

Live Tumour Segmentation with Magnetic Resonance (MR) Image

Tachi Aso

Paul Ou

Department of Automation, Tsinghua University

Abstract

Convolutional Neural Network (CNN) is generally used in image recognition. For image segmentation, several approaches, which are CNN models with extra process, have been developed in recent years. This paper introduces three major algorithms for image segmentation: FCN, SegNet, and U-Net, and evaluates segmentation methods of liver tumor with magnetic resonance (MR) images using them.

1. Introduction

Accurate labelling of regions of interest, known as semantic segmentation, is highly useful in many fields such as autonomous vehicles, geo sensing and bio medical image diagnosis. In this project, we adopt three different segmentation algorithms which are U-Net, SegNet, and Fully Convolutional Network (FCN), and investigate the performance to an in-house MR image dataset consisting of 40 patients, in particular the arterial phase, to address the missing annotation challenge. These algorithms are based on convolutional neural network, and the difference lies in the final layers. In convolutional neural network, the last few layers are full connected layers and produces a one-dimensional vector for classification (patch classification). On the other hand, U-Net, SegNet, and FCN has an extra decoding process corresponding to the encoding process, which is accomplished by upsampling, hence producing a two-dimensional image that preserves the location information.

2. Method

The task required in segmenting tumour from magnetic resonance image is to label each pixel of an image with a corresponding class, which is commonly referred to dense prediction. The difference between dense prediction and the previous tasks accomplished with simply convolutional neural network is that the expected output in semantic segmentation is a high-resolution image with corresponding label for each pixel, rather than a one-dimensional vector.

In a typical convolutional network, the size of image decreases due to pooling which helps the filters in the deeper layer to focus on a larger receptive field. Intuitively speaking, by down sampling the model can make a good prediction on “what” is in the image but loses information on “where” it is in the image.

This is where fully convolutional network comes in (2014), which is one of the prominent pioneers in deep learning architecture. It does dense prediction without any full connected layers. This allows segmentation maps to be generated for image of any size and much faster. Segnet, u-net and many other similar segmentation models based upon it. Simply put, it introduces up sampling mechanism to convert a low-resolution image to high resolution image to recover the “where” information. Transposed convolution (also known as deconvolution) is probably the most preferred technique to perform up sampling, which learns parameters through back propagation.

- FCN-32s: $deconv(featureMap_{16_{16}}, stride = 32) \rightarrow output: 512_{512}$

- FCN-16s: $temp = add(deconv(featureMap_{16_{16}}, stride = 2), pool4)$

$deconv(16s: temp, stride = 16) \rightarrow output: 512_{512}$

- FCN-8s: $temp1 = add(deconv(featureMap_{16_{16}}, stride = 2), pool4)$

$temp2 = add(deconv(temp1, stride = 2), pool3)$

$deconv(temp, stride = 8) \rightarrow output: 512_{512}$

U-Net consists of two paths as well. First path is the contraction path (encoder) is simply convolutional neural network with convolutional and max pooling layers. The

second path is the symmetric expanding path (decoder) to enable precise localization using transposed convolutions, which is performing concatenation on image with the same dimensions originally and after deconvolution.

SegNet adopts pooling indices. This is because location information is loss at pooling layer, so saving pooling indices can help deconvolution layer to recover location information more easily. This makes SegNet more memory efficient than FCN.

