Deep Learning Lab

Exercise 3:Decoder Encoder Networks(Image-Segmentation)

Manav Madan (01.12.2018)

Introduction:

In this exercise, a Decoder network was build, which involved understanding the concepts of a upsampling(decoder module) of Fully Connected Networks(FCN's). The idea is to identify labels pixel wise, so that an image could be segmented according to the different labels given to different pixels. The used data set is the Camvid semantic segmentation dataset. The basic API of Tensorflow library is used. Which is a Deep learning framework from Google. The encoder networks which involves downsampling of image or converting an image to an higher abstract level(Feature-Maps) using convolution operations. The task was to built an upsampling model(Decoder module) using transpose convolution ,train the model and test it.

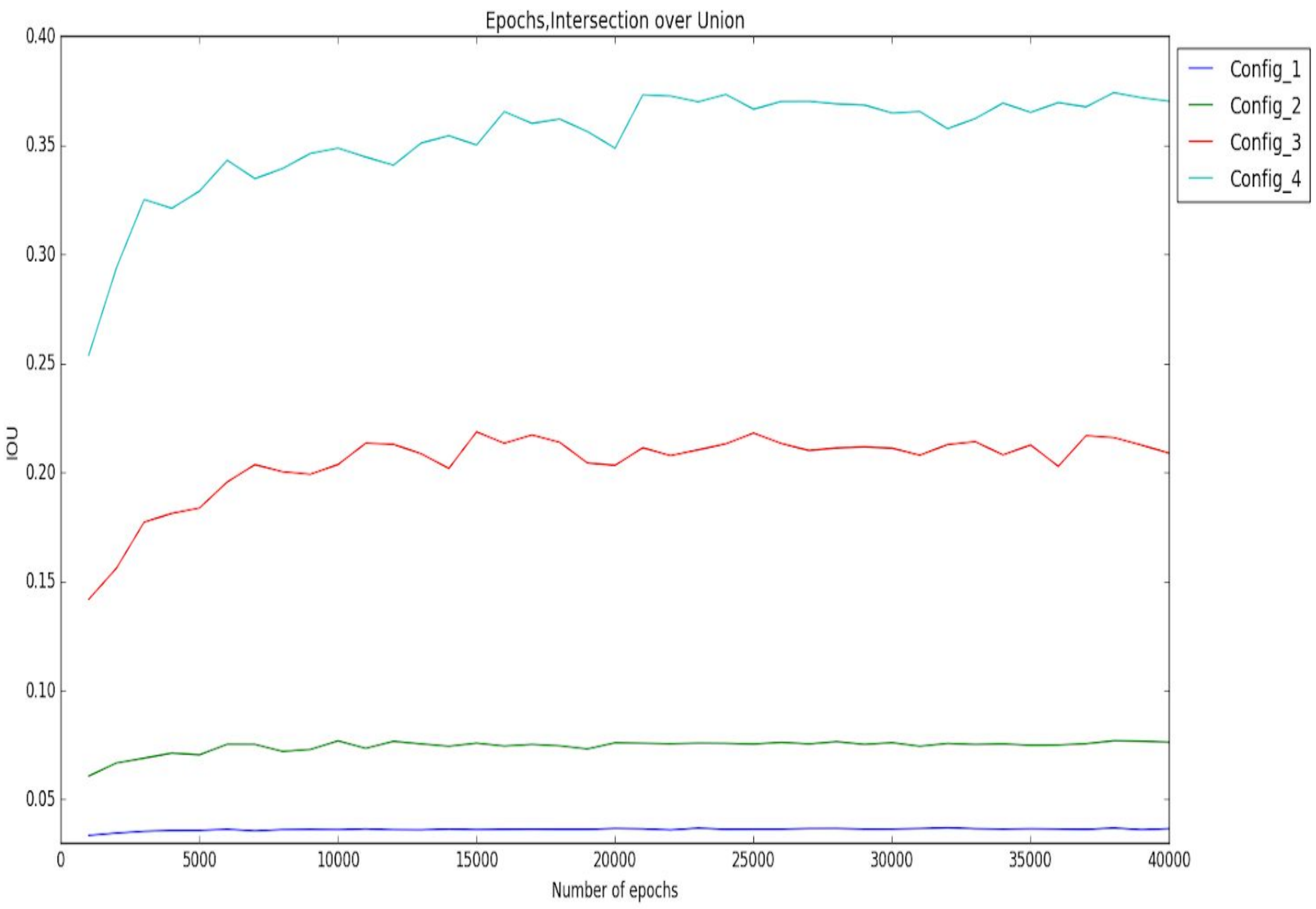
Implementation:

There are four configurations by which model is built and trained and tested, for ach single configuration the structure is different that is explained below.

1. **Config1**: After rigorous amount of downsampling the input from pixel dimension 300*300 was downsampled to 19*19, so by using the function TransitionUp_elu the final downsampled input was upsampled 16 times. It could happen that the upsampled input has an extended dimension for that reason we use crop function. Finally a convolution layer slim.conv2d is used so that at the end the number of feature maps are equal to the number of classes.
2. **Config2**: In the second configuration we could use the skip connection DB4 which help us to improve the refinement so first we upsample using the functionTransitionUp_elu with a stride of 2 and then again with a stride of 8, i.e. sampling it two times. Using the concatenate function with DB4 skip connection so to achieve an upsampling rate of 16 times higher than the original input, it could happen that the upsampled input has an extended dimension for that reason we use crop function.Finally a convolution layer slim.conv2d is used so that at the end the number of feature maps are equal to the number of classes.
3. **Config3**: In the third configuration we used the skip connection DB4 and DB3 which help us to improve the refinement much further than the previous configuration.we upsample using the same function TransitionUp_elu with a stride of 2,2, and 4. convolution layer slim.conv2d is used so that at the end the number of feature maps are equal to the number of classes.
4. **Config4**: In the fourth configuration we used the skip connection DB4,DB3 and DB2 which help us to improve the refinement much further than the previous configuration.we upsample using the same function TransitionUp_elu with a stride of 2,2,2 and 2. convolution layer slim.conv2d is used so that at the end the number of feature maps are equal to the number of classes.

Results:

Configuration_1	Maximum IOU = 0.0370627283283319
Configuration_2	Maximum IOU = 0.07690423656779853
Configuration_3	Maximum IOU = 0.21797843687976395
Configuration_4	Maximum IOU = 0.3741004167412187


Epochs,Intersection over Union
IOU
Number of epochs
Config_1
Config_2
Config_3
Config_4