Regression of Instance Boundary by Aggregated CNN and GCN

Yanda Meng[1], Wei Meng[1], Dongxu Gao[1], Yitian Zhao[2], Xiaoyun Yang[3], Xiaowei Huang[1], and Yalin Zheng[1]

[1] Department of Eye and Vision Science, Institute of Life Course and Medical Sciences, University of Liverpool, L7 8TX, UK
{Yanda.Meng,M.wei14,Dongxu.Gao,Xiaowei.Huang,yzheng}@liverpool.ac.uk
[2] Cixi Institute of Biomedical Engineering, Ningbo Institute of Industrial Technology, Chinese Academy of Sciences
yitian.zhao@nimte.ac.cn
[3] China Science IntelliCloud Technology Co., Ltd
xiaoyun.yang@intellicloud.ai

Abstract. This paper proposes a simple, intuitive deep learning approach for (biomedical) image segmentation tasks. Different from the existing dense pixel classification methods, we develop a novel multi-level aggregation network to directly regress the coordinates of the boundary of instances in an end-to-end manner. The network seamlessly combines standard convolution neural network (CNN) with Attention Refinement Module (ARM) and Graph Convolution Network (GCN). By iteratively and hierarchically fusing the features across different layers of the CNN, our approach gains sufficient semantic information from the input image and pays special attention to the local boundaries with the help of ARM and GCN. In particular, thanks to the proposed aggregation GCN, our network benefits from direct feature learning of the instances' boundary locations and the spatial information propagation across the image. Experiments on several challenging datasets demonstrate that our method achieves comparable results with state-of-the-art approaches but requires less inference time on the segmentation of fetal head in ultrasound images and of optic disc and optic cup in color fundus images.

Keywords: Regression, Segmentation, GCN, Attention, Aggregation

1 Introduction

The accurate assessment of anatomic structures in biomedical images plays an important role in the management of many medical conditions or diseases. For instance, fetal head (FH) circumference in ultrasound images is a critical indicator for prenatal diagnosis and can be used to estimate the gestational age and to monitor the growth of the fetus [25]. Similarly, the size of the optic disc (OD) and optic cup (OC) in color fundus images is of great importance for the diagnosis of glaucoma, an irreversible eye disease [34]. Manual annotation of this kind of structures by delineating their boundaries in clinics is unrealistic

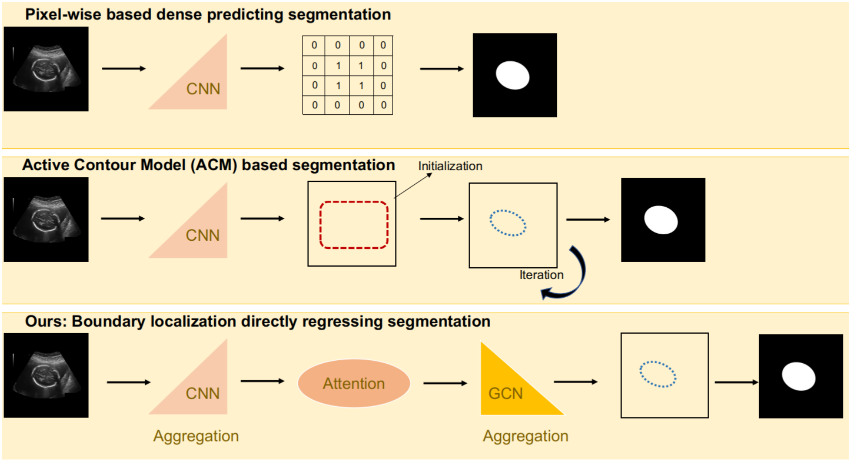

Fig. 1: Three different segmentation paradigms by deep learning. Top row: pixel-wise based methods [14,22,6] that classify each pixel into objects or background. Middle row: active contour based methods [32,9] that need iterative optimization in action to find the final contours. Bottom row: our proposed method that directly regresses the locations of object boundaries by information aggregation through CNN and GCN, enhanced with an attention module.

as it is time-consuming, costly, and subject to human experience and errors. Automatic segmentation of biomedical images is much desired to help improve the efficiency of workflow in clinical scenarios. Inspired by the method in which clinicians annotate images, we propose an aggregated network to solve the segmentation tasks through directly regressing the locations of objects' boundaries, and demonstrate the effectiveness of the network in the segmentation of FH in ultrasound and OD & OC in color fundus images, respectively.

The (biomedical) image semantic segmentation task is an important problem in the field of computer vision. The commonly-used deep learning-based semantic segmentation methods [22,6,40] (top row of Fig. 1) classify each pixel of an image into a category or class. These methods benefit from the CNN's excellent ability to extract high-level semantic features. Being a part of the understanding of scenes or global contexts, these methods need to learn the object location, object boundary, and object category from the high-level semantic information and local location information [31]. However, they suffer from the loss of local location information at the pixel-level [8], because a large receptive field corresponds to a small feature map, and this dilemma has increased the difficulties of dense prediction tasks. In order to solve this problem, approaches in [48,4] either maintain the resolution of the input image with dilated convolution, or capture sufficient receptive fields with pyramid pooling modules. The insights behind these methods indicate that the spatial information and the receptive field are both important to achieving high accuracy. However, it is hard to meet these two requirements simultaneously with CNN [43]. In particular, it is often challenging to maintain enough spatial information of the input image.

