
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

TBA

1 Introduction

2 Related Work

3 Methods

3.1 Baseline

3.2 Experimentation

4 Results

4.1 Baseline model

4.2 Improved baseline model

4.3 Custom model

Inspired by this paper, 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 severely 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.

4.4 Transfer learning

In this part we employ the DEEPLABV3-RESNET101 model, which is a default pre-trained net work in PyTorch, from the official document 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.

4.5 U-Net

Following the routine demonstrated in the U-net paper, 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.

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