StripNet: Towards Topology Consistent Strip Structure Segmentation

Guoxiang Qu[1,2,3†], Wenwei Zhang[4,5†], Zhe Wang[4†], Xing Dai[4], Jianping Shi[4], Junjun He[1,2], Fei Li[6], Xiulan Zhang[6], Yu Qiao[1,2*]
[1]Guangdong Key Lab of Computer Vision and Virtual Reality, [2]SIAT-Sensetime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[3]Sino-Dutch Biomedical and Information Engineering School, Northeastern University, Shenyang, China
[4]Sensetime Group Limited, Beijing, China
[5]Computer Science School, Wuhan University, Wuhan, China
[6]Zhongshan Ophthalmic Center, State Key Laboratory of Ophthalmology, Sun Yat-Sen University Guangzhou, China
gx.qu@siat.ac.cn,{zhangwenwei,wangzhe,daixing,shijianping}@sensetime.com,hejunjun@sjtu.edu.cn,lifei28@mail2.sysu.edu.cn,zhangxl2@mail.sysu.edu.cn,yu.qiao@siat.ac.cn

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 strip 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, and takes account of 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

• **Computing methodologies** → **Computer vision**; **Neural networks**; • **Theory of computation** → *Structured prediction*;

KEYWORDS

Strip Segmentation, Lane Detection, Retinal Layer Segmentation

ACM Reference Format:
Guoxiang Qu, Wenwei Zhang, Zhe Wang, Xing Dai, Jianping Shi, Junjun He, Fei Li, Xiulan Zhang, Yu Qiao. 2018. StripNet: Towards Topology Consistent Strip Structure Segmentation. In *2018 ACM Multimedia Conference (MM '18), October 22–26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, Article 4, 9 pages. https://doi.org/10.1145/3240508.3240553

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, such as blood vessels, medical tomographic images, tree trunks, road lines in satellite maps, as well as retinal layers in OCT images and lanes on the roads, which are shown in Fig. 1 (a). Analysing these structures is always an important yet challenging 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 for autonomous driving.

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, 25, 35] is to classify each pixel independently into one of the predefined categories. These pixel-level segmentation methods naturally encounter the topological inconformity problem, as shown in Fig. 1 (b) [33, 43].

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 divides 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

*The corresponding author.
†Contributed equally to the paper.

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240553

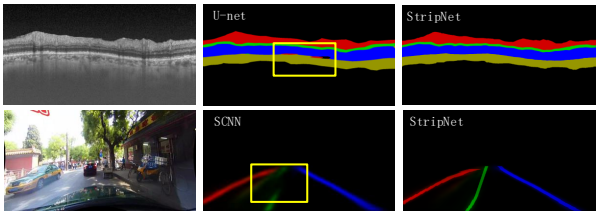

Figure 1: Examples of two strip structures and their segmentation results by previous methods and our proposed method StripNet. (a) retinal layers in OCT images and lanes on the road. (b) Results of previous methods (U-net [33] and SCNN [43]). (c) Results of StripNet. Note conventional FCN based methods exhibit topological errors (highlighted by rectangles), while the proposed StripNet could avoid topological inconformity problems.

row. By doing so, at most one connected component of a strip will be obtained, and the strip can be constrained by the boundary.

The strip structures may only occupy a small portion of the image and are difficult to directly predict its locations from whole divided 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 bottom 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 can clean out many other unrelated areas that may distract the prediction, which is helpful 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. However, heatmap regression can reflect the distribution of objects in regions more directly and precisely. And this proves to be more stable and accurate than directly predicting coordinates in [9, 28] or using the anchor mechanism [32]. After that, we connect the points obtained in *precise boundary regression* that belong to the same boundary, and arrange areas between same boundaries of 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 and representative tasks, i.e., retinal layer segmentation in OCT images and lane detection. Extensive experiments on self-collected dataset (for retinal layer segmentation) and publicly available dataset (for lane detection) show that our method has good generalization ability and outperforms the state-of-the-art approaches on both tasks, without suffering 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 to this field [1, 5, 6, 24, 30]. 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) to improve the smoothness of the prediction. Pinheiro et al [29] introduced a Recurrent Neural Network (RNN) to recurrently refine its predictions by concatenating the RGB image with its predicted masks as input. Both [10, 29] are patch-based deep models, which are redundant in computation and time-consuming. In 2015, [25] 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 dramatically improves the efficiency. Since then, a lot of FCN-based works [6, 33, 36, 44] are proposed and further boost the performance of semantic segmentation. [33] used skip-connections between lower layers and higher layers to add more detailed information for the fine resolution prediction. [6] proposed 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, 22, 27, 27, 31]. [22, 27] exploited random forest and level set to produce accurate boundaries of retinal layers in B-scan OCT images. For the segmentation of 3-D OCT images, a graph-theoretic method is proposed by [11], and [31] presents a novel probabilistic approach

