END-TO-END DETECTION-SEGMENTATION NETWORK WITH ROI CONVOLUTION

Zichen Zhang⋆, Min Tang⋆, Dana Cobzas⋆, Dornoosh Zonoobi†, Martin Jagersand⋆, Jacob L. Jaremko†

⋆ Computing Science Department, University of Alberta, Edmonton, Canada
† Radiology Department, University of Alberta, Edmonton, Canada

ABSTRACT

We propose an end-to-end neural network that improves the segmentation accuracy of fully convolutional networks by incorporating a localization unit. This network performs object localization first, which is then used as a cue to guide the training of the segmentation network. We test the proposed method on a segmentation task of small objects on a clinical dataset of ultrasound images. We show that by jointly learning for detection and segmentation, the proposed network is able to improve the segmentation accuracy compared to only learning for segmentation.

Index Terms— segmentation, detection, fully convolutional neural networks, ultrasound.

1. INTRODUCTION

Fully Convolutional Network (FCN) [1] is the de-facto network architecture in semantic segmentation and has been the basis for many advanced network models. U-Net [2] was developed specifically for medical imaging segmentation. The network had a U shape architecture where it tried to make upsampling and downsampling streams symmetrical, with skip connections that concatenated the feature maps from downsampling layers to those in the upsampling layers. Simon et al. [3] employed dense skip connections for semantic segmentation. Similar to U-Net, their network was also in a U shape. The dense skip connections were added to the downsampling stream only. The common architectural design is that the input image will go through downsampling layers (convolution and pooling). Then the encoded features will be passed through upsampling layers (transposed convolution) to get the score map of the same size as the input. All these networks are trained with the same cross-entropy loss function that essentially makes the network learn a classifier for each pixel and average or sum up the loss for all pixels. This makes it hard to train the network when the object to segment is small.

Intuitively, humans perform semantic segmentation by first attending to some rough region where each pixel inside that region is then classified. In other words, object detection can be a first step for accurate semantic segmentation. There have been some attempts in using object detection for

segmentation tasks [4, 5], but they relied on external region proposal methods such as Selective Search [6].

Some recent works have started to jointly learn detection and segmentation [7, 8]. Mask RCNN [8] is the state-of-the-art for instance segmentation where it first predicts the bounding box of each object instance in the image, then predicts pixel-wise labels inside each predicted box. The network was trained end-to-end with the multi-task loss of localization, classification and segmentation. It was shown that this multi-task loss improved the accuracy compared to training for each task individually. However, a pooling layer called "ROI-align" was used to transform the regions of interest (ROI) to fixed size input for the segmentation stream. It reduced the resolution of the features which caused a loss of spatial details especially along the object boundary. Also the segmentation stream caused computational overhead since it processed each region individually. BlitzNet [7] adopted a fully convolutional architecture and performed image-wise computation. But it did not make explicit use of the detection result to improve segmentation accuracy.

In this paper, we propose a fully convolutional architecture that incorporates an object localization unit to improve semantic segmentation accuracy. Our work addresses the issues discussed above by introducing a new type of convolutional layer named ROI convolution, that replaces the ROI pooling layer by convolving directly on top of the region proposals from the localization network. It applies convolution on the features inside all the ROIs in one shot without the need of passing them individually through the downstream convolutional layers. This proposed network performs image-wise computation and can be trained end-to-end. To demonstrate the efficacy of our method, we apply it to the problem of segmentation of acetabulum from ultrasound images. It is a challenging task since the object of interest is small. We show that with the localization unit, our method performs well despite the high class imbalance.

2. PROPOSED METHOD

The overall architecture of the proposed network is shown in Figure 1. In this section, we will go over the details of each component.

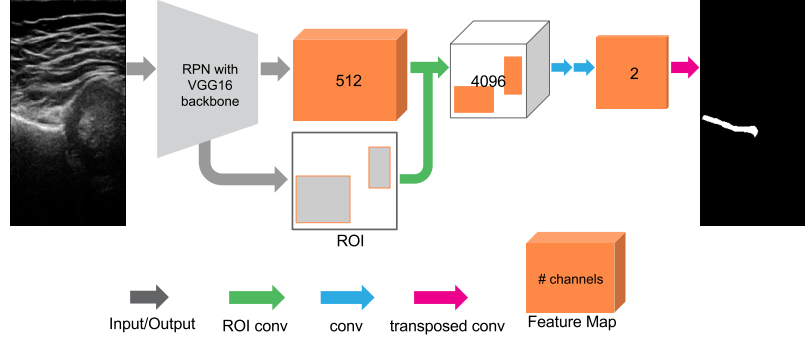**Backbone** We use VGG16 as the feature extractor, i.e., the