To address the aforementioned challenges, we follow a simple and intuitive methodology that human operators take to segment objects and regard segmentation as a regression task. Compared with the preserving abstraction of spatial details [48,4], we use a combination of CNN, ARM, and GCN to directly regress the boundary locations of the instances in the Euclidean space. Our method is different from the recent polygon-based active contour models (ACM) methods [32,9,20] (middle row of Fig. 1), which need to initialize the boundaries and iteratively find the final object boundaries for a new image. On the contrary, we directly supervise the model to learn the precise location of boundaries and produce the boundaries without iteration during inference. Compared with the pixel-wise based methods, our method needs to learn and extract more spatial information to regress the location directly. To address this issue, the local spatial information propagation nature of GCN is exploited. GCN has recently been applied to many low-level tasks, such as scene understanding [29], semantic segmentation [6], and pose estimation [50], because GCN can propagate the information through neighbor nodes (short range) and hence allow the model to learn local spatial correlation structure.

We propose an aggregated GCN decoder with graph vertices sampling from sparse to dense, which contributes to globally propagate the spatial relationship information across the whole image. This will provide greater representational power and more sufficient information propagation than previous segmentation methods based on Conditional Random Fields or Markov Random Fields [2,30]. Thus, we can directly regress explicit boundary location with the Euclidean space coordinate representation. This strategy addresses the concerns of most recent works [41,42], which share the similar idea but convert the Euclidean space representation into polar representation, and regressing the low-level distance between the center point and boundary points. They found that CNN cannot regress the Euclidean space coordinate representation of the boundary well as some more noise may be added, and the CNN may not maintain enough spatial information [41,42]. Our proposed aggregation GCN can handle this issue well, and our experiment results prove that. Besides, those methods' performance may suffer from the low-quality of center point, so, Xie *et al.* [41] utilized center sample methods to classify and selected high-quality center points to improve the segmentation result. In contrast, our methods can directly regress the boundary location without any further center selection process. As for the proposed CNN aggregation mechanism, some low-level features are unnecessarily over-extracted while object boundaries are simultaneously under-sampled. In order to extract more useful and representative features, we apply the ARM working as a filter between CNN encoder and GCN decoder, which cooperates with the GCN to gain more effective semantic and spatial features, especially the boundary location information from CNN.

In summary, this work has the following contributions:

– We take a simple and intuitive approach to (biomedical) image semantic segmentation and regard it as a direct boundary regression problem in an end-to-end fashion.
– We propose aggregating mechanisms on both CNN and GCN modules, to enable them to reuse and fuse the contextual and spatial information. The additional attention mechanism helps the GCN decoder to gain more useful semantic and spatial information from the CNN encoder.
– We propose a new loss function suitable for object boundary localization, which helps prevent taking a large update step when approaching a small range of errors in the late training stage.

It is envisaged that the proposed framework may serve as a fundamental and strong baseline in future studies of biomedical semantic segmentation tasks.

2 Related Work

2.1 Pixel-based Methods

Fully Convolution Neural Networks (FCNs) [31] and U-Net architectures [37] are widely used in semantic segmentation tasks [22,6]. These methods are aimed at extracting more spatial information or extending the receptive field that is of pivotal importance in semantic segmentation tasks. However, it is still difficult to capture longer-range correspondence between pixels in an image [46].

Aggregation module In order to gain global contextual dependencies of an image, methods like [48,51,45,40] proposed to fuse multi-scale or multi-level features through aggregating across semantic and spatial feature domains. Zhao *et al.* [48] proposed a pyramid network that utilizes multiple dilated convolution blocks [44] to aggregating global feature maps on different scales. Other approaches such as Deeplab methods [4,5,6] exploited parallel dilated convolution with different rates to extract features at an arbitrary resolution and preserve the spatial information. However, it is still hard to efficiently learn the discriminative feature representation as many low-level features are unnecessarily over-extracted. Therefore, these aggregation methods may result in an excessive use of information flow.

Attention mechanism Alternatively, some other algorithms exploited the benefits of attention mechanism to integrate local discriminative representation and global contextual features. For example, DANet and CSNet [15,33] used the attentions in spatial and channel dimensions respectively to adaptively integrate local features with their global dependencies. Furthermore, Zhao *et al.* proposed the point-wise spatial attention network [49], which connected each position in the feature map with all the others through self-adaptive attention maps to harvest local and long-range contextual information flexibly and dynamically. In this work, an ARM module is also used to supervise our model to learn discriminate features from input images.

2.2 Polygon-based Methods

Instead of assigning each pixel with a class, some recent methods [32,9,20,41,42] started to predict the position of all vertices of the polygon around the boundary of the target objects. The recent work [41,42] used polar coordinates to represent object contours. Both methods achieved comparable results with pixel-based segmentation methods in instance segmentation tasks. Also, the combination of FCNs and Active Contour Models (ACMs) [27] has been exploited. Some methods formulated new loss functions that were inspired by the ACMs principles [7,21] to tackle the task of ventricle segmentation in cardiac MRI. Other approaches used the ACMs as a post-processor of the output of an FCN, for example, Marcos *et al.* [32] proposed a Deep Structured Active Contours model that combined ACMs and pre-trained FCNs to learn the energy surface of the reference map. These ACM-based methods achieved state-of-the-art performance in many segmentation tasks. However, there are still two main limitations. First, the contour curve must be initialized, while the initialized curve is far away from the ground truth, it may be insufficient to optimize or make an inference. Second, due to the iterative inference mechanism of ACMs, they require a relatively longer running time during training and testing.

2.3 GCNs in Segmentation

GCNs have been applied to image segmentation tasks recently, as they can propagate and exchange the local short-range information through the whole image to learn the semantic relations between objects [38,46]. In 2D image semantic segmentation tasks, Li *et al.* proposed a Dual Graph Convoltional Network (DGCNet) [46], which applied two orthogonal graphs frameworks to compute the global relational reasoning of the whole image and the reasoning process can help the whole network to gain rich global contextual information. Another work [38] proposed by Shin *et al.* shared the similar idea, and utilized GCN to learn the global structure of the shape of the object, which reflected the connectivity of neighbouring vertices. Apart from using GCN to learn global contextual information from 2D input, our approach also exploits spatial and local location information.