and achieves impressive results. In recent years, some deep learning approaches [16, 34] 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, 19, 38, 39] 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, 37] and segmentation [6, 25], neural networks were introduced to tackle the lane detection problem [12, 13, 17, 21, 23]. At first the CNN was used as feature extractor [12] or for image enhancement [21]. 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, [23] combined lane detection with vanishing point prediction task to enhance the learning of context information. [43] proposed Spatial CNN (SCNN) to learn the spatial relationship of 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, [41] 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. [26] 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* regresses 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. 3.2 to 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 bottom 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, 28, 42]. For pose estimation, the network is trained to predict the location of body joints on specifically designed heatmaps as shown in Fig. 3. Let V be the number of RoIs in the image. We generate the ground truth maps G_u and G_b by convolving a vertical Gaussian kernel g with the up and bottom binary boundary map B_u and B_b of the image.

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

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

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

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 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 map 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.

On the basis of the predicted heatmap, we can obtain an RoI for each column which contains the retinal layers with less 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

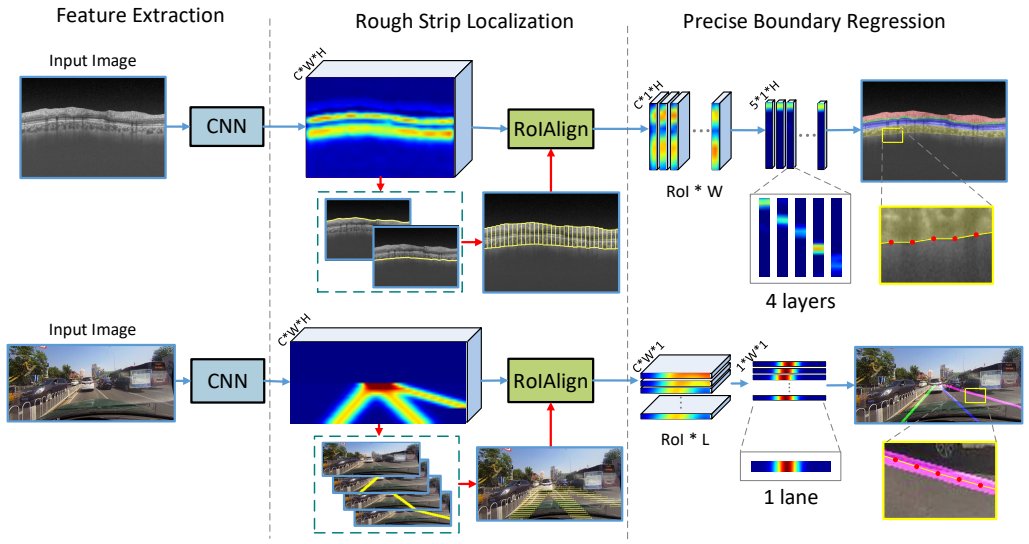

Figure 2: Overview of StripNet. It first regresses heatmap to predict the RoI that fully covers the strip structure in each columns or rows in *rough strip localization* stage, then precisely regresses the boundary for each strip in *precise boundary regression* stage.

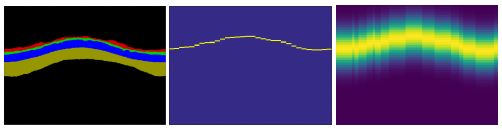

Figure 3: Comparisons between the label map, up boundary map and up heatmap.

