The Detection of the Plasma Boundary in Tokamak

Zishu Liu Tingzhi Chang*
Tsinghua University, Department of Engineering Physics
Beijing, China
zs-liu16@mails.tsinghua.edu.cn

Abstract

We use the FCN to deal with the image of discharging plasma and try to get the boundary of the plasma. However, although the training accuracy seems high in some way, the network doesn't work as we expect on the origin image.

1. Introduction

1.1. Problem description

When the plasma is constrained a magnetic field and reaches a etremely high temperature, nuclear fusion can occur. Because of the high temperature ($> 10^7$K), we usually use fast CCD to catch the image of plasma.

SUNIST is a small Tokamak located in Tsinghua University. In this project, we aim at using CNN to detect the edge of plasma during the discharge process.

Moreover, if more time is given, we will also try to fit several important parameters of discharge process.

1.2. Overall plan

We view the edge of plasma as the bound which divides the inner space(i.e. the plasma) and outer space. Then the problem converts to a segmentation problem. After that, we can use network like FCN to deal with it. Cause labeled images doesn't exist, we first label some images manually for training. We don't prepare data for testing, but use the network to process a video and check it manually.

2. Related work

2.1. Past research

Scientists have tried lots of traditional ways to deal with the discharge processin EAST, which is a huge Tokamak located in AnHui. A lots of traditional ways, such as Canny Edge Detect, and Active ContourModels. However, some of them need to fix some points in advance, the others don't always perform well.

In this paper [3], the author comes up with an improved edge detect algorithm, which works much better than the past model. However, the size of SUNIST is much smaller than of EAST. And when the discharges nearly ends, the boundary comes much more blur.

2.2. Network structure

The paper [1] shows that FCN is good at object segmentation problem and it works well on dataset like PASCAL VOC(mean IU $\sim$ 62.7%). It uses VGG-16 as the classification convolution-net and replaces the fully-connected-linear-layer with fully-convolution-layer. It also adds unsampled layer to make the result consistent with the label.

3. Project framework

3.1. Segmentation architecture

Cause the characters of the plasma boundary are not very complex and there are only two types of object, the depth of the network isn't very deep but the kernel of convolution layer is little larger.

3.2. Unsampled layer

The unsampled layer resize the heat image to the origin image. Cause we use convolution layer before, we use transposed-convolution layer to resize the image. And we view the stride of the transposed-convolution layer as the amplification factor. To make the training process quicker, I use the bilinear kernel to initialize the weight of bilinear kernel, which is indicated as the

*Not enrolled in this course, but gives suggestions to the project. Also, the course teacher and the tutor give lots of useful suggestions to this project.

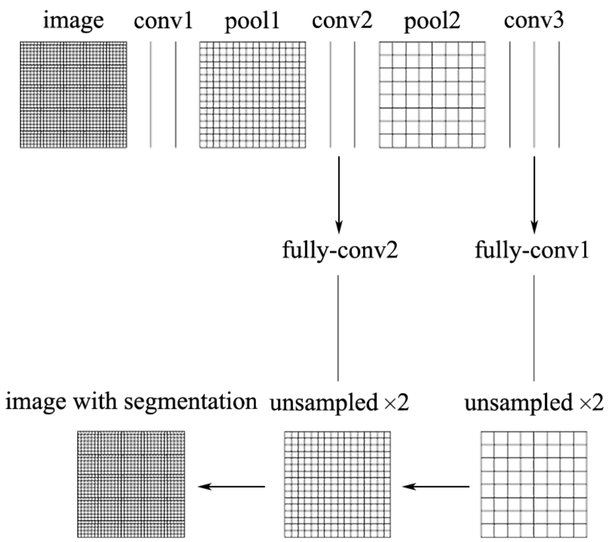

Figure 1. Network Structure

equation (1).

$$
\begin{aligned}
image(x, y) = (y - b)((x - a)image[a + 1][b] \\
+ (1 + a - x)image[a][b]) + (b + 1 - y) \\
((x - a)image[a + 1][b + 1] \\
+ (1 + a - x)image[a][b + 1])
\end{aligned}
\tag{1}
$$

where (x, y)(on the origin image) refers to the corresponding point of (a, b)(on the head image).

3.3. Accuracy and loss

Since we only have two types, we use the simple accuracy, although sometimes it can't reflect the real situation. To calculate the loss, we first map $log-softmax$ on the output and then use the nill-loss layer, which is indicated as the equation (2).

$$
\begin{aligned}
l_n = -w_{y_n} \sum_{i=1}^{K} x_i^n \ln \frac{\exp\{y_i\}}{\sum_j \exp\{y_j\}} \\
l(x, y) = \sum_{n=1}^{N} \frac{1}{\sum_{n=1}^{N} w_{y_n}} l_n
\end{aligned}
\tag{2}
$$

where x_i refers to ground truth, y_i refers to output data, K refers to the number of categories, N refers to the minibatch-size, weight $w_{y_n} \equiv 1$.

4. Experiment

4.1. Dataset

We label 100 pictures manually and then use opencv2 to crop image to proper size (500×500) for training. To enlarge the dataset, we also flip the images. Considering the small size of dataset, we don't keep extra data for validation or testing.

4.2. Evalidation

To evalidate the network, we use the network to process with the given video, and evalidate the video manually.

4.3. Result

Generally, we try two network, one of them has one layer more than the other.

