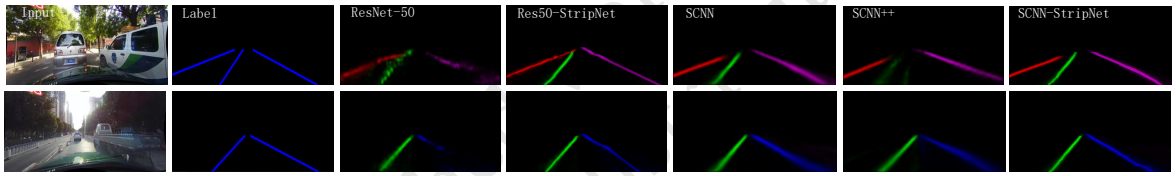

Figure 7: Comparisons between lane detection results of ResNet, Res50-StripNet, SCNN, SCNN++ and SCNN-StripNet

parameters, we replace the original upsample layer with stride 8 to 2 deconvolution layer with stride 2 to get a deeper SCNN, named SCNN++. We test our method using ResNet-50 (Baseline) and SCNN, which are Res50-StripNet and SCNN-StripNet respectively. To get a finer regression output, we upsample the 36*100 feature map obtained in the former stage to 72*200. And we segment the feature map into 36 slices horizontally, predict four boxes in each slice for each lane, which is totally 144. We draw a lane segment based on the heatmap if the score threshold is greater than 0.5. All experiments are implemented on the Torch7. The test results on different challenging scenarios are shown in Table. 7.

The Baseline result is consistent with the result of ResNet-50 shown in [42]. And the SCNN result is the same as [42], which is previously the best result in CULane dataset. From the results, we can see that increasing parameters to get a deeper network brings little improvement, while our method improves both in ResNet-50 and SCNN. This indicates StripNet's generalization ability across different backbone models. What is more, our method outperforms other methods especially in shadow or dazzle light cases, where FCNs are faced with topological errors due to dark or reflective circumstances. The comparison examples in shadow and dazzle light scenarios are shown in Fig. 7, holes, isolated islands appear in outputs of conventional FCNs, but StripNet avoid this problems due to good use of topological constraints.

5 CONCLUSIONS

In this paper, we propose StripNet to segment long and continuous strip patterns in different image modalities. StripNet avoids to make topological segmentation errors by specially the structured output, which decomposes the original segmentation problem into more easily solved boundary-regression problems, in a coarse-to-fine manner. The experimented results show that StripNet achieves state-of-the-art performance in both retinal layer segmentation and lane detection tasks, and has good generalization abilities across datasets and backbone architectures.

REFERENCES

[1] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. 2012. Semantic segmentation with second-order pooling. In *Proc. ECCV*.
[2] Dengfeng Chai, Wolfgang Förstner, and Florent Lafarge. 2013. Recovering line-networks in images by junction-point processes. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 1894–1901.
[3] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. (2014).
[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR*