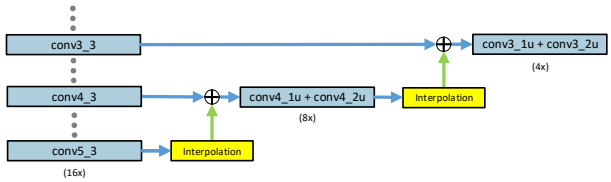

Figure 4: Upsampling architecture for encoding more detailed information.

each column in the predicted heatmaps P_u and P_b,

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

Moreover, inaccurate predictions of rough localization may mislead the search region of next step. To ensure all retinal layers included in the RoI, we enlarge the search region by a constant η. After that, features in RoI are extracted for regressing precise boundaries.

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

3.2 Precise Boundary Regression

This stage precisely regresses the boundaries between retinal layers. In order to achieve that, we concentrate in the RoI obtained from the former stage. In the same manner described in Sec. 3.1, we predict the heatmaps to identify the boundaries between any two neighboring retinal layers. Let there be N retinal layers to be segmented, then we have $N-1$ internal boundaries and two up and bottom boundaries. We generate the ground truth R_n for the n-th map as

$$R_n = g * B_n, n = 1, 2, ..., N+1, \tag{6}$$

where B_n denotes the n-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 bottom boundaries. Two $1*1$ convolution layers with batch normalization and ReLU Units and one sigmoid layer are placed on top of RoIAlign layer to generate the score map.

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 in order to 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 as shown in Fig. 4. We apply bilinear interpolation to feature maps of $conv5\_3$ and concatenate it with that of $conv4\_3$, then two 3×3 convolution layers are exploited to fuse the feature maps. This architecture results in twice up-sampled feature maps. We apply the same operation to $conv3\_3$ and obtain $4\times$ down-sampling ratio finally.

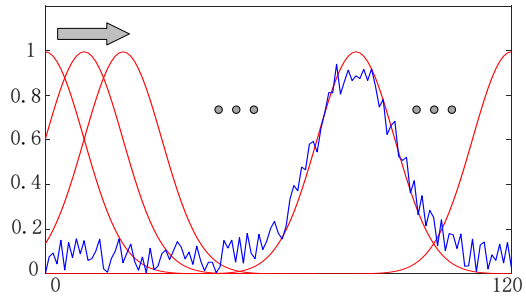

Figure 5: Method of Sliding Gaussian peaks. The red lines are standard Gaussian curves and the blue line is the predicted score vector. Note how sliding Gaussian kernel generates more robust results avoids picking the local maximum response locations.

3.3 Post Processing

To robustly obtain the boundary location and avoid drifting due to small prediction errors, we do not directly take the maximum response location as result. Instead, we propose a method to choose the boundary location accurately by Sliding Gaussian Peaks (SGP). We slide a one-way Gaussian peak with the same σ as training process along the score map and calculate the $L2$ distance between the two vectors as shown in Fig. 5, and the result is then obtained at the location with maximum response,

$$r_n = \underset{t=\rho,\rho+1,\ldots,T}{\arg\max} \{s_n * g\}_t \tag{7}$$

where s_n is the $T \times 1$ score vector output from the network and r_n is the coordinate for the n-th boundary. ρ is an index for preventing boundary disorder, and is set to 1 for $n = 1$ and r_{n-1} for $n \geq 2$.

From the RoI position and the obtained boundary coordinates, we can calculate a boundary map for each boundaries using nearest-neighbour upsampling. The segmentation mask can be easily obtained with the boundary maps. By doing so we ensure that neither holes nor isolated areas could appear in the segmentation results.

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 an 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 images (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 train *rough strip localization* and *precise boundary regression* separately, where they share low-level features and the precise regression network takes the ground truth boundaries as input. Then after several epoch's optimization, the precise regression network switches to take the prediction results of *rough strip localization* as inputs for joint optimization. We adopt the stochastic gradient descent for optimization with batch size 1. We train the whole network for 50 epochs, with a decreased learning rate from 10^{-5} to 10^{-7} by reducing learning rate by 0.1 every 10 epochs. The VGG16 model is pretrained on a large-scale dataset ImageNet for image classification. The whole framework is implemented by caffe [18].