The first network parameter and structure are shown as the follows:

- learning rate: 0.001
- weight decay: 0.001
- epoch: 9
- batch size: 20
- convolution layer number: 3
- unsampled layer: the last two convolution layers

This network gets better result and higher training accuracy($\sim 90\%$). The training loss and accuracy is shown as the figure 2 and 3. Mention that the "epoch" axis refers to the double number of epoch (because we record loss and accuracy twice in each epoch.)

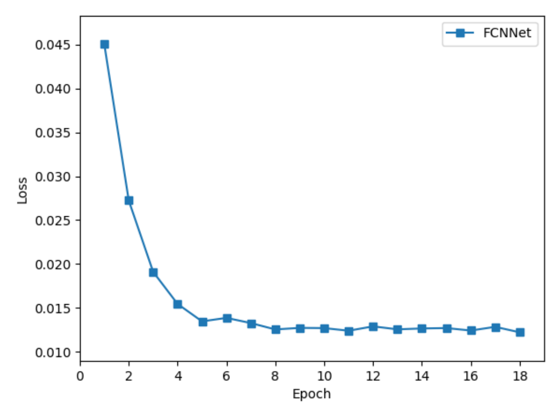

Figure 2. Training Loss

The second network parameter and structure are shown as the follows:

- learning rate: 0.001
- weight decay: 0.001
- epoch: 8

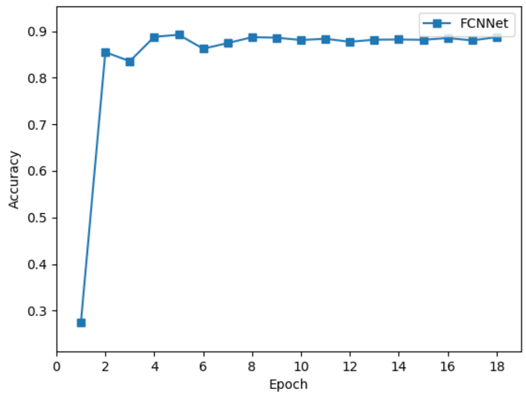

Figure 3. Training Accuracy

- batch size: 20
- convolution layer number: 4
- unsampled layer: the last two convolution layers

Although this network is deeper, it doesn't get better result and higher training accuracy($\sim 80\%$). The training loss and accuracy is shown as the figure 4 and 5.

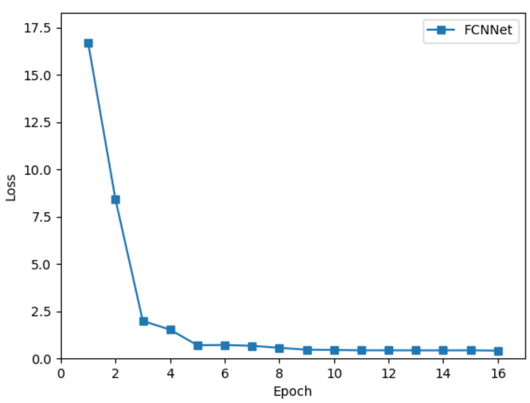

Figure 4. Training Loss

The two network both converge before 10 epochs, and both seems get "high" accuracy after training. Here we give one image after processing by the network as the figure 6. Clearly, compare to the left side, although the right side successfully detects the outer boundary, but it doesn't give the inner boundary, which is actually the boundary we most care of.

Moreover, cause we don't use the prior in the FCN, which the boundary should be a circle and the plasma should be continuous, the plasma we detect has some "holes".

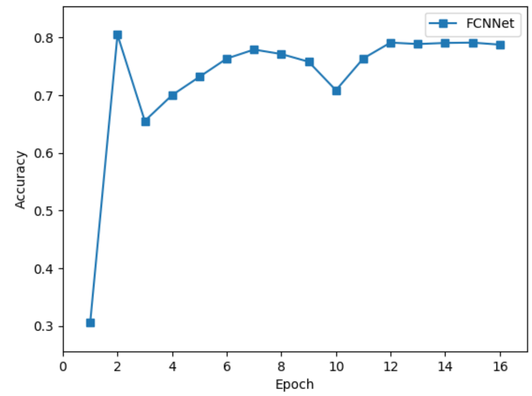

Figure 5. Training Accuracy

Figure 6. Image After Processing

5. Conclusion

First, the deeper network doesn't work better than the one is more shallow, which indicates that deeper network won't help us more in this project. The reason of it may be that information gets lost when pasts several layers and vanishing gradient appears.

Second, we don't put the prior into the network. The boundary of the plasma can be described as the equation (3). But it seems hard to put this prior into the FCN.

$$\begin{aligned} x &= x_0 + a\cos\left(\theta + \delta\sin\theta\right) \\ z &= \kappa a \sin\theta \end{aligned} \tag{3}$$

where $a, x_0, \theta, \delta, \kappa$ are parameters of the parametric equation, and the point (x, z) refers to a point on the boundary of plasma..

The paper [2] combines CRF with RNN to get a better result, may be a solution for the prior. However, due to the limit of knowledge and time, we don't try this method.

Finally, the most important problem is that we don't have enough data for training and testing. What's more, the data is all labeled by human, which is not 'accurate' in some way.

References

[1] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(4):640–651, 2014.

[2] Z. Shuai, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, H. Chang, and P. H. S. Torr. Conditional random fields as recurrent neural networks. 2015.

[3] H. Zhang, B. Xiao, Z. Luo, H. Qin, J. Yang, and D. Weldon. Reconstruction of the plasma boundary of east tokamak using visible imaging diagnostics. *IEEE Transactions on Plasma Science*, 46(6):2162–2169, 2018.