Cityscapes Segmentation

K.H.W. Stolle
Eindhoven University of Technology
k.h.w.stolle@student.tue.nl

TABLE I
TARGET LABELS

Category	Labels
Flat	road, sidewalk, parking, rail track
Human	person, rider
Vehicle	car, truck, bus, on rails, motorcycle, bicycle, trailer
Construction	building, wall, fence, guard rail, bridge, tunnel
Object	pole, pole group, traffic sign, traffic light
Nature	vegetation, terrain
Sky	sky
Void	ground, dynamic, static

***Abstract*—In this paper, a possible solution for the Cityscapes Pixel-Level Segmentation task is discussed. The U-Net architecture is taken as a baseline.**

***Index Terms*—Cityscapes, Segmentation, CNN**

I. Introduction

The Cityscapes Dataset involves a large number of pictures taken from the front of a car while driving through various cities in Germany [**?**].

In this paper, a practical solution to the pixel-level semantic labeling task is discussed. This involves predicting a per-pixel semantic labeling of the image without consiering higher-level object instance or boundary information.

First, a baseline is set using an off-the-shelf architecture for semantic segmentation. This baseline network is then improved by data augmentation and iterative design, after which this solution is tested. The results of testing are finally interpreted and discussed.

II. Method

A. The dataset

As with any machine learning project, first the available data has to be parsed into a practical format. The dataset consists of a collection of PNG-encoded images (the input) and a corresponding segmentation mask (the ground truth). The ground truth is a color-coded image, with RGB-channels, where every class corresponds to a single and unique color. The desired output of the network is an image with one channel per class, where each channel represents the probability of that pixel containing a class. Parsing the ground truth-image into the encoded channel representationn (one-hot encoding) is done via

$$E_n = L_n[3]$$

B. Setting a baseline

The baseline implementation sets a reference point to improve upon and score our method against.

The architecture U-Net is suitable for semantic segmentation of biological cells under a microscope [**?**]. In this paper, the same implementation is used as a baseline for the segmentation of the cityscapes.

Our implementation of the UNet is based on [**?**] with some modifications to work with the Cityscapes Dataset.

C. Data augmentation

A straightforward way to improve the accuracy of the model is to increase the size of the training set by applying a set of transforms. The following transforms were used:

- Random cropping
- Random mirroring over the horizontal axis ($p = 0.5$)

D. Measuring performance

In order to measure how well the network is performing, the Intersection-over-Union (IoU) metric is implemented, given by

$$\mathrm{IoU} = \frac{T\&P}{TP + FP + FN} \tag{1}$$

E. Identifying the model's weaknesses

F. Decision threshold

Sometimes, the case can occur where the softmax-likelyhood of a pixel corresponding to a single class is less than a certain value. Thresholding [ref] adresses this issue by classifying all pixels with a likelyhood less than a set value as zero. Because the Cityscapes dataset has 20 classes, the threshold was set at

$$p_{\mathrm{T}} = 2 \cdot \frac{1}{N_{\mathrm{classes}}} = 0.1 \tag{1}$$

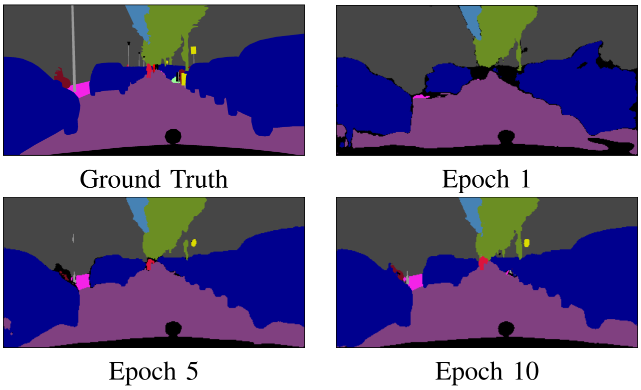
G. Edge detection as input

A lot of semantic segmentation tasks struggle with classes bleeding into other classes [ref]. A possible was to solve this could be to use a static (non-learned) edge detection filter, and feed this into the network by replacing the alpha channel with the single-channel output of this layer.

The hypothesis is that this will help the network detect higher-frequency information such as edges of objects.

H. Edge loss

...



Ground Truth

Epoch 1

Epoch 5

Epoch 10

Fig. 1. Ground truth and predicted masks after training for 1, 5 and 10 epochs of the sample `munster_000098_000019_leftImg8bit.png` using the U-Net model with thresholds and data augmentation

I. Increasing the effective receptive field

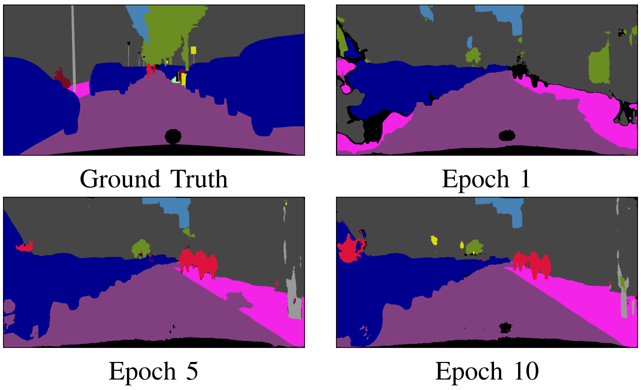
In order to make the network practical, the input images must be scaled to a size that the training hardware can handle. The scaling of images causes a loss of information. A better way to deal with this, would be to downsample the images using strided convolutions. From [red- course sides Convolutional Networks] the output size of a convolutional layer may be calculated by

$$O = \frac{I + P - K}{S} + 1 \tag{1}$$

where I is the input size, P the amount of padding, K the kernel size and S the stride.

J. Automatic learning rate adjustment

The learning rate can be automatically adjusted based on the `ReduceLROnPlateau` method [ref].



Ground Truth

Epoch 1

Epoch 5

Epoch 10

Fig. 2. Ground truth and predicted masks after training for 1, 5 and 10 epochs of the sample `munster_000098_000019_leftImg8bit.png` using the U-Net model with reduced dimensions

III. Results

A. Baseline

B. Data augmentation

C. Threshold

IV. Discussion

V. Conclusion