Faculty of Mathematics

Image Segmentation: U-NET

Isidora Slavković 30/2018

Profesor:
dr Aleksandar Kartelj

6. 7. 2022.

Contents

1 Introduction

Image segmentation is an image processing task in which the image is segmented or partitioned into multiple regions such that the pixels in the same region share common characteristics. There are different kinds of Image Segmentation: Instance and Semantic Segmentation. In this paper we will focus on Semantic Segmentation. Semantic Segmentation assignes a class to each pixel of the image where every pixel of the same class has the same color.

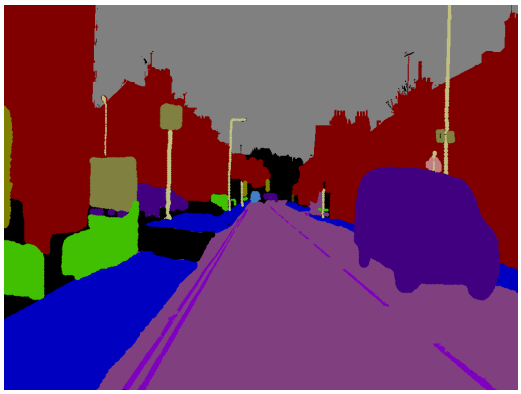

Figure 1: In this picture we can see 32 classes where the car class is represented by purple, the class sky is represented by gray etc...

2 Dataset

Here we will discuss Semantic Segmentation on The Cambridge-driving Labeled Video Database (CamVid) [1] [2] dataset. This dataset provides ground truth labels that associate each pixel with one of 32 semantic classes.

2.1 Current results

Many different neural networks were trained on this dataset:

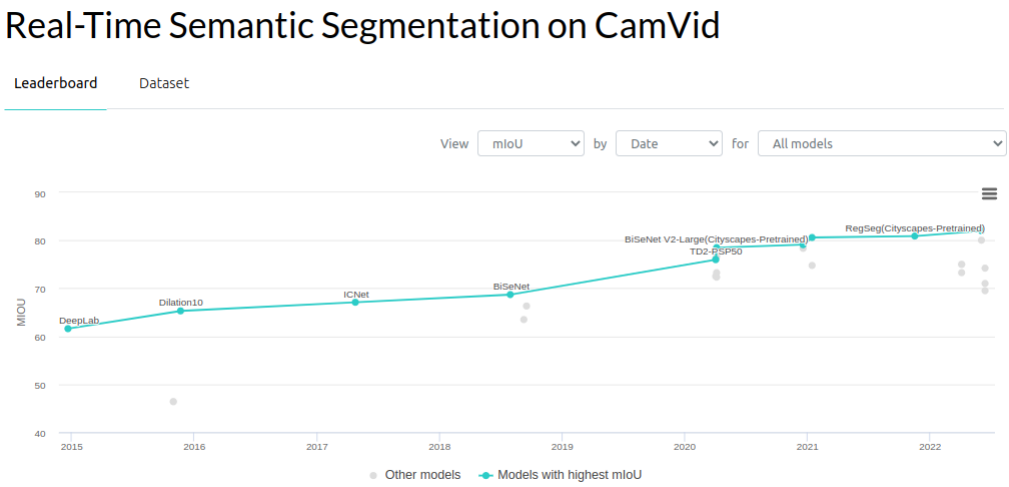

Figure 2: Resaults other neural networks had on this dataset. [3]

2.2 Metrics

The metric that was used in the current results and that will be used here is the mean Intersection-Over-Union metric, better known as mIoU. Intersection-Over-Union is a common evaluation metric for semantic image segmentation, it measures the number of pixels common between the target and prediction masks divided by the total number of pixels present across both masks. For an individual class, the IoU metric is defined as follows:

$$iou = \frac{true\_positives}{true\_positives + false\_positives + false\_negatives}$$

The IoU score is calculated for each class separately and then averaged over all classes to provide a mean IoU score.