3 Method

3.1 Data Representation

The manually annotated object boundaries are extracted from the binary image and uniformly divided into N vertices with the same interval $\varDelta\,\theta$ (e.g. N = 360, $\varDelta$ $\theta = 1°$). The geometric center of the boundary is defined as the center vertex. We represent the object boundary with vertices and edges as $B = (V, E)$, where V has $N+1$ vertices in the Euclidean space, $V \in \mathbb{R}^{N\times 2}$, and $E \in \{0, 1\}^{(N+1)\times(N+1)}$ is a sparse adjacency matrix, representing the edge connections between vertices, where $E_{i,j} = 1$ means vertices V_i and V_j are connected by an edge, and

$E_{i,j} = 0$ otherwise. In our work, every two consecutive vertices on the boundary are connected with each other and are both connected to the center vertices to form a triangle. For the OD & OC segmentation, the OD and OC are divided separately while the centre of the OC is shared as the centre vertex. Thus, there are 360 triangles and 361 vertices for instances in FH images and 720 triangles and 721 vertices for OD & OC images. For more details, please refer to supplementary material.

We directly use the coordinates in the Euclidean space to represent all the vertices and exploit the semantic and spatial correspondence between the inputs' instance and boundaries. Besides, our boundary representation method is not sensitive to the center point as the boundary does not have too many correlations with the center point.

3.2 Graph Fourier Transform & Convolution

Following [10], the normalized Laplacian matrix is $L = I - D^{-\frac{1}{2}} E D^{-\frac{1}{2}}$, where I is the identity matrix, and D is a diagonal matrix that represents the degree of each vertex in V, such that $D_{i,i} = \sum_{j=1}^{N} E_{i,j}$. The Laplacian of the graph is a symmetric and positive semi-definite matrix, so L can be diagonalized by the Fourier basis $U \in \mathbb{R}^{N \times N}$, such that $L = U \Lambda U^T$. The columns of U are the orthogonal eigenvectors $U = [u_1, ..., u_n]$, and $\Lambda = diag([\lambda_1, ..., \lambda_n]) \in \mathbb{R}^{N \times N}$ is a diagonal matrix with non-negative eigenvalues. The graph Fourier transform of the vertices representation $x \in \mathbb{R}^{N \times 3}$ is defined as $\hat{x} = U^T x$, and the inverse Fourier transform as $x = U\hat{x}$. The spectral graph convolution of i and j is defined as $i * j = U((U^T i) \odot (U^T j))$ in the Fourier space. Since U is not a sparse matrix, this operation is computationally expensive. To reduce the computation, Defferrard *et al.* [12] proposped that the convolution operation on a graph can be defined in Fourier space by formulating spectral filtering with a kernel g_θ using a recursive Chebyshev polynomial [12]. The filter g_θ is parametrized as a Chebyshev polynomial expansion of order K, such that

$$g_\theta(L) = \sum_{k=1}^{K} \theta_k T_k(\hat{L}) \tag{1}$$

where $\theta \in \mathbb{R}^K$ is a vector of Chebyshev coefficients, and $\hat{L} = 2L/\lambda_{max} - I_N$ represents the rescaled Laplacian. $T_k \in \mathbb{R}^{N \times N}$ is the Chebyshev polynomial of order K, that can be recursively computed as $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0 = 1$ and $T_1 = x$. Therefore, the spectral convolution can be defined as

$$y_j = \sum_{i=1}^{F_{in}} g_{\theta_{i,j}}(L) x_i \tag{2}$$

where x_i is the i-th feature of input $x \in \mathbb{R}^{N \times F_{in}}$, which has F_{in} features, with $F_{in} = 2$ in this work and $y \in \mathbb{R}^{N \times F_{out}}$ is the output. The entire filter operation is computationally faster and the complexity drops from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ [3].

3.3 Graph Vertices Sampling

To achieve multi-scale aggregated graph convolutions on different vertex resolutions, we follow [35] to form a new topology and neighbour relationships of vertices. More specifically, we employ the permutation matrix $Q_d \in \{0,1\}^{m \times n}$ to down-sample m vertices, m = 360 or 720 in our work. Q_d is obtained by iteratively contracting vertex pairs, which uses a quadratic matrix to maintain surface error approximations [17]. The down-sampling can be seen as a pre-processing, and the discarded vertices are recorded with barycentric coordinates so that the up-sampling can map the discarded vertices back with the same barycentric location information. We conduct up-sampling with another transformation matrix $Q_u \in \mathbb{R}^{m \times n}$. The up-sampled vertices V_u can be obtained by a sparse matrix multiplication, i.e., $V_u = Q_u V_d$, where V_d are down-sampled vertices. The up-sampling is applied during learning, and it operates convolution transformations on retained vertices.

3.4 Proposed Aggregation Network

Our novel aggregation graph regression network is motivated by fusing features hierarchically and iteratively [45,51,40], which consists of an image context encoder, an attention refinement module and a vertex location decoder. Both the encoder and decoder contain aggregation mechanisms through up-samplings and down-samplings, which provide improvements in extracting the full spectrum of semantic and spatial information across stages and resolutions. Besides, the attention module plays an essential role to guide the feature learning and refine the output from the CNN encoder, then passes to the GCN decoder through multi-paths. In Section 5.3, our ablation study demonstrates that the proposed aggregation module helps to extract more useful information, and the attention module helps to refine the extracted features from the encoder to guide feature learning better.