abs/1706.05587 (2017). arXiv:1706.05587 http://arxiv.org/abs/1706.05587
[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2015. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *Proc. ICLR*.
[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915* (2016).
[7] KY Chiu and SF Lin. 2005. Lane detection using color-based segmentation. *WOS:000235518700117* (2005). https://ir.nctu.edu.tw/handle/11536/17998
[8] François Chollet. 2016. Xception: Deep Learning with Depthwise Separable Convolutions. *CoRR* abs/1610.02357 (2016). arXiv:1610.02357 http://arxiv.org/abs/1610.02357
[9] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. Structured Feature Learning for Pose Estimation. *CoRR* abs/1603.09065 (2016). arXiv:1603.09065 http://arxiv.org/abs/1603.09065
[10] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. 2013. Learning hierarchical features for scene labeling. *TPAMI* 35, 8 (2013), 1915–1929.
[11] Mona Kathryn Garvin, Michael David Abramoff, Xiaodong Wu, Stephen R Russell, Trudy L Burns, and Milan Sonka. 2009. Automated 3-D intraretinal layer segmentation of macular spectral-domain optical coherence tomography images. *IEEE transactions on medical imaging* 28, 9 (2009), 1436–1447.
[12] Raghuraman Gopalan, Tsai Hong, Michael Shneier, and Rama Chellappa. 2012. *A Learning Approach Towards Detection and Tracking of Lane Markings.* Technical Report. IEEE Transactions on Intelligent Transportation Systems.
[13] Bei He, Rui Ai, Yang Yan, and Xianpeng Lang. 2016. Accurate and robust lane detection based on dual-view convolutional neutral network. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*. IEEE, 1041–1046.
[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. 2017. Mask R-CNN. *CoRR* abs/1703.06870 (2017). arXiv:1703.06870 http://arxiv.org/abs/1703.06870
[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[16] Yufan He, Aaron Carass, Yeyi Yun, Can Zhao, Bruno M. Jedynak, Sharon D. Solomon, Shiv Saidha, Peter A. Calabresi, and Jerry L. Prince. 2017. Towards Topological Correct Segmentation of Macular OCT from Cascaded FCNs. (2017).
[17] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. 2015. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716* (2015).
[18] Claudio Rosito Jung and Christian Roberto Kelber. 2004. A robust linear-parabolic model for lane following. In *Computer Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*. IEEE, 72–79.
[19] Jihun Kim and Minho Lee. 2014. Robust lane detection based on convolutional neural network and random sample consensus. In *International Conference on Neural Information Processing*. Springer, 454–461.
[20] Andrew Lang, Carass Aaron, Hauser Matthew, Elias S Sotirchos, Peter A Calabresi, Howard S Ying, and Jerry L Prince. 2013. Retinal layer segmentation of macular OCT images using boundary classification. *Biomedical Optics Express* 4, 7 (2013), 1133–1152.
[21] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. 2017. VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition. In *The IEEE International Conference on Computer Vision (ICCV)*.
[22] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. 2015. Semantic image segmentation via deep parsing network. In *Proc. ICCV*.
[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3431–3440.
[24] Agata Mosinska, Pablo Marquez-Neila, Mateusz Kozinski, and Pascal Fua. 2017. Beyond the Pixel-Wise Loss for Topology-Aware Delineation. *arXiv preprint arXiv:1712.02190* (2017).
[25] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool. 2018. Towards End-to-End Lane Detection: an Instance Segmentation Approach. *ArXiv e-prints* (Feb. 2018). arXiv:cs.CV/1802.05591
[26] Jelena Novosel, Koenraad A. Vermeer, Gijs Thepass, Hans G. Lemij, and Lucas J. Van Vliet. 2003. Loosely coupled level sets for simultaneous 3D retinal layer segmentation in optical coherence tomography. In *Simulation Conference, 2003. Proceedings of the*. 59–65.
[27] Tomas Pfister, James Charles, and Andrew Zisserman. 2015. Flowing ConvNets for Human Pose Estimation in Videos. *CoRR* abs/1506.02897 (2015). arXiv:1506.02897 http://arxiv.org/abs/1506.02897
[28] Pedro H. O. Pinheiro and Ronan Collobert. 2014. Recurrent Convolutional Neural Networks for Scene Labeling. In *Proc. ICML*.
[29] Xiaojuan Qi, Jianping Shi, Shu Liu, Renjie Liao, and Jiaya Jia. 2015. Semantic Segmentation With Object Clique Potential. In *Proc. ICCV*.
[30] Fabian Rathke, Stefan Schmidt, and Christoph Schnörr. 2014. Probabilistic intra-retinal layer segmentation in 3-D OCT images using global shape regularization. *Medical image analysis* 18, 5 (2014), 781–794.
[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Proc. NIPS*.
[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 234–241.
[33] Abhijit Guha Roy, Sailesh Conjeti, Sri Phani Krishna Karri, Debdoot Sheet, Amin Katouzian, Christian Wachinger, and Nassir Navab. 2017. ReLayNet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks. *Biomedical optics express* 8, 8 (2017), 3627–3642.
[34] Alexander G Schwing and Raquel Urtasun. 2015. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351* (2015).
[35] Abhishek Sharma, Oncel Tuzel, and David W Jacobs. 2015. Deep Hierarchical Parsing for Semantic Segmentation. *Proc. CVPR* (2015).
[36] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[37] Ben Southall and Camillo J Taylor. 2001. Stochastic road shape estimation. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 1. IEEE, 205–212.
[38] Zhu Teng, Jeong-Hyun Kim, and Dong-Joong Kang. 2010. Real-time Lane detection by using multiple cues. In *Control Automation and Systems (ICCAS), 2010 International Conference on*. IEEE, 2334–2337.
[39] Chuang Wang, Yaxing Wang, Djibril Kaba, Zidong Wang, Xiaohui Liu, and Yongmin Li. 2015. Automated Layer Segmentation of 3D Macular Images Using Hybrid Methods. In *International Conference on Image and Graphics*. 614–628.
[40] Jan D Wegner, Javier A Montoya-Zegarra, and Konrad Schindler. 2013. A higher-order CRF model for road network extraction. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 1698–1705.
[41] Shih En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *Computer Vision and Pattern Recognition*. 4724–4732.
[42] Pan Xingang, Shi Jianping, Luo Ping, Wang Xiaogang, and Tang Xiaoou. 2018. Spatial As Deep: Spatial CNN for Traffic Scene Understanding. In *AAAI Conference on Artificial Intelligence (AAAI)*.
[43] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2881–2890.