GRAPH-SEARCH BASED UNET-D FOR THE ANALYSIS OF ENDOSCOPIC IMAGES

Shufan Yang[1], *Sandy Cochran* [1]

[1]School of Engineering, University of Glasgow, Glasgow, UK

CEUR-WS.org/Vol-2366/EAD2019_paper_6.pdf

ABSTRACT

While object recognition in deep neural networks (DNN) has shown remarkable success in natural images, endoscopic images still cannot be fully analysed using DNNs, since analysing endoscopic images must account for occlusion, light reflection and image blur. UNet based deep convolutional neural networks (DNNs) offer great potential to extract high-level spatial features, thanks to its hierarchical nature with multiple levels of abstraction, which is especially useful for working with multimodal endoscopic images with white light and fluoroscopy in the diagnosis of esophageal disease. However, the currently reported inference time for DNNs is above 200ms, which is unsuitable to integrate into robotic control loops. This work addresses real-time object detection and semantic segmentation in endoscopic devices. We show that endoscopic assistive diagnosis can approach satisfy detection rates with a fast inference time.

Index Terms— Endoscopic images, Deep neural networks, Decoder-Encoder neural networks

1. INTRODUCTION

A common strategy in deep convolutional neural network for semantic segmentation tasks requires the down-sampling of an image between convolutional and ReLU layers, and then up-sampling the output to match the input size [1]. Atrous convolution is designed to obtain the spatial resolution after several convolution layers [2]. Although, when compared to normal convolution layers, the atrous convolution inserts holes into its filters, thus enlarging the receptive field to a greater extent, this method often loses low level information, and is therefore unsuitable for a medical environment. To deal with multi-scale images, a new Atrous Spatial Pyramid Pooling(ASPP) layer has been developed to allow the network to work on different image size and thus increase the flexibility of the input scale [3]. Capturing more information, some of networks also directly used the output from convolution layers as the low-level features, passing it into the decoder to increase accuracy [4]. However, these structures currently report an average inference time above 300ms [4]: it is essential to have a fast inference time in order to achieve real-time image analysis.

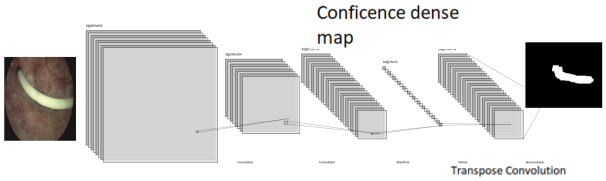


Fig. 1: The architecture of Network

2. METHOD

As shown in Fig. 1, the architecture of network for this challenge is based on the UNet architecture. The convolution network layers are used as an encoder to abstract low level spatial information. A decoder is then implemented using transposed convolution. Instead of using an ASPP layer, a general auto-encoder class label is kept into a dense layer. This compressed feature vector connects to a series of up-sampling layers using the coast mask.

2.1. Algorithm

Regular classification DCNNs generate a coast mask containing probabilities for each class in a dense confidence regions using the following steps:

1 Generate feature map using fully convolutional neural network

2 Initialize a segmentation with feature detected

3 Transpose convolutional using confidence check to keep one weak edge on the common boundary

4 Merge neighbouring regions (Ri and Rj) using an optimal objective function with the confidence of whole image from feature map

5 Generate a new maximum of confidence map through all adjunct regions

Here, the considered objective function is:

$$C_{image} = \frac{\sum_{l=1}^{N_r} C_\beta}{N_\gamma}(1 - P_j)^\gamma, \quad (1)$$

Layer name	Output size	Parameters
Conv-1	H/2, W/2	8×8, stride 2
Max pooling	H/4, W/4	3×3, stride 2
Conv-block-1	H/4, W/4	$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 512 \end{bmatrix}$
Dense-confi-block	H/8, W/8	$\begin{bmatrix} 1 \times 1 & 64 \\ 3 \times 3 & 512 \end{bmatrix}$

Table 1: Network architecture and layers specification.

where $C\beta$ is the current region of confidence and N_γ is the number of region of the corresponding specific adjunct region.P_j is the probability of the j^{th} class. γ is a free parameter which can be used to scale up confidence level to avoid ignoring small regions.

After calculating the dense confidence feature map, the resulting features are fed to a 1x1 convolution kernel with 256 filters. Finally, the result is bi-linearly up-sampled to the correct dimensions. The dense confidence pyramid uses atrous convolutional layers in a cascade fashion, where the dilation rate of each layer increases layer by layer; layers with small dilation rates are located in the lower part, while layers with large dilation rates are located in upper part.