Semantic Encoder Fig. 2 (a) shows the detailed structure of our image context encoder, which maintains high-resolution representations by connecting low-to-high resolution convolutions in parallel, where multi-scale fusions are repeated across different levels (rows). Our encoder is designed to lessen the location information loss and extract a wider spectrum of semantic features through different receptive fields. The encoder takes input images of shape 314×314×3 (Fundus OD & OC images) or 140×140×1 (Ultrasound FH images), with operations of up-sampling and down-sampling. The aggregation block can extract and reuse more features across various resolutions and scales, which helps to reduce spatial information loss during the encoding process. Our encoder has six output features, for each, the shape is 5×5×128, and then those intermediate features will be as input to the attention refinement module.

Attention Module: We propose an Attention Refinement Module (ARM) to refine the features of each level (row) from the outputs of the encoder. As Fig. 2 (a) & (b) shows, ARM contains five attention blocks, and each block employs global channels average pooling to capture global context through the different

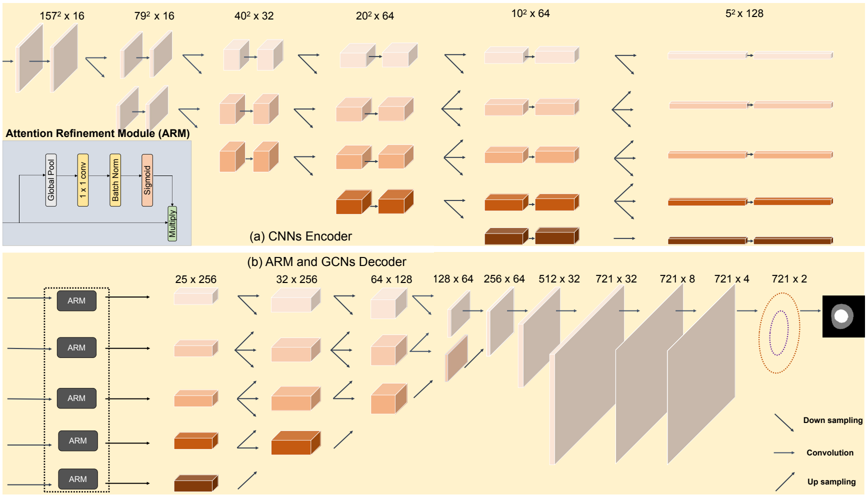

Fig. 2: Overview of our proposed network structure. The size of feature maps of the CNN encoder and vertex maps of the GCN decoder for each stage (columns) are shown. In the CNN encoder, the horizontal arrow represents CNN convolutional operations that are achieved by a standard CNN Residual Block [24] with kernel size 3 x 3, stride 1, followed by a Batch Normalization (BN) layer [26] and Leaky ReLU as the activation function. The down-sampling is conducted by setting stride size as 2, the lower level feature is bi-linearly up-sampled by a factor 2. In the GCN decoder, down-sampling and up-sampling are conducted by graph vertices sampling, which is described in Section 3.3, and the horizontal arrow represents residual graph convolution (ResGCN) blocks [28] with polynomial order 3. In this figure, the example is for OD & OC segmentation, and for FH segmentation, the convolution operation will be the same. Still, the feature map and vertex map size will be different because of different input size and number of contours of instances.

channels, and computes an attention vector to guide the feature learning through a convolution layer followed by a BN layer and sigmoid as the activation function. For the filter, the kernel size is 1×1, and the stride is 1. This design can refine the output features of each stage in the Semantic Encoder, which easily integrates the global context information.

Spatial Decoder The decoder takes refined multi-paths outputs from the attention module, then decodes with ResGCN blocks [28] through different stages and levels, which has been shown that as layers go deeper, ResGCN blocks can prevent vanishing gradient problems. As Fig. 2 (b) shows, our decoder fuses and reuses the features extracted by ResGCN blocks through different stages. Benefits from the graph sampling, our decoder can regress the location of the object boundary from sparse to dense, which allows the ResGCN blocks to hierarchically extract spatial location information from refined outputs of the attention module. For each ResGCN Block, it consists of 4 graph convolution layers, and each graph convolution layer is followed by a Batch Normalization layer [26] and Leaky ReLU as the activation function. After going through ResGCN blocks and graph vertices up-samplings, the number of vertices is up-sampled from 25 to 721, and each vertex is represented by a vector of length 32. At last, three graph convolution layers are added to generate 2D object contour vertices, which reduces the vertex feature map channels to 2, as each contour vertex has two dimensions: x and y. With the output from the decoder, we connect every two consecutive vertices on the boundary to form a polygon contour as the final segmentation result.

3.5 Loss Function

We regard segmentation as a vertices location regression problem. L2 and L1 loss have been used in regression tasks by CNN based networks [19,23]. However, it is difficult for the L1 loss to continuously converge and find the global minimization in the late training stage without careful tuning of the learning rate. It is commonly known that the L2 loss is sensitive to outliers which may lead to unstable training in the early training stage.

Inspired by Wing-loss [13] and Smooth-L1 loss [18], we propose a new loss function (Fig. 3) that can prevent the model from taking large update steps when reaching small range errors in the late training stage and can recover quickly when dealing with large errors during the early training stage. Our loss function is defined as:

$$L(x) = \begin{cases} W[e^{(|x|/\epsilon)} - 1] & if|x| < W \\ |x| - C & otherwise \end{cases} \tag{3}$$

Where W should be non-negative and limit the range of the non-linear part, ϵ decides the curvature between $(-W, W)$ and $C = W - W[e^{(|w|/\epsilon)} - 1]$ connects the linear and non-linear parts. After several evaluation experiments, the parameter W is set to 8 and ϵ to 5 for FH segmentation and W = 6, $\epsilon = 5$ for OD & OC segmentation. For the OD & OC segmentation tasks, we integrate a weight mask and assign more weights to the vertices that belong to the OC, to improve the OC segmentation performance, as OC is usually difficult to segment due to the image quality or poor color contrast.