Fig. 1: Overall architecture of our network. The legend at the bottom illustrates different components. Best viewed in color. The input to the network can be any 2D grayscale (duplicated to 3 channels) or color image. The RPN network is built on top of VGG16. It takes an input image and outputs feature maps of 512 channels and a number of ROIs. The arrow for 'ROI conv' represents one ROI convolutional layer that takes as input the feature map and the ROIs. Each 'conv' and 'transposed conv' arrow represents one convolutional layer and transposed convolutional layer, respectively.

backbone of the architecture. The network that we propose could be built on top of a deeper backbone like ResNet [9] to further improve accuracy. We adopt the simpler backbone VGG16 just to demonstrate how the new architecture improves segmentation with detection.

Localization unit We adopt the Region Proposal Network (RPN) from Faster R-CNN [10] to perform object localization. Each region is classified as foreground or background. The foreground regions are passed to the downstream segmentation network where each pixel inside the regions is classified.

ROI convolution The output of the RPN is a list of ROIs in the form of bounding boxes. The features of the ROIs can be extracted from the feature map of the previous layer and passed to downstream segmentation network. Region-wise method has been widely used [8, 10, 5] where each ROI is processed individually as a new input image. To obtain a fixed-size input, each ROI is first fed into a pooling layer which produces a fixed-size grid. The features for each cell in the grid are those of the pixel that has the maximum activation in that cell. In contrast, we propose an image-wise method to simplify the computation, where the downstream segmentation network processes all the ROIs in one shot. In order to do that, a new layer ROI convolution is introduced. A similar idea has been proposed in [11]. But they used ROI convolution only in the inference time to speed up computation. We show that ROI convolution is differentiable and is trainable through backward propagation. We use ROI convolution at both training and inference time.

Forward and back propagation of ROI convolution
The forward pass of ROI convolution is shown in Eq. 1.

$$
y_{i,j}^l = (w^l * z^{l-1})_{i,j} = \mathbb{1}[(i,j) \in \cup\text{ROI}] \sum_{b=0}^{k-1} \sum_{a=0}^{k-1} w_{a,b}^l z_{i+a,j+b}^{l-1} \tag{1}
$$

where $y_{i,j}^l$ represents the pre-activation output of layer l at pixel (i,j). z^l represents the activation of layer l. w^l is the filter at layer l, of size $k \times k$. $*$ denotes the ROI convolution. $\cup\,\text{ROI}$ denotes the union of the set of indices inside all the ROIs. The indicator function checks if the pixel (i,j) is inside any ROI. The forward pass is just like the regular convolution except that the pixels outside the ROIs are set to zeros.

The backpropagation of ROI convolution is derived as follows. Suppose that we have the cost function E, and $\frac{\partial E}{\partial z^l} = \delta^l$ for the current layer l, we need to compute $\frac{\partial E}{\partial z^{l-1}}$ and $\frac{\partial E}{\partial w^l}$:

$$
\frac{\partial E}{\partial z_{i,j}^{l-1}} = \sum_{a=0}^{k-1} \sum_{b=0}^{k-1} (\delta^l \, \sigma')_{i-a,j-b} w_{a,b}^l \mathbb{1}[(i-a, j-b) \in \cup\text{ROI}]
$$

$$
\frac{\partial E}{\partial w_{a,b}^l} = \sum_{j=0}^{W_o^l} \sum_{i=0}^{H_o^l} \delta_{i,j}^l \sigma_{i,j}' z_{i+a,j+b}^{l-1} \mathbb{1}[(i,j) \in \cup\text{ROI}] \tag{2}
$$

where W_o^l, H_o^l denotes the width and height of the output of layer l. σ' denotes the derivative of the activation function. Unlike in regular convolution, the pixels outside the ROIs do not contribute to the gradient in the ROI convolution layer.

Loss function We use a multi-task loss that is similar to that defined in Mask RCNN [8]: $L = L_{reg} + L_{cls} + L_{seg}$ where the bounding box regression loss L_{reg} and the bounding box classification loss L_{cls} are combined with the segmentation loss L_{seg}, which is the cross-entropy loss for all the pixels inside the ROIs. Different from the loss in [8], our loss is defined on the full image as opposed to on each ROI, which avoids considering overlapping pixels multiple times.