Because of the great imbalance of different classes in this test dataset, some classes have large number of pixels in almost every image and others doesnt exist in some images at all. By setting $\gamma > 0$, we reduce the relative loss for well-classified examples to avoid miss classifying objects. In other words, the dense confidence layer works to alleviate errors using a smaller scaling factor.

2.2. Data Argumentation

Imagnet pre-trained Resnet-50 is used for training with 320 images that EAD2019 challenge provides for the semantic segmentation task [5, 6]. Among those images, 20% is kept for evaluation and the rest is kept for training. The following data argumentation methods are applied: the RGB value (66.32, 76.13, 120.58) is used for normalization with batch size 4. A random flip and rotation with (-50, 50) are used to rescale the picture to 0.5-0.75 of its original size with the pad size of (600 pixel, 512 pixel). After the data augmentation, about 1300 images are obtained which are 4 times larger than the original dataset.

2.3. Training processes

The following table 1 shows the hyper-parameters chosen for feature map abstraction.

We uses a normal distribution to pick a tensor from the interval of (0, std), where the equation of std is:

$$std = \sqrt{(2/((1+a^2)fan_in))} \tag{2}$$

Where, a is the negative slope of the rectifier that used after this layer which is 0 for Relu activation layer.

The typical batch size for SGD is generally set to 6, 12, 24 [7]. However, in this work, the batch size was set to 5, which is the optimal number to strike the tradeoff of GPU memory and speed of training.

During the training process, a poly learning rate policy is implemented on the learning rates. To begin with, the learning rate is relatively high and, after several iterations, the weights have improved and the distance between current and the best weights decreased. Learning rates also become smaller correspondingly to find the best weights. The decay learning rate policy is employed with the formula

$$\eta = \eta \left(1 - \frac{ep}{maxep}\right)^{power}, \tag{3}$$

where ep and maxep are the current epoch and the maximum epoch, which is set to 500. Here the power is set to 0.9 based on previous published method [8]. Since the training dataset includes some of very similar data, a weight decay method [10] is followed the equation 3 and equation 4.

$$R(w) = \sum k \sum l \, w_{k,l}^2, \tag{4}$$

where $w_{k,l}$ is the weights stored in the network. The total loss from the loss function will now have two parts:

$$L(w) = \frac{1}{N} \sum_{i=1}^{N} L_i \left(f\left(x_i, w\right), y\right) + \lambda R(w) \tag{5}$$

The first term represents the loss calculated by the loss function chosen; the second part is the regulation part, making the network more simple. If two sets of weights all have a similar loss calculated by the loss functions, the bigger weights will have a bigger regular term and therefore has a bigger total loss.

3. EVALUATION

3.1. Sematic segmentation results

Results obtained from the trained models of challenge validation set are listed in Figure 2. The various resolution im-ages are shown from the top to bottom row: (1003 x 1003pixel, 628 x 628 pixel, 576 x 576 pixel).

A detailed example of segmentation results for 5 classes from the endoscopic dataset is shown in Appendix[9].

3.2. Training process

Figure 3 shows the loss rates at validation epoch. Although the evaluation processes was not as good as the loss at training epoch, it was still acceptable. The MIoU curve dramatically increase during the initiative 30 epochs, but then slowly converged to the final value, achieving 65%.

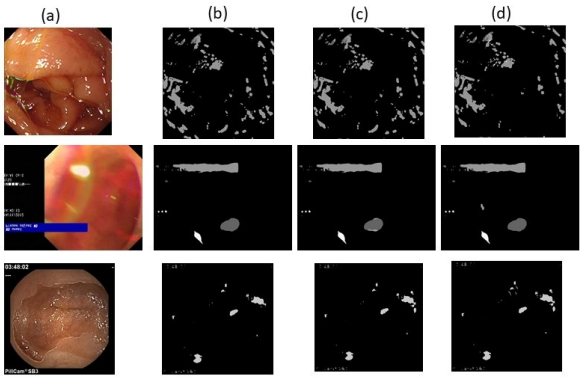
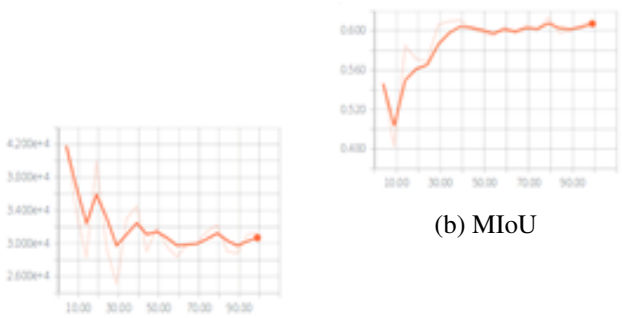