4 Experiments

4.1 Datasets

We evaluate our approach with two major types of biomedical images on two segmentation tasks respectively: fundus images of retinal for OD & OC segmentation, and ultrasound images of the fetus for FH segmentation.

Fudus OD & OC images: 2068 images from five datasets are merged together. 190 fundus images are randomly selected as the retina test dataset, the rest 1878

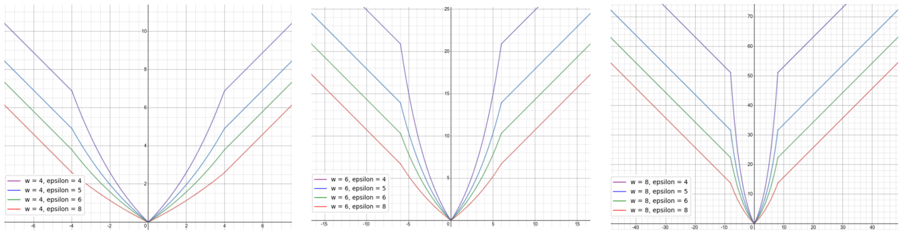

Fig. 3: The proposed new loss function plotted with different parameter settings, where w controls the non-linear part and epsilon (ϵ) limits the curvature.

fundus images are used for the training .Considering the negative influence of non-target areas in fundus retina images, we first localize the disc centers by detector [36] and crop to 314 $\times$ 314 pixels and then transmit into our network. **Refuge** [34]. The dataset consists of 400 training images and 400 validation images. The pixel-wise OD & OC gray-scale annotations are provided. **Drishti-GS** [39], which contains 50 training images and 51 validation images. All images are taken centered on OD & OC with a field-of-view of 30 degrees. The annotations are provided in the form of average boundaries. **ORIGA** [47], contains 650 fundus images. The OD & OC boundaries were manually marked by experienced graders from the Singapore Eye Research Institute. **RIGA** [1], contains 750 fundus images from MESSIDOR [11] database, which are labeled manually by six ophthalmologists. **RIM-ONE** [16], contains 169 fundus images, annotated by five different expert.
Ultrasound FH images: The HC18-Challenge dataset [4] [25], contains 999 two-dimensional (2D) ultrasound images with size of 800 $\times$ 540 pixels, and collected from the database of Radboud University Medical Center. We zero-padding each image to shape of 840 $\times$ 840, and then resize into 140 $\times$ 140 as the input image, then we randomly select 94 images as the test dataset, and the model is trained on the rest 905 images.

4.2 Implementation Details

To augment our dataset, we perturb the input image of training dataset by randomly rotating images for both segmentation tasks. Specifically, the rotation ranges from -15 to 15 degree. We randomly select 10 % of training dataset as the validation dataset. We use stochastic gradient descent with a momentum of 0.9 to optimize our loss function. The number of graph vertices is sampled to 361, 256, 128, 64, 32, 25 crosses five stages with Graph Vertices Sampling introduced in Section 3.3. We trained our model 300 epochs for all datasets, with a learning rate of 1e-2 and decay rate of 0.997 every epoch. The batch size is set as 48. All training processes are performed on a server with 8 TESLA V100 and 4 TESLA P100, and all test experiments are conducted on a local machine Geforce RTX 2080Ti.

[4] https://hc18.grand-challenge.org/

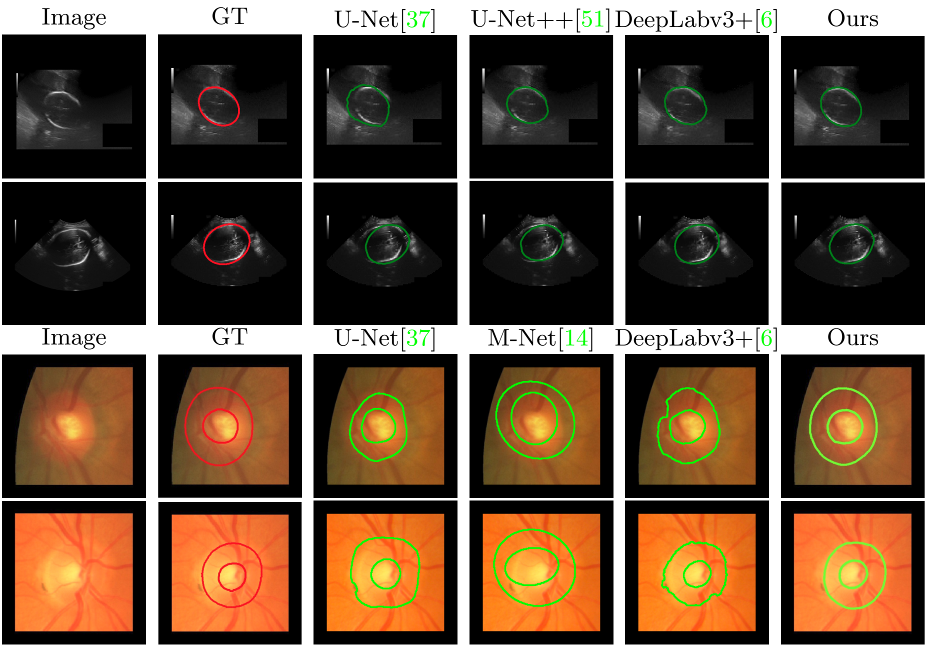

Fig. 4: Qualitative results of segmentation on retina test dataset and HC18-Challenge [25]. Top two rows are the ultrasound FH segmentation results, and the bottom two rows are the fundus OD & OC segmentation results.

