Comparison of different techniques for image augmentation.

Ralph Lesch and Joshua Heipel

University of Freiburg, Department of Computer Science, Deep Learning Bachelor Project

joshua.heipel@gmx.de

1. Introduction

Convolutional Neural Networks (CNNs) have become a major aproach for the task of Semantic Segmentation and are nowadays used widely over many different fields of applications. Although for some use cases large datasets with thousands of (manually) labeled images have been published (such as CamVid or CityScape in the context of city traffic), in many situations appropriate training data still remains sparse. As a consequence Deep Neural Networks with lots of trainable parameters tend to overfit small and monotonous datasets while generating poor predictions for new (unseen) observations. In order to improve generalization of such CNNs existing training data can be extended by employing different techniques of image augmentation.

2. Architecture

In our case study we use a hierarchical encoder-decoder network with skip connections (CNN of exercise 3) to compare different settings:

(1) **No Augmentation**

(2) **Shape Augmentation** by applying different geometric transformation and dropouts (Horizontal Flip, Scaling, Crop and Padding, rectangular Cutouts)

(3) **Color Augmentation** by varying the intensity values (Adjustment of Brightness and Contrast, Color shifts)

(4) **Shape and Color Augmentation**

3. Training

The network is trained on CamVid dataset, which contains ???? different labeled images. To allow for a fair comparison between the 4 different configurations, we use the same hyper-parameters (namely the total amount of iterations and batch size) for all types of augmentation. Batches are created by sampling images from the original dataset with pseudo-random numbers and then applying augmentations on 90% of the images. To facilitate multiple independent modifications A_k (e.g. cropping and flipping an image) we implemented our own statistical framework based on the principle of inclusion and exclusion, where:

$$P\left(\bigcup_{k=1}^{n} A_k\right) = \sum_{k=1}^{n} (-1)^{(k+1)} \binom{n}{k} P(A_k)^k,$$

which randomly selects different types of augmentations in varying order. We use the same probability for shape or color modification when investigating setting (4). Because of storage limitations all augmentations are calculated *"on the fly"* using Python 3.? together with the Numpy, Scipy, OpenCV and Scikit-Image packages, before feeding them into the CNN.

7. References

4. Augmentations

5. Results

6. Conclusions