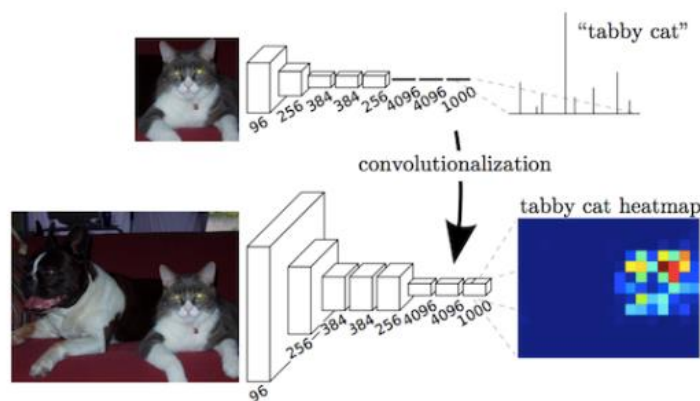


Figure 1. FCN

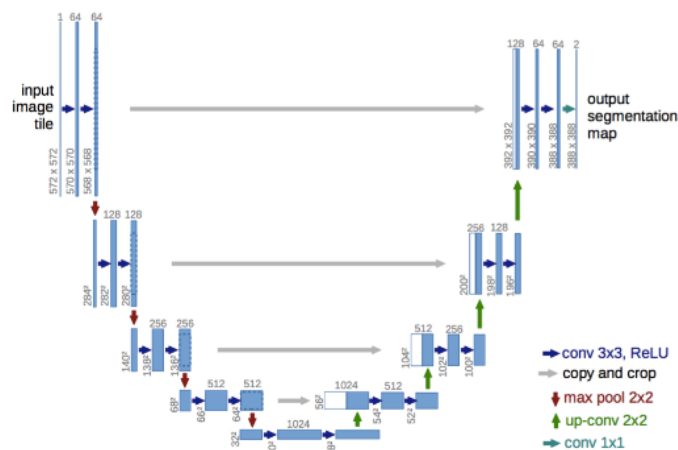


Figure 2. U-Net

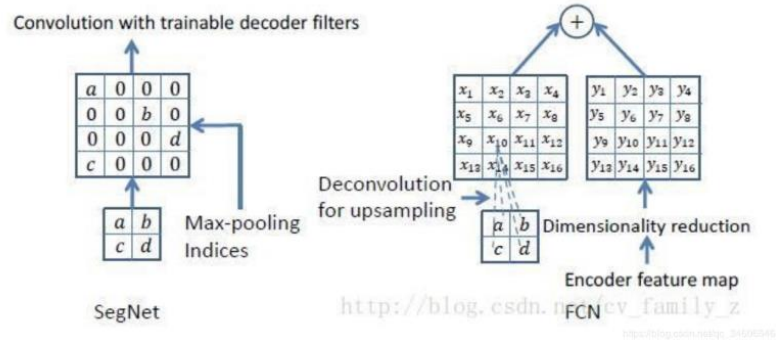


Figure 3. SegNet

3. Experiments and Conclusion

As it is explained above, three segmentation approaches, which are FCN, SegNet, and U-Net, are built and trained with MR images of the artery phase from 40 patients. As a loss function, dice coefficient is used. The formula of dice coefficient is defined as:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

where X = desired output, Y = predicted output

In this experiment, the image data are normalized when converted a NumPy array and used as an original image size as inputs and outputs which is 320×260 pixel. The time to complete each epoch among the models is totally different. For example, FCN does not take much time to train one epoch, but it takes a huge number of epochs to improve the accuracy on the training set. As the table 1 shows, the dice coefficients on the training set by FCN, SegNet, and U-Net were 10, 0.9720, and 0.8888 respectively and on the test set, 10, 0.8180, and 0.7265.

Table 1. Dice Coefficient on Training and Test set

	FCN	SegNet	U-Net
Training Set	0.6836	0.9720	0.8888
Test Set	0.5307	0.8895	0.7424

As it is expected, since SegNet and U-Net is based on FCN and improved, they have higher efficiency. SegNet and U-Net get better results than FCN. The structure of SegNet and U-Net is similar. The main differences are when it decodes, SegNet decode with uppooling but U-Net concatenates corresponding layers, and SegNet has Batch Normalization after every convolution layer. Batch Normalization allows to use higher learning rates and be less careful about initialization. Hence it might be the reason of approximate 0.1 gap of dice coefficient on both of training and test set between SegNet and U-Net. Figure 4 shows one of the test images and its desired and predicted output by the models. SegNet and U-Net sufficiently detected where the tumour is. However, in this experiment, we use data from only 40 patients, which contains 720 images having a liver tumour in total. It is few training datasets for deep learning, therefore for further improvement, an approach such as adding some noise or making some transformations would be needed to increase the number of training set.