5 Results

In this section, we show our qualitative and quantitative results on the OD & OC segmentation and FH segmentation task. We compare our model with other state-of-the-art methods, including U-Net [37], PolarMask [41], M-Net [14], U-Net++ [51], DANet [15], DARNet [9], DeepLabv3+ [6] through running their open public source code. Dice score and Area Under the Curve (AUC) are used as the segmentation accuracy metrics. The results of an ablation study are shown in order to demonstrate the effectiveness of the proposed aggregation mechanism, attention mechanism and loss function, respectively.

5.1 Optic Disc & Cup Segmentation

We perform evaluation experiments on the retina test dataset, which is merged with five different fundus OD &OC images datasets. In terms of different dataset sources, they may contain different annotation standards for ground truths by different doctors. However, our model still achieve good performance, which shows the robustness and generalizability of our model. Fig. 4 shows some qualitative results. We achieve 96.88 % and 92.46 % Dice score on OD & OC segmentation separately without any bells and whistles (multi-scale training, ellipse fitting, longer training epochs, etc.), comparable with other pixel-wise based state-

Methods \ Tasks	OC		OD		FH	
	Dice Score	AUC	Dice Score	AUC	Dice Score	AUC
U-Net [37]	0.9016	0.9186	0.9522	0.9648	0.9625	0.9688
M-Net[14]	**0.9335**	**0.9417**	0.9230	0.9332	-	-
U-Net++ [51]	0.9198	0.9285	0.9626	0.9777	0.9701	0.9789
DANet [15]	0.9232	0.9327	0.9654	0.9726	0.9719	0.9786
DARNet [9]	0.9235	0.9339	0.9617	0.9684	0.9719	**0.9790**
PolarMask [41]	0.9238	0.9366	**0.9670**	**0.9782**	**0.9723**	0.9780
DeepLabv3+ [6]	**0.9308**	**0.9406**	**0.9669**	**0.9779**	**0.9779**	**0.9819**
Our method	**0.9246**	**0.9376**	**0.9688**	**0.9784**	**0.9738**	**0.9796**

Table 1: Segmentation results on retina test dataset for OD & OC and on HC18-Challenge [25] for FH. The performance is reported as Dice score (%) and AUC (%). The top three results in each category are highlighted in bold.

of-the-art methods. Tab. 1 provides the results of ours and the other methods. As for the inference speed, our model achieves faster result with 66.6 milliseconds (ms) per image than PolarMask [41] (72.1 ms) and DeepLabv3 [6] (323.9 ms). In the supplementary material, we also show some 'failed' cases compared with ground truth. According to the comments from an expert at the anonymous institute, our model produces more accurate results than the ground truth. This highlights the potential issue of imperfect ground truth.

5.2 Fetal Head Segmentation

Tab. 1 and Fig. 4 shows the quantitative and qualitative results respectively, our model achieves 0.9738 % Dice score and 0.9796 % AUC, which outperforms U-Net++ [51] and DANet [15] by 0.2 % in terms of Dice score. Our model (60.2ms) is faster than PolarMask [41] (65.5 ms) and Deeplabv3+ [6] (290.3ms) for per image inference.

5.3 Ablation Study

We investigate the effect of each component in our proposed model step by step. All the ablation experiments are performed with the same setting as section 4.2 described. The performance of each experiment in the form of Dice Score (%) and AUC (%) are reported in Fig. 5, Tab. 4, Tab. 2 and Tab. 3. The top performance in each category is highlighted in bold. For more qualitative results, please refer to supplementary material.

Ablation on Parameters of Loss Function We perform Experiments to evaluate the effect of parameter settings of our proposed loss function. When w = 6, $\epsilon = 5$, our model achieve the best performance on OD & OC segmentation test dataset, and w = 8, $\epsilon = 5$, for FH segmentation test dataset. For more details, please refer to Fig. 5.

Ablation on Loss Function We conduct experiments to evaluate the effectiveness of our proposed loss function. We compare with L1, L2, Smooth-L1 [18] loss functions, which are commonly used in the regression problem. Tab. 2 shows

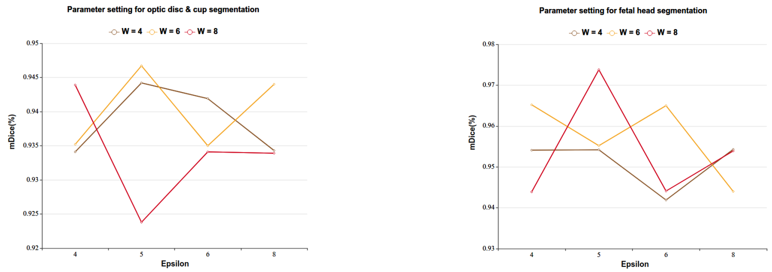

Fig. 5: A comparison of different parameter settings (w and ϵ) for the proposed loss function, measured in terms of the mean Dice score (%) on retina test dataset for OD & OC. With w = 6, $\epsilon = 5$, our model achieves the best performance (92.46 % & 96.88 %). On HC18-Challenge test dataset [25] for FH segmentation, with w = 8, $\epsilon = 5$, our model gains the best results (97.38%). It shows that our network is not sensitive to these parameters as no significantly different results are found.

