Semantic segmentation using full convolutional network

Fangzhou Ai
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
faai@eng.ucsd.edu

Yue Qiao
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
yuq021@eng.ucsd.edu

Zunming Zhang
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
zuz008@eng.ucsd.edu

Abstract

In this report, we explore the full convolutional network (fcn) and use full convolutional network to perform semantic segmentation task on Indian driving dataset. Using our baseline dataset, we archive pixel accuracy of 0.79 and average IOU of 0.33.

1 Introduction

In the report we try to solve the sematic segmentation task using full convolutional network. The goal of semantic segmentation is to label each pixel of an image with a corresponding class of what is being represented [5]. To solve the task, we utilize full convolutional network. In full convolutional network, there is no fully connected layer, only convolutional layer. In the full convolutional network structure, we use Xavier weight initialization. Deep neural network suffers from weight vanishing problem. If we use normal weight initialization methods like uniform distribution, the weight will vanish quickly during matrix calculation. Therefore, we introduces Xavier weight initialization [3], which can help us eliminate the problem. To further improve the training speed and stability of the network, we introduce batch normalization layer [4]. The batch normalization layer solves that the distribution of each layer's inputs changes during training by normalizing layer inputs.

Preprint. Under review.

2 Related Work

3 Methods

3.1 Baseline

The baseline model is an encoder-decoder structure. It first encode the picture using convolution layers and then decode the segmentation map using deconvolution layers. In the baseline model, we use batch normalization and use ReLU as the activation function. For the last layer, we use softmax to get the classify results.

3.2 Improved baseline

3.3 Custom model

Inspired by this paper[2], we replace the encoder part in FCN model with the RRCNN layer instead, the RRCNN represent better abstraction ability, thus we would expect better IoU and precision. However the RRCNN is really memory-intensive, we have to crop the image to only 256 *256 size and delete 3 layers from original model's encoding part, which severly influence the performance of our model. What's worse is the RRCNN encoding part contains maxpooling operation after each RRCNN block, which means the images size would shrinks even more, since we have already started from a relative small size, maxpool will almost ruin all the details in the feature map.

3.4 Transfer learning

In this part we employ the DEEPLABV3-RESNET101 model, which is a default pre-trained net work in PyTorch, from the website[1] we know that this model is constructed by a Deeplabv3 model with a ResNet-101 backbone. The pre-trained model has been trained on a subset of COCO train2017, on the 20 categories that are present in the Pascal VOC dataset. We modifier the classifier part so that it could match the number of our feature, and we freeze the parameters of the pretrained part to accelerate the whole procedure.

3.5 U-Net

Folowing the routine demostrated in the U-net paper [6], we build the U-net by our self. The obvious difference is the U-net convolution block contains 2 convolution of the same size, while the FCN model only contains 1 convolution operation. We notice that the U-net model is also memory intensive while the datahub only give us a GTX 1080 ti with 11GB memory, hence there's a trade-off of between batch size, image size and model's completeness. We decide to crop the image and shrink the batch size to run the whole model, though this might not be the best solution, but this could give us a flavor of how the complete network looks like.