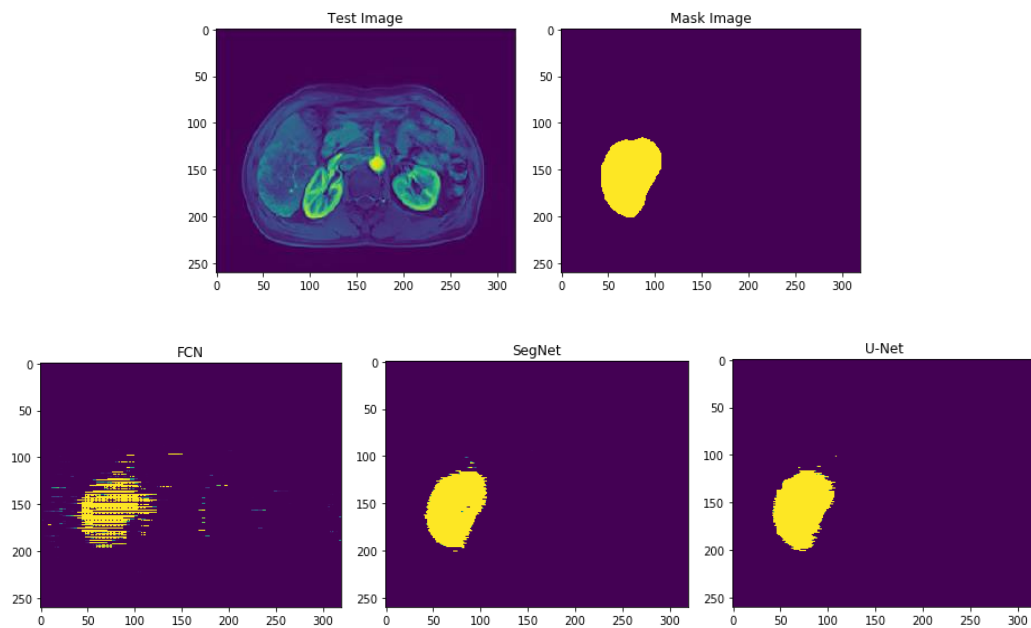


Figure 4. Test image, Desired image, FCN result, SegNet result, U-Net result

References

Chilamkurthy, S. (2017, July 5). A 2017 Guide to Semantic Segmentation with Deep Learning. *Qure.ai Blog*. Retrieved from <http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review>.

Sørensen Dice coefficient. (2019, December 26). *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Sørensen–Dice_coefficient.

Lamba, H. (2019, February 19). Understanding Semantic Segmentation with UNET. *Medium*. Retrieved from <https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>.

Lofe, S, & Szegedy, C. (2015, Feb 11). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.

Mody, P. (2018, March 6). List of Semantic Segmentation Models for Autonomous Vehicles. *Playment Blogs*. Retrieved from <https://blog.playment.io/semantic-segmentation-models-autonomous-vehicles/>.

Sancinetti, F. (2018, August 29). U-NET ConvNet for CT-Scan segmentation. *Medium*. Retrieved from <https://medium.com/@fabio.sancinetti/u-net-convnet-for-ct-scan-segmentation-6cc0d465eed3>.

Shah, M (2017, June 1). Semantic Segmentation using Fully Convolutional Networks over the years. *int const change*. Retrieved from <https://meetshah1995.github.io/semantic-segmentation/deep-learning/pytorch/visdom/2017/06/01/semantic-segmentation-over-the-years.html>.

语义分割——对 FCN、U-Net、SegNet 的一点理解, (2019, April 21). *CSDN*. Retrieved from https://blog.csdn.net/qq_34606546/article/details/89434487.