Augmentation Parameter Search

Sara Al-Rawi[1], Juliane Weilbach[1], Ahmed Abdelhadi[1], Nima Riahi Dehkordi[1]

[1]Institute for Computer Vision, University of Freiburg

Introduction

This work represents the final project for the deep learning laboratory. The main topic of this work is searching for optimal augmentation parameters.

- A encoder - decoder fully convolutional neural network with four convolution layers has been used for this work. The network consists of three refinement blocks. These refinement blocks add skip connections from the encoder network to the decoder network.
- For upsampling the features in the decoder network a transposed convolution is used.
- The network has been trained on the CamVid dataset, which is a collection of videos with perspective of a driving automobile. Each pixel is associated with one of 31 semantic classes.
- For finding the optimal hyperparameter the exploitation approach was used. It consists of a limited searching space and is therefore related to local search.
  The advantage is a reduced training time compared to the exploration approach.