Loss Function \ Tasks	OC		OD		FH	
	Dice Score	AUC	Dice Score	AUC	Dice Score	AUC
L1	0.9108	0.9256	0.9543	0.9636	0.9503	0.9684
L2	0.9103	0.9208	0.9553	0.9668	0.9442	0.9571
Smooth-L1 [18]	0.9086	0.9112	0.9521	0.9652	0.9395	0.9452
Our proposed Loss						
weight mask = 0	0.9183	0.9218	0.9616	0.9738		
weight mask = 3	0.9223	0.9338	0.9646	0.9768		
weight mask = 5	**0.9246**	**0.9376**	**0.9688**	**0.9784**	**0.9738**	**0.9796**
weight mask = 7	0.9173	0.9238	0.9623	0.9718		
weight mask = 9	0.9109	0.9215	0.9603	0.9708		

Table 2: Performance comparisons (%) of the different loss function and weight mask parameter settings on the OD & OC segmentation and the FH segmentation respectively. For weight mask = 5, our model achieves best performance on the OD & OC segmentation.

the quantitative results on OD & OC and FH segmentation tasks respectively. As illustrated, our loss function attains a superior performance over the other three loss functions. In particular, it achieves a mean Dice score that is 1.5 % relatively better than that of L1 loss function on OD & OC and 2.5 % relatively better than L1 loss function on FH segmentation. Tab. 2 shows comparing with no-weight mask loss function, our proposed weight mask helps to improve OD & OC segmentation results by 0.7 % when weight mask = 5 is used.

Ablation on Angle Interval Experiments are conducted to evaluate the effect of different angle intervals $\Delta\theta$ for vertices sampling. The larger angle interval indicates that, the smaller number of vertices are sampled on the boundary. When $\Delta\theta = 1°$, our model achieves best performance on both the FH segmentation and the OD & OC segmentation. The results are shown in Tab. 3.

Ablation on Structure Components In this section, we evaluate the effectiveness and compactness of our aggregation module, attention module and GCN decoder through several experiments. First, we compare with no-aggregation

Angle Interval \ Tasks	OC		OD		FH	
	Dice Score	AUC	Dice Score	AUC	Dice Score	AUC
40°	0.9023	0.9093	0.9154	0.9233	0.9415	0.9502
18°	0.9103	0.9193	0.9488	0.9553	0.9515	0.9599
10°	0.9195	0.9282	0.9583	0.9647	0.9601	0.9694
5°	0.9238	0.9306	0.9628	0.9714	0.9708	0.9776
2°	0.9243	**0.9376**	0.9685	0.9780	0.9737	**0.9797**
1°	**0.9246**	**0.9376**	**0.9688**	**0.9784**	**0.9738**	0.9796

Table 3: Ablation study on different angle interval samplings. With angle interval = 1° or 2°, our model achieves comparable segmentation results on the OD & OC and FH segmentation tasks, and at the end, angle interval = 1° is chosen for our model. Dice score (%) and AUC (%) are reported for the segmentation on OD & OC and FH test dataset.

Methods \ Tasks	OC		OD		FH	
	Dice Score	AUC	Dice Score	AUC	Dice Score	AUC
No Aggregation (Encoder + Decoder)	0.9023	0.9063	0.9588	0.9664	0.9565	0.9689
Aggregation	0.9205	0.9302	0.9623	0.9658	0.9699	0.9774
Aggregation + ARM (with CNN decoder)	0.9098	0.9176	0.9528	0.9634	0.9638	0.9756
Aggregation + ARM (Our method)	**0.9246**	**0.9376**	**0.9688**	**0.9784**	**0.9738**	**0.9796**

Table 4: Ablation study on different structure components with our proposed loss function (w = 8, ϵ = 5 for FH segmentation and w = 6, ϵ = 5 for OD & OC). Dice score (%) and AUC (%) are reported respectively.

structure network, in which we remove all the aggregation parts and attention modules, leaving the one path CNN encoder and one path GCN decoder to form a standard encoder-decoder network structure. Then we add an aggregation module to form an aggregation network with an aggregated CNN encoder and aggregated GCN decoder. To further improve the performance, we design an ARM, and the effect of the ARM is presented in Tab 4. Furthermore, we evaluate the effectiveness of proposed GCN decoder and change the GCN into CNN, which are the same as we used in our encoder. As illustrated in the table, for FH segmentation, the proposed aggregation module helps to improve 1.4 % on Dice score over the no-aggregation method, the ARM module further improves 0.4 %, and GCN decoder further improves 0.6 %. For OD & OC segmentation, the aggregation module improves 1.2 % on average by Dice score, the ARM improves 0.6 %, and the GCN decoder improves 1.7 %.

6 Conclusion

We propose a simple and intuitive regression methodology to tackle segmentation tasks by directly regressing the boundary of the instances instead of utilising pixel-wise dense predictions. We have demonstrated its potentials on the segmentation problems of the fetal head and optic disc & cup. It is anticipated that our approach can be widely applicable in the real world.