4.2 Ablation Study

In this chapter, extensive experiments are conducted to 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. 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 bottom boundaries in each column. This network outputs a 2×79 score map which denotes the normalized coordinate offset of the up

and bottom 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 expanding 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 expanded their selection area to 16 pixels higher.

Method	RNFL	GCC	Retina	Choroid	mean
Coordinate	77.9	71.8	91.9	82.5	81.0
Gaussian Map	82.3	71.1	92.3	85.8	82.9
Ground Truth	87.2	73.6	92.7	88.6	85.5
Coordinate+	84.9	73.4	92.7	85.9	84.2
Gaussian Map+	85.8	73.7	92.8	86.7	84.7
Ground Truth+	86.5	73.7	92.7	87.7	85.1

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 simultaneously. So we should enlarge the selecting area of RoI to a suitable degree to make a balance. A set of experiments showed that a length of 8 pixels 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 extracts 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 ideal, because the rounding operation to coordinate introduces misalignments between RoI and extracted feature maps, while strip structures are thin and vulnerable to such nearest neighbor downsampling. Therefore, we adopt RoIAlign layer in replace of RoI pooling, which uses bilinear interpolation to compute the exact values of the input features at four regularly sampled locations. It guarantees the spatial correspondence between features and images, which 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	72.4	58.9	86.2	80.1	74.2
RoI Align	84.9	73.2	92.9	86.9	84.5

As illustrated in Sec. 3.3 , the final segmentation result benefits from the denser sampling of RoIs and extracted feature vectors of higher resolutions. So we compared three sampling ratios, i.e., 16×, 8×, and 4×. The resolution of feature vectors is selected as 40, 120 and 160. In this experiment, we adapt RoIAlign layer to extract feature maps in RoI. The results shown in Table. 3 confirm the point that in the same sampling ratio, the performance of the StripNet shows an overall upward trend as the resolution 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.

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	84.4	72.1	92.3	86.4	83.8
16	120	84.9	73.2	92.9	86.9	84.5
16	200	85.1	73.9	93.0	86.9	84.7
8	40	86.8	74.9	93.2	87.9	85.7
8	120	87.4	75.9	93.7	88.4	86.3
8	200	87.9	76.8	93.9	88.4	86.8
4	40	87.0	74.4	93.1	87.7	85.5
4	120	88.6	75.2	93.8	88.3	86.5
4	200	**89.7**	**76.9**	**94.1**	**89.1**	**87.4**

4.2.3 Evaluation of Sliding Gaussian Peak (SGP). We perform sliding window of the Gaussian kernel along the score map and calculates *L2* distance directly for each position as the final score, and then obtain the position that has the minimum distance 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	89.7	76.9	94.1	89.1	87.4
Sliding peak	90.0	78.2	94.5	89.3	88.0

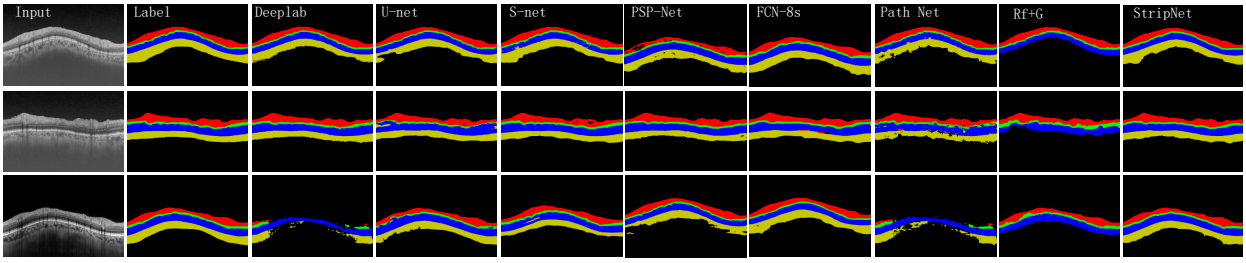

Figure 6: Comparisons between results of FCN-based models and StripNet. The first two rows are DRI-OCT images, and the third row is Spectralis image.