Fig. 2: Results obtained from the validation set are listed using various grey scales for five classes: Instrument(255), Specularity(204), Artefacts(153), Bubbles(102), Saturation(51). From left to right: (a) input (b) Unet (c) DeeplabV3+ (d)Unet-D



(a) Loss rates

(b) MIoU

Fig. 3: The train process performance at the loss rates and MIoU at each evaluation epoch

3.3. Comparison

Our evaluation was implemented using the validation set. We use the Mean Intersection over Union to evaluate the capacity of the model:

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij}+\sum_{j=0}^{k}p_{ji}-p_{ii}} \quad (6)$$

The prediction (p_{ii}) was made by finding the maximum output features map of the segmentation model, and is upsampled by 8 using bilinear interpolation. As shown in Fig. 4, our approach (UNet-D) had very similar performance in training compared with state of the art semantic segmentation methods. In this challenge, the rules for evaluation of segmentation was based on the DICE and Jaccard value. Our results achieved the same results as other technologies, shown in Table 2 and Table3.

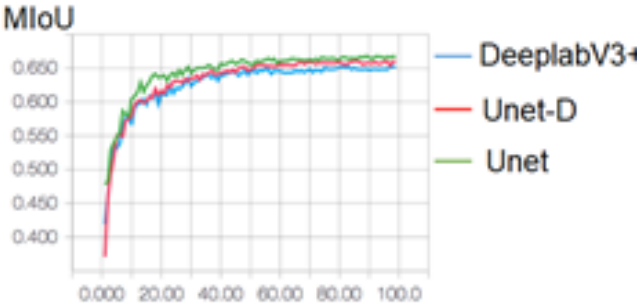


Fig. 4: The comparison among DeeplabV3 , UNet, UNet-D(our proposed approach)

Method	Over-lap	F2-Score	Score_s
UNet	0.36	0.48	0.42
Deeplab_v3+	0.54	0.56	0.55
UNet-D	0.39	0.44	0.41

Table 2: Sematic Segmentation score in the EAD2019 Challenge

Model	Training time	Prediction time	Size
UNet	20h	213.5ms	28.7MB
Deeplab-V3+	40h	320.8ms	182.7MB
UNet-D	30h	126.3ms	23.2MB

Table 3: The comparison of training and inference performance

However, the measurement is an inadequate measurement for semantic segmentation. Since the DICE calculate is based on binary cases, this means that no cross regions appeared in multiple classes. Furthermore, the $score_s$ is in favor with the high DICE value.

The experiment environment used was Windows 10, 64-bit with an Intel Core i7-7700HQ CPU and GeForce GTX 1080 Ti. The number of inferences to calculate the average result was 20. Although the UNet-D network does not have the best performance in terms of its $score_s$ value in the EAD2019 challenge [5], it had a smaller computational footprint, making it an excellent candidate for real-time semantic segmentation tasks.

4. CONCLUSION

This work demonstrates that a skipped connection, keeping low level spatial information, and removing the connection with the ReLu layer, using a confidence relay, can reduce the inference time. The UNet-D performance was not, however, outstanding at this challenge; part of reason was that we use small batch size to keep system memory low. Using the PASCAL VOC2012 dataset, 85% MUOI was reported at the evaluation processes. With careful data argument methods, the

sematic segmentation based on deep convolution neural network has great potential to be used in the real-time control loop for the next generation of endoscopic devices.

5. REFERENCES

[1] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, vol. 9351 of *LNCS*, pp. 234–241, Springer.

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018.

[3] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang, "Learning a discriminative feature network for semantic segmentation," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 1857–1866.

[4] Tobias Pohlen, Alexander Hermans, Markus Mathias, and Bastian Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 3309–3318.

[5] Sharib Ali, Felix Zhou, Christian Daul, Barbara Braden, Adam Bailey, Stefano Realdon, James East, Georges Wagnires, Victor Loschenov, Enrico Grisan, Walter Blondel, and Jens Rittscher, "Endoscopy artifact detection (EAD 2019) challenge dataset," *CoRR*, vol. abs/1905.03209, 2019.

[6] Sharib Ali, Felix Zhou, Adam Bailey, Barbara Braden, James East, Xin Lu, and Jens Rittscher, "A deep learning framework for quality assessment and restoration in video endoscopy," *CoRR*, vol. abs/1904.07073, 2019.