3 Neural Network Architectures

For Semantic Segmentation of CamVid dataset we will use a convolutional neural network called U-Net [4]. It has a unique U-shape which was different than the CNNs at the time. It consists

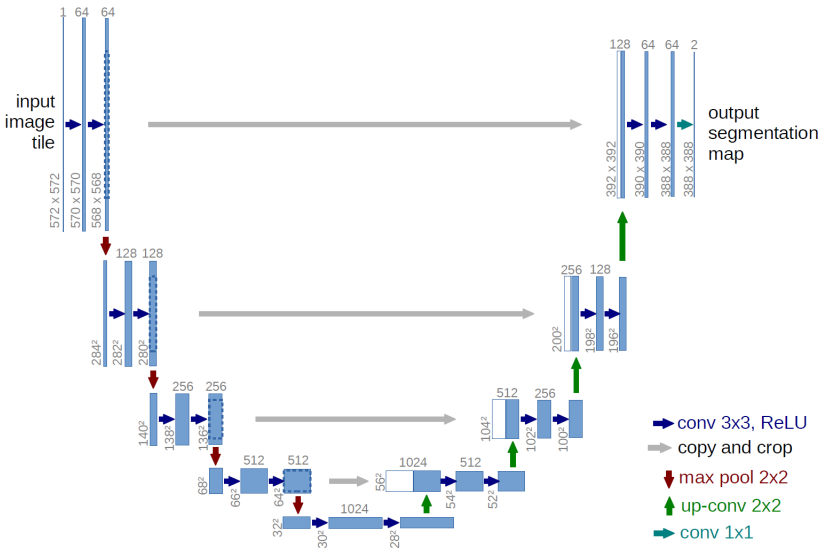

Figure 3: U-Net architecture

of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional neural network. It consists of the repeated application of two 3x3 convolutions, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes. In total the network has 23 convolutional layers.

4 Experimental Results

4.1 Models

Three slightly different U-Net architectures were used:

- model1 - same architecture as in the original paper
- model2 - the only difference with model1 is that the model2 uses BatchNormalization as suggested here
- model3 - the only difference with model1 is that the model3 has additional Dropout layer as suggested here

4.2 Results

Google Colab was used to train the networks. The results achieved can be found here:

num	model	loss	mIoU(%)
1	model1	Bce_dice_loss	51.58
2	model2	Bce_dice_loss	49.94
3	model3	Bce_dice_loss	49.19
4	model1	Mse	48.70

Where Bce_dice_loss represents a summary of Binary Crossentropy and Dice Coefficient Loss:

$$Dice\_Coef\_Loss = 1 - Dice\_Coef$$

where the Dice Coefficient is two times the area of intersection divided by the total number of pixels in both images.

The best performing model is model1 with mIoU of 51.58% on test dataset:

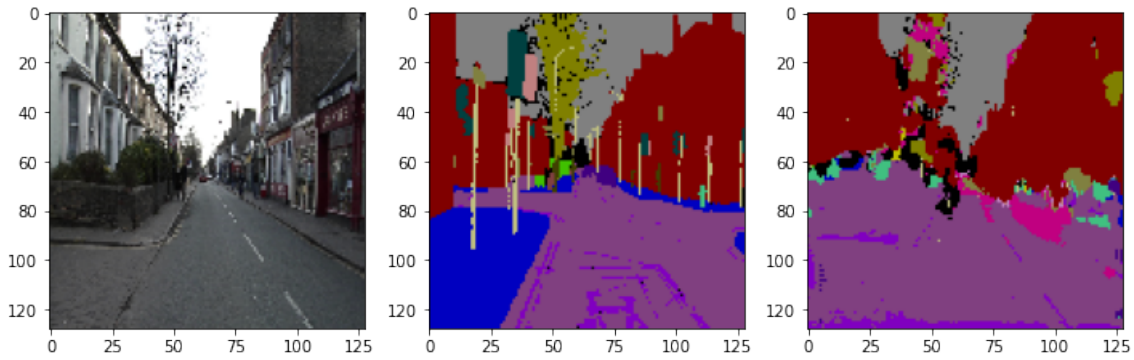

Figure 4: The real image, ground truth mask and model1 prediction.

5 Conclusion

Comparing these results with other CNNs we can see that our models performed significantly worse. The next step would be to make adjustments to the models, change the parameters and use more epochs in order to get better results. If not, then it would be reasonable to consider different network architectures.

References

[1] Segmentation and Recognition Using Structure from Motion Point Clouds, ECCV 2008 Brostow, Shotton, Fauqueur, Cipolla

[2] Semantic Object Classes in Video: A High-Definition Ground Truth Database Pattern Recognition Letters Brostow, Fauqueur, Cipolla

[3] Kaggle, Real-Time Semantic Segmentation on CamVid: link

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation": link