4.2.4 Comparisons with Existing Methods. In this section, we compare StripNet with some state-of-the-art deep models including Deeplab-v3 [4] , U-net [33] , PSPnet [33] , FCN [25] and S-net[16]. All models are pretrained on COCO dataset. We also compare with graphic methods like vectorization. [20] propose VectorNet to vectorize raster image of line arts, where PathNet is used for segmenting paths and OverlapNet is used to segment overlap regions, and the results are optimized by a MRF. Since our task is different from vectorization and we do not have overlap regions in our tasks, we only trained PathNet on OCT dataset with the same setting as ours, using the official code provided by the author.

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 [40] 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. Meanwhile, we quantify two types of topological errors, i.e., holes and isolated islands. To define the error, we first identify the largest connected component for each predicted layer, and count the others as holes or isolated islands. The average number of topological errors per image for each method is given in the last column of 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	Topo Err
Deeplab-v3 [4]	89.3	76.7	93.9	86.5	86.6	2.0
U-net [33]	88.1	77.2	94.2	87.6	86.8	2.1
S-net [16]	88.5	77.2	94.0	85.7	86.4	13.1
PSPnet [33]	88.4	77.1	94.2	86.5	86.6	6.8
FCN-8s [25]	87.7	74.6	93.1	84.8	85.0	0.7
PathNet[20]	88.4	76.2	93.9	84.1	85.7	20.5
RF+graph [40]	86.2	69.7	90.8	-	-	0
StripNet,4×,200	**90.0**	**78.2**	**94.5**	**89.3**	**88.0**	0

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.

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

Method	RNFL	GCC	Retina	Choroid	mean	Topo Err
Deeplab-v3 [4]	23.3	39.3	71.6	47.6	45.5	51.3
U-net [33]	85.7	63.6	88.5	76.6	78.6	17.3
S-net [16] [16]	84.9	66.6	78.4	70.8	75.2	46.4
PSPnet [33]	72.0	65.2	81.6	68.1	71.7	96.4
FCN-8s [25]	90.3	75.4	92.1	86.9	86.2	4.3
PathNet[20]	81.6	46.1	64.1	60.8	63.2	196.8
RF+graph [40]	83.5	68.1	77.4	-	-	0
StripNet,4×,200	**91.5**	**79.7**	**93.5**	**87.5**	**88.0**	**0**

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, although there are certain differences in the internal gray scale, their topology is guaranteed. 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 utilization of rough prediction architecture gives more position-sensitive guidance to the network, and the structured output confirms the consistency of the topology, thus leading to the excellent performance in Spectralis OCT images.

However, the traditional FCN-based deep 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 the face of a new dataset, the change in image scale introduced more noise on atrous convolution, thus led to the worst performance. The RF+Graph method also shows topology-correct segmentation results, but still performs unsatisfactory.

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	Topo Err
ResNet-50 [15]	86.1	64.2	53.5	59.7	36.9	78.1	62.3	2092	59.7	66.2	2.2
Res50-StripNet	86.7	65.3	55.5	66.6	39.2	79.7	63.9	2468	61.4	67.4	**0**
SCNN [43]	90.6	69.7	58.5	66.9	43.4	84.1	64.4	1990	66.1	71.6	0.5
SCNN++	90.7	69.7	58.9	69.7	44.1	84.9	64.9	**1891**	65.9	71.9	0.3
SCNN-StripNet	**90.8**	**69.9**	**60.0**	**69.7**	**44.5**	**85.3**	**66.1**	2020	**66.9**	**72.2**	**0**

