Evaluation of different techniques for image augmentation.

Ralph Lesch and Joshua Heipel
University of Freiburg, Department of Computer Science, Deep Learning Bachelor Project
ralph.lesch@neptun.uni-freiburg.de , 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 & Color Augmentation**

3. Training

The network is trained on the CamVid dataset which contains 7707 (manually) labeled images. To allow for a fair comparison between our different configurations (1) - (4), we use the same hyperparameters (namely the total amount of iterations and batch size) for all types of augmentations. Batches are created by sampling images from the original dataset with pseudo-random numbers and then applying augmentations for setting (2) - (4) on 90% of the images. In order to investigate the effect of applying augmentation on sparse data we train the network (I) on the full dataset and (II) with a small subset of 468 images.

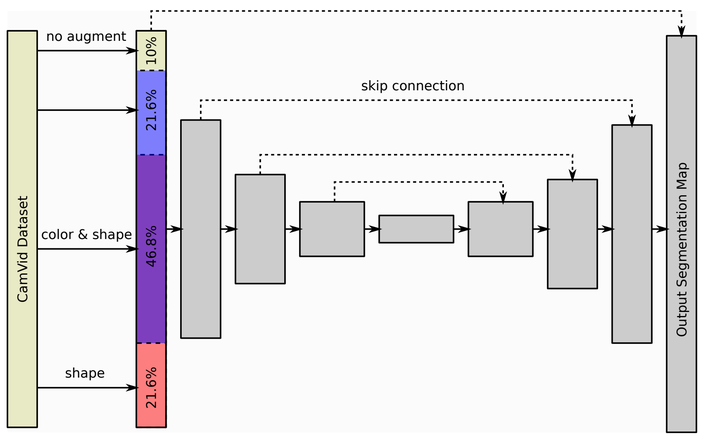

Figure 1: Configuration (4) with shape & color augmentations. 90% of all images are randomly modified and fed into the encoder-decoder network with skip connections.

4. Examples

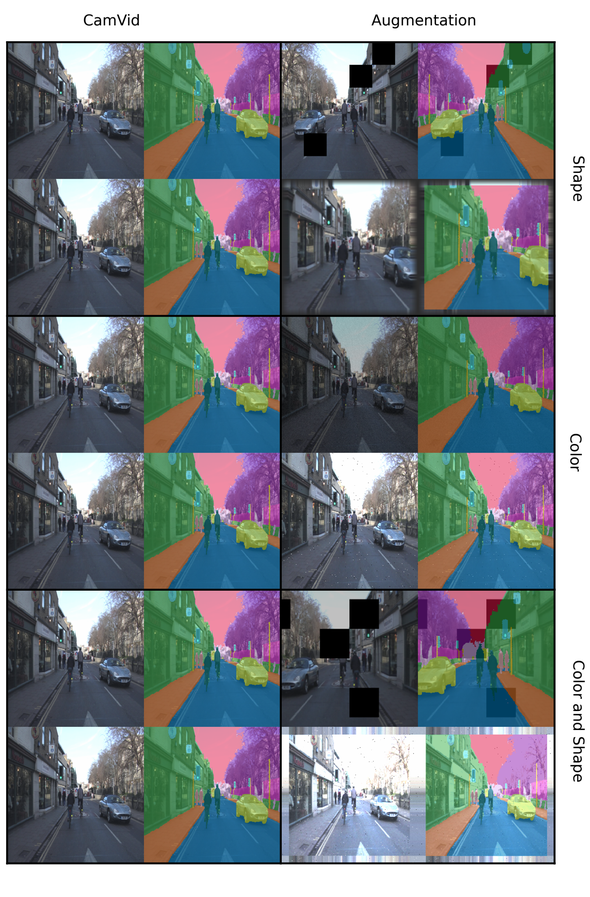

Figure 2: Different types of augmentations. From left: original image, original segmentation map, augmented image and augmented segmentation map.

To facilitate multiple independent modifications A_k (e.g. cropping and flipping an image) we implemented our own probabilistic 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 3.5 together with Numpy, Scipy and the Imgaug package, before feeding the batches into the CNN. In order to parametrize shape and color modifications we draw samples from uniform distribution. Upper and lower bounds are adjusted by manually applying augmentations on different images of the CamVid dataset. The range of possible modifications is thereby limited to realistic variations that correspond to natural changes of lightning conditions and scale transformations. Cutout and padding modifications are chosen to cover at most 1/10 resp. 1/3 of all pixels.

5. Results

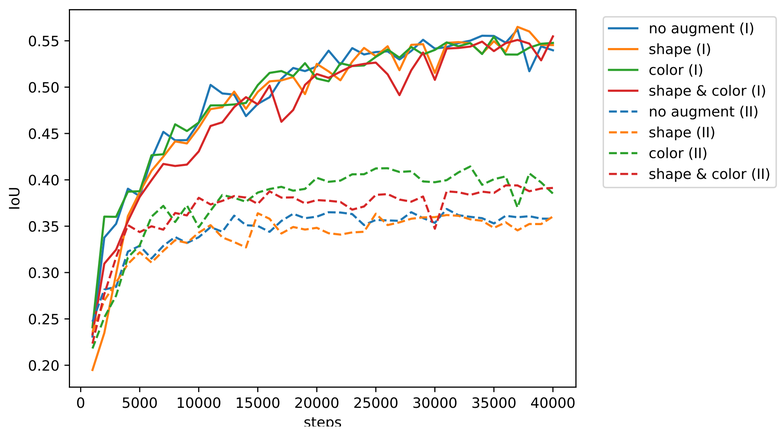

Figure 3: Learning curves with the Intersections over Union (IoU) of both large (I) and small (II) dataset with configurations (1) - (4).

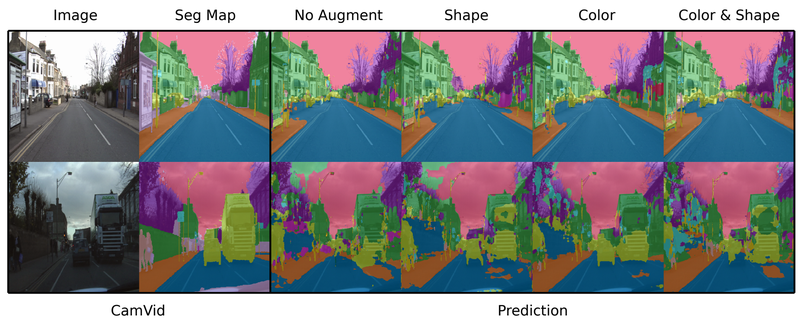

Figure 4: Images and predicted segmentation maps for the test data. From left: original image and segmentation map, predictions without augmentation, with shape, color, shape & color augmentations.

We observe a different performance of the network for configurations (1) - (4) when trained either on the small or on the full dataset. If a large amount of data is available (I), the learning curves show similar trajectories. With sparse training data (II) randomly applied augmentations have a positive effect on the quality of network predictions. Here we see the color augmentation (3) clearly outperform the shape augmentation (2), which gives no clear improvement.
However the gain in performance is not enough to achieve results that are as accurate as with a larger and more variable dataset.

6. Conclusions

- Random modifications applied on sparse datasets can improve performance of CNNs.
- Color augmentation (3) results in a better performance than shape augmentation (2).
- Parameters for different augmentations must be chosen carefully to have the best positive effects and might be optimized with hyperparameter optimization methods.

7. References

- Source: github.com/RalphLesch/dl-lab-2018-project
- Network based on exercise3_CV_ML: github.com/aisrobots/dl-lab-2018