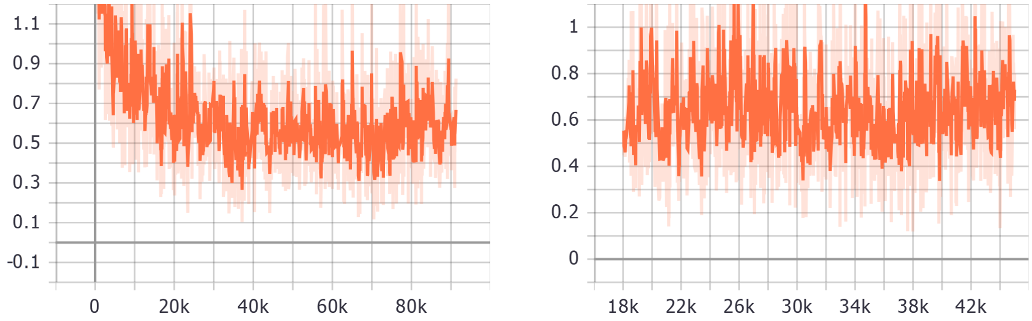
4 Results

Table 1: Pixel accuracy and IOUs for different models

Model	Pixel accuracy	Mean IOU	road IOU	sidewalk IOU	car IOU	billboard IOU	sky IOU
Baseline	0.7944	0.3371	0.8632	0.2323	0.5084	0.2419	0.9403
Improved baseline	TBA	TBA	TBA	TBA	TBA	TBA	TBA
Custom	TBA	TBA	TBA	TBA	TBA	TBA	TBA
Transfer Learning	TBA	TBA	TBA	TBA	TBA	TBA	TBA
U-Net	TBA	TBA	TBA	TBA	TBA	TBA	TBA

4.1 Baseline model

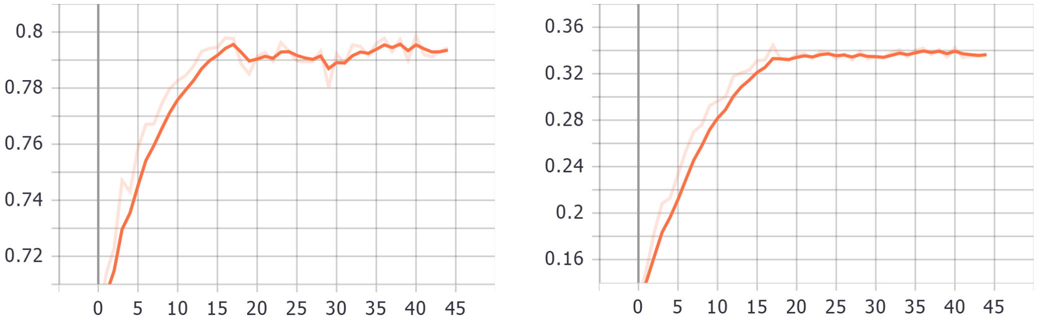
Training and Validation loss for baseline model is in Figure 1. Training and Validation pixel accuracy and mean IOUs for baseline model is in Figure 2. Various IOUs for baseline model is in Figure 3.



(a) Training loss

(b) Validation loss

Figure 1: Training and Validation loss for baseline model



(a) Pixel accuracy

(b) Mean IOUs

Figure 2: Training and Validation pixel accuracy and mean IOUs for baseline model

4.2 Improved baseline model

4.3 Custom model

4.4 Transfer learning

4.5 U-Net

5 Discussion

6 Individual contributions to the project

Fangzhou Ai Custom model, Transfer learning and U-Net model.

Yue Qiao Evaluation metrics, Dataset loading and Baseline Model.

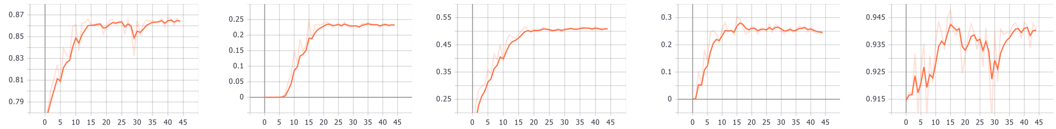
Zunming Zhang

References

[1] Deeplabv3-resnet101 | pytorch. `https://pytorch.org/hub/pytorch_vision_deeplabv3_resnet101/`. (Accessed on 02/15/2021).

[2] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation, 2018.

[3] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth*



(a) Road IOU (b) Sidewalk IOU (c) Car IOU (d) Billboard IOU (e) Sky IOU

Figure 3: Various IOUs for baseline model

International Conference on Artificial Intelligence and Statistics, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. JMLR Workshop and Conference Proceedings. URL `http://proceedings.mlr.press/v9/glorot10a.html`.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[5] Jeremy Jordan. An overview of semantic image segmentation. `https://www.jeremyjordan.me/semantic-segmentation/`, 5 2018. (Accessed on 02/15/2021).

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.