References

1. Almazroa, A., Alodhayb, S., Osman, E., Ramadan, E., Hummadi, M., Dlaim, M., Alkatee, M., Raahemifar, K., Lakshminarayanan, V.: Retinal fundus images for glaucoma analysis: the riga dataset. In: Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications. vol. 10579, p. 105790B. International Society for Optics and Photonics (2018)
2. Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F., Torr, P.H.: Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. IEEE Signal Processing Magazine **35**(1), 37–52 (2018)
3. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
5. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
6. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: ECCV (2018)
7. Chen, X., Williams, B.M., Vallabhaneni, S.R., Czanner, G., Williams, R., Zheng, Y.: Learning active contour models for medical image segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11632–11640 (2019)
8. Chen, Y., Zhao, D., Lv, L., Zhang, Q.: Multi-task learning for dangerous object detection in autonomous driving. Information Sciences **432**, 559–571 (2018)
9. Cheng, D., Liao, R., Fidler, S., Urtasun, R.: Darnet: Deep active ray network for building segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7431–7439 (2019)
10. Chung, F.R., Graham, F.C.: Spectral graph theory. No. 92, American Mathematical Soc. (1997)
11. Decencière, E., Zhang, X., Cazuguel, G., Lay, B., Cochener, B., Trone, C., Gain, P., Ordonez, R., Massin, P., Erginay, A., et al.: Feedback on a publicly distributed image database: the messidor database. Image Analysis & Stereology **33**(3), 231–234 (2014)
12. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
13. Feng, Z.H., Kittler, J., Awais, M., Huber, P., Wu, X.J.: Wing loss for robust facial landmark localisation with convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2235–2245 (2018)
14. Fu, H., Cheng, J., Xu, Y., Wong, D.W.K., Liu, J., Cao, X.: Joint optic disc and cup segmentation based on multi-label deep network and polar transformation. IEEE transactions on medical imaging **37**(7), 1597–1605 (2018)
15. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3146–3154 (2019)

16. Fumero, F., Alayón, S., Sanchez, J.L., Sigut, J., Gonzalez-Hernandez, M.: Rim-one: An open retinal image database for optic nerve evaluation. In: 2011 24th international symposium on computer-based medical systems (CBMS). pp. 1–6. IEEE (2011)
17. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. pp. 209–216. ACM Press/Addison-Wesley Publishing Co. (1997)
18. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
19. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
20. Gur, S., Shaharabany, T., Wolf, L.: End to end trainable active contours via differentiable rendering. arXiv preprint arXiv:1912.00367 (2019)
21. Gur, S., Wolf, L., Golgher, L., Blinder, P.: Unsupervised microvascular image segmentation using an active contours mimicking neural network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 10722–10731 (2019)
22. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
23. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence **37**(9), 1904–1916 (2015)
24. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
25. van den Heuvel, T.L., de Bruijn, D., de Korte, C.L., van Ginneken, B.: Automated measurement of fetal head circumference using 2d ultrasound images. PloS one **13**(8) (2018)
26. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
27. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International journal of computer vision **1**(4), 321–331 (1988)
28. Li, G., Müller, M., Thabet, A., Ghanem, B.: Can gcns go as deep as cnns? arXiv preprint arXiv:1904.03751 (2019)
29. Li, Y., Gupta, A.: Beyond grids: Learning graph representations for visual recognition. In: Advances in Neural Information Processing Systems. pp. 9225–9235 (2018)
30. Liu, Z., Li, X., Luo, P., Loy, C.C., Tang, X.: Semantic image segmentation via deep parsing network. In: Proceedings of the IEEE international conference on computer vision. pp. 1377–1385 (2015)
31. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
32. Marcos, D., Tuia, D., Kellenberger, B., Zhang, L., Bai, M., Liao, R., Urtasun, R.: Learning deep structured active contours end-to-end. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8877–8885 (2018)
33. Mou, L., Zhao, Y., Chen, L., Cheng, J., Gu, Z., Hao, H., Qi, H., Zheng, Y., Frangi, A., Liu, J.: Cs-net: Channel and spatial attention network for curvilinear structure segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 721–730. Springer (2019)

34. Orlando, J.I., Fu, H., Breda, J.B., van Keer, K., Bathula, D.R., Diaz-Pinto, A., Fang, R., Heng, P.A., Kim, J., Lee, J., et al.: Refuge challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs. Medical image analysis **59**, 101570 (2020)
35. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3d faces using convolutional mesh autoencoders. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 704–720 (2018)
36. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
38. Shin, S.Y., Lee, S., Yun, I.D., Lee, K.M.: Deep vessel segmentation by learning graphical connectivity. Medical image analysis **58**, 101556 (2019)
39. Sivaswamy, J., Krishnadas, S., Joshi, G.D., Jain, M., Tabish, A.U.S.: Drishti-gs: Retinal image dataset for optic nerve head (onh) segmentation. In: 2014 IEEE 11th international symposium on biomedical imaging (ISBI). pp. 53–56. IEEE (2014)
40. Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., Wang, J.: High-resolution representations for labeling pixels and regions. arXiv preprint arXiv:1904.04514 (2019)
41. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P.: Polarmask: Single shot instance segmentation with polar representation. arXiv preprint arXiv:1909.13226 (2019)
42. Xu, W., Wang, H., Qi, F., Lu, C.: Explicit shape encoding for real-time instance segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5168–5177 (2019)
43. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 325–341 (2018)
44. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
45. Yu, F., Wang, D., Shelhamer, E., Darrell, T.: Deep layer aggregation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2403–2412 (2018)
46. Zhang, L., Li, X., Arnab, A., Yang, K., Tong, Y., Torr, P.H.: Dual graph convolutional network for semantic segmentation. arXiv preprint arXiv:1909.06121 (2019)
47. Zhang, Z., Yin, F.S., Liu, J., Wong, W.K., Tan, N.M., Lee, B.H., Cheng, J., Wong, T.Y.: Origa-light: An online retinal fundus image database for glaucoma analysis and research. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. pp. 3065–3068. IEEE (2010)
48. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
49. Zhao, H., Zhang, Y., Liu, S., Shi, J., Change Loy, C., Lin, D., Jia, J.: Psanet: Point-wise spatial attention network for scene parsing. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 267–283 (2018)
50. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3d human pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3425–3435 (2019)

51. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: Unet++: A nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp. 3–11. Springer (2018)