[7] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, 2013, pp. 1139–1147.

[8] Anders Krogh and John A. Hertz, "A simple weight decay can improve generalization," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 4*. 1992, pp. 950–957, Morgan Kaufmann.

[9] Sharib Ali, Felix Zhou, Christian Daul, Barbara Braden, Adam Bailey, Stefano Realdon, James East, Georges Wagnires, Victor Loschenov, Enrico Grisan, Walter Blondel, and Jens Rittscher, "Endoscopy artifact detection (ead 2019) challenge dataset," 2019.

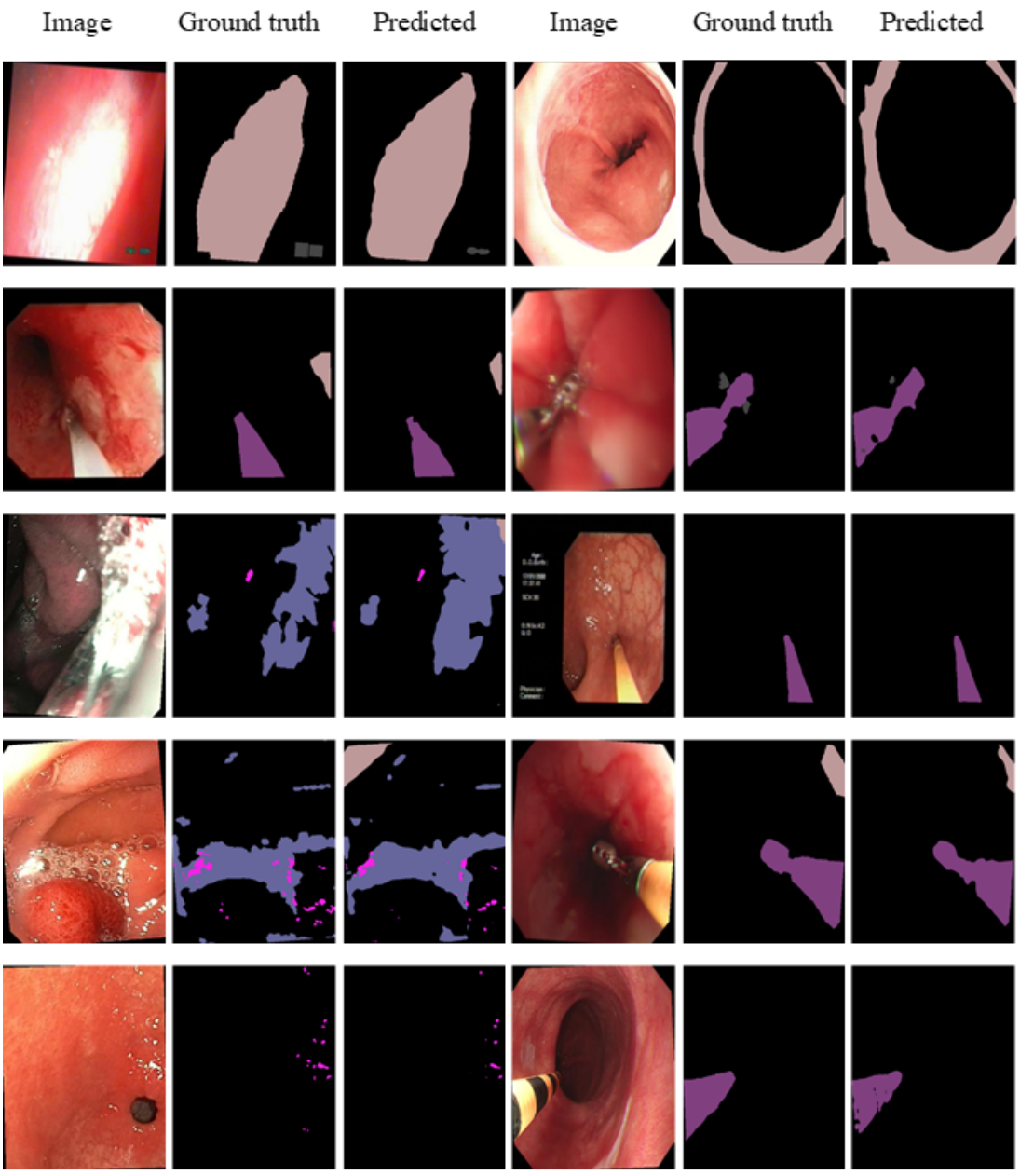


Fig. 5: Sample semantic segmentation results for five classes