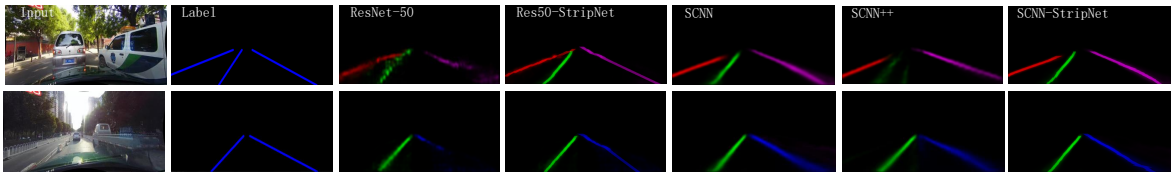

Figure 7: Comparisons between lane detection results of ResNet, Res50-StripNet, SCNN, SCNN++ and SCNN-StripNet

4.3 Lane detection

4.3.1 Data and Implementation Details. We evaluate StripNet on CULane dataset [43], 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-the-art methods. To verify the effect of StripNet on lane detection, we compare it with two state-of-the-art deep methods: ResNet-50 and SCNN. Following the settings of [43], 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 [43] and performs best on CULane dataset up to now. The proposed StripNet are implemented on both ResNet-50 and SCNN, named 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. Moreover, to verify whether the improvement of StripNet is brought by simply adding more model parameters, we compare StripNet with SCNN++, which replaces the original upsample layer with stride 8 to 2 deconvolution layers with stride 2 to make the whole network deeper. All experiments are implemented on the Torch7. The test results on different challenging scenarios are shown in Table. 7.

From the table, we can see that increasing parameters brings little improvement, and SCNN-StripNet outperforms both SCNN and SCNN++. Our method also improves both in ResNet-50 and SCNN, which indicates StripNet's generalization ability across different backbone models. What's more, our method outperforms other methods especially of shadow or dazzle light cases, where FCNs are faced with topological errors due to dark or reflective circumstances. We also evaluate the average number of topological errors per image for each method as shown in the last column of Table. 7. StripNet fixes these topologically errors completely. The comparison examples in shadow and dazzle light scenarios are shown in Fig. 7, holes and islands appear in outputs of conventional FCNs, while StripNet avoids these 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.

6 FUTURE DISCUSSION

StripNet is specifically designed for strip structures. However, the key idea of modeling segmentation patterns as parametric shapes can be extended to other tasks. For examples in cell instance segmentation task, one can approximate each cell by an ellipse or a mixture of ellipses. The obtained results ensure to be topologically correct, i.e., without holes or isolated islands. We hope our paper may inspire researches in related domains.

ACKNOWLEDGMENTS

This work was supported in part by Shenzhen Basic Research Program (JCYJ20150925163005055, JCYJ20170818164704758), Guangzhou Research Program (201803010066), and International Partnership Program of Chinese Academy of Sciences (172644KYSB20160033, 172644KYSB20150019).

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] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[19] 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.

[20] Byungsoo Kim, Oliver Wang, A. Cengiz Öztireli, and Markus Gross. 2018. Semantic Segmentation for Line Drawing Vectorization Using Neural Networks. *Computer Graphics Forum (Proc. Eurographics)* 37, 2 (2018), 329–338.

[21] 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.

[22] 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.

[23] 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)*.

[24] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. 2015. Semantic image segmentation via deep parsing network. In *Proc. ICCV*.

[25] 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.

[26] 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).

[27] 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.

[28] 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

[29] Pedro H. O. Pinheiro and Ronan Collobert. 2014. Recurrent Convolutional Neural Networks for Scene Labeling. In *Proc. ICML*.

[30] Xiaojuan Qi, Jianping Shi, Shu Liu, Renjie Liao, and Jiaya Jia. 2015. Semantic Segmentation With Object Clique Potential. In *Proc. ICCV*.

[31] 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.

[32] 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*.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*. Springer, 234–241.

[34] 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.

[35] Alexander G Schwing and Raquel Urtasun. 2015. Fully connected deep structured networks. *arXiv preprint arXiv:1503.02351* (2015).

[36] Abhishek Sharma, Oncel Tuzel, and David W Jacobs. 2015. Deep Hierarchical Parsing for Semantic Segmentation. *Proc. CVPR* (2015).

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

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] Shih En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. 2016. Convolutional Pose Machines. In *Computer Vision and Pattern Recognition*. 4724–4732.

[43] 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)*.

[44] 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.