3. EXPERIMENTS

We demonstrate the performance of our method on segmentation of acetabulum bone from ultrasound images.

Dataset Experiments were performed on a clinical dataset of 3D ultrasound images of infant hips for diagnosis of developmental dysplasia of the hip (DDH). The segmentation of acetabulum is an essential step towards making diagnosis of DDH fully automated and less reliant on the operator. However the task is very challenging due to the highly noisy nature of ultrasound images, compared to CT or MRI images. Also the classes are highly imbalanced: the acetabulum only takes up a small portion of the image (0.3 percent in our dataset).

Data collection was performed at the Radiology Department at University of Alberta. The dataset consists of 49 three-dimensional ultrasound scans that has manually annotated segmentation labels for acetabulum provided by the lead pediatric musculoskeletal radiologist. We randomly divide them into a training set of 30 scans and a testing set of 19 scans. The scans from the same patient either all go into the training or testing set. We crop some of the scans to make all the 2D slices share a unified size of 367x192 pixels (height by width). Most 3D volumes contain 256 slices.

Metrics We evaluated the segmentation on 2D slices using the following metrics: precision = TP/(TP+FP), recall = TP/(TP+FN), dice score = 2TP/(2TP+FP+FN), where TP, FP, FN denote True Positive, False Positive, and False Negative. The results were the mean scores for all the 4864 test slices.

Implementation details We used the same VGG16-based RPN as in Faster R-CNN [10], except that we reduced the padding of the first convolutional layer from 100 to 50. The network was trained using SGD with a momentum of 0.9 and weight decay of 0.0005. Learning rate was set to 1e-5, and reduced by a ratio of 0.1 for every 50k iterations. We tried to keep the parameters the same as those in the original work to make a fair comparison. The implementation was based on Caffe [12], where we added the implementation of ROI convolution layer. We compared our method with its FCN counterpart FCN-32s, and U-Net. We used the original Caffe implementation for FCN-32s. For U-Net, we tried three different implementations: the original Caffe implementation, an implementation from the 3D-U-Net [13], and a Tensorflow [14] implementation. The best result was reported.

Results The test scores are summarized in Table 1. Note that the baseline method "FCN-noPool5-scratch" in the table represents the RPN-FCN network trained without the detection loss or the ROI convolution, i.e., with segmentation loss only.

Our method RPN-FCN achieves better segmentation accuracy than the other methods. The distribution of the dice score are shown in Fig. 2 by sorting the slices that contain positive labels based on their dice score. RPN-FCN yields better dice scores on a wide range of slices. Adding RPN on top of FCN-32s confines the segmentation problem in a smaller region. The result suggests that it is an effective way to train the downstream segmentation layers.

Our network is similar to FCN-32s in that the backbone is both VGG16 and the upsampling part is similar. The major difference lies in the use of ROI convolution after 'conv5'

Table 1: Comparison of our proposed method **RPN-FCN** with FCN-32s, U-Net and the baseline method FCN-noPool5-scratch (RPN-FCN without using the detection loss or ROI convolution in training). Scores are averaged over all the test slices. **Bold** numbers denote the best of these methods. For all the methods, the best results from hyper-parameter tuning and different implementations are reported.

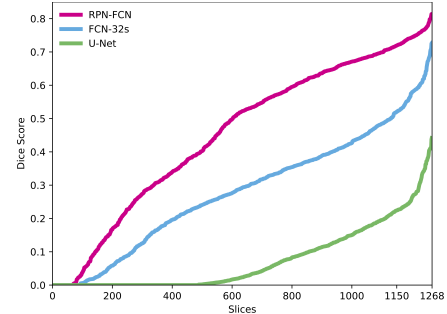| Method	Precision	Recall	Dice Score
RPN-FCN	**0.297**	**0.551**	**0.386**
U-Net	0.039	0.063	0.049
FCN-32s	0.182	0.285	0.223
FCN-noPool5-scratch	0.204	0.259	0.228

Fig. 2: Dice score of all the test slices that contain positive labels, sorted in ascending order of dice score.

layer of VGG16. Due to the removal of the pooling layer, the feature map has a higher resolution. It only needs to be upsampled by a factor of 16 instead of 32 as in the original FCN. Upsampling from a higher resolution feature map helps to retain the features of small objects.

The dice score is low compared to other medical image segmentation tasks. The main reason is that the object of interest is small. The representations learned by the network are dominated by the majority class: the background. U-Net introduced class weighting in the loss to combat this issue. We tried weighting by the class frequency but did not see improvement. The best result for U-Net was in fact obtained from the 3D U-Net implementation of the 2D network, where the input was a 2D image patch instead of the full image. It might be possible to achieve better performance with U-Net by intensive parameter tuning on the class weights or the solver. But our main point here is that small object segmentation poses challenges on full-image-based methods. The detection mechanism in our proposed network offers an effective means for alleviating the issues in training the network. We visualized the segmentation results of some slices in Fig. 3. There is no visible prediction for U-Net since it predicted all pixels as background. Our network was able to produce smooth regions when the shape is relatively clear in the view.

4. CONCLUSION

We presented a novel method for improving the segmentation accuracy of fully convolutional networks. Drawing inspira-
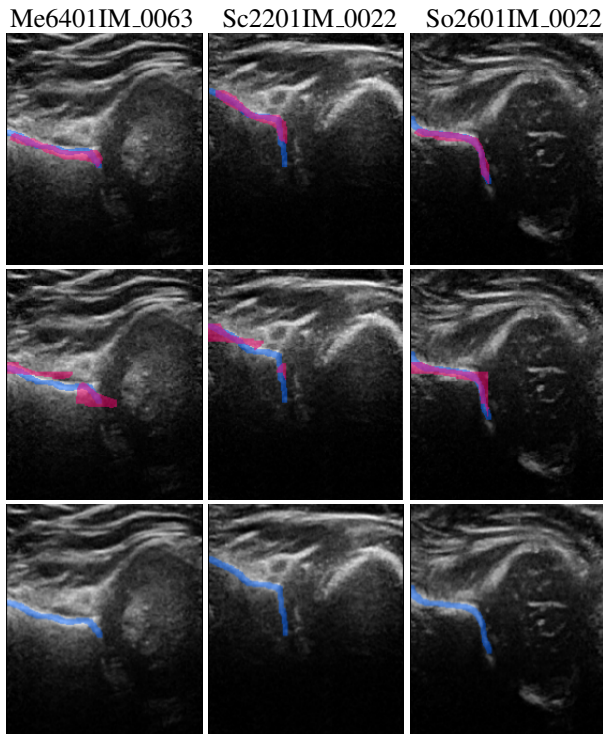
Fig. 3: Segmentation of slices on different volumes. Blue masks denote the ground truth; purple denotes the prediction. The heading in each column is the name of the volume containing the slice. The three rows correspond to the results of RPN-FCN, FCN-32s, U-Net, respectively.

tion from the object detection literature, we incorporated the Region-Proposal-Network (RPN) which was part of the backbone of Faster R-CNN. The bounding box prediction from RPN was used as an extra input to the downstream convolutional layers for segmentation to improve its localization capability. We introduced a ROI convolution layer that directly convolved on all the ROIs in one pass without the need of ROI pooling or region-wise processing, and showed that this layer was trainable through backpropagation. We essentially broke down the segmentation task into two steps: first localizing a rough region in which the object had been detected, then performing segmentation in the region. The proposed network was trained end-to-end on an ultrasound image segmentation task. We showed that by breaking down the segmentation problem into a joint detection and segmentation process, the segmentation accuracy was improved.

5. REFERENCES

[1] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, 2015, pp. 3431–3440.

[2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[3] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference*, 2017, pp. 1175–1183.

[4] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *ICCV*, 2015, pp. 1635–1643.

[5] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *CVPR*, 2015, pp. 3992–4000.

[6] J. Uijlings, K. Van De Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[7] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: A real-time deep network for scene understanding," *CoRR*, vol. abs/1708.02813, 2017.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *arXiv preprint arXiv:1703.06870*, 2017.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[10] S. Ren, K. He, Ross Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[11] D. Chen, G. Hua, F. Wen, and J. Sun, "Supervised transformer network for efficient face detection," in *ECCV*. Springer, 2016, pp. 122–138.

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 675–678.

[13] Ö. Çiçek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in *MICCAI*. Springer, 2016, pp. 424–432.

[14] M. Abadi, A. Agarwal, Paul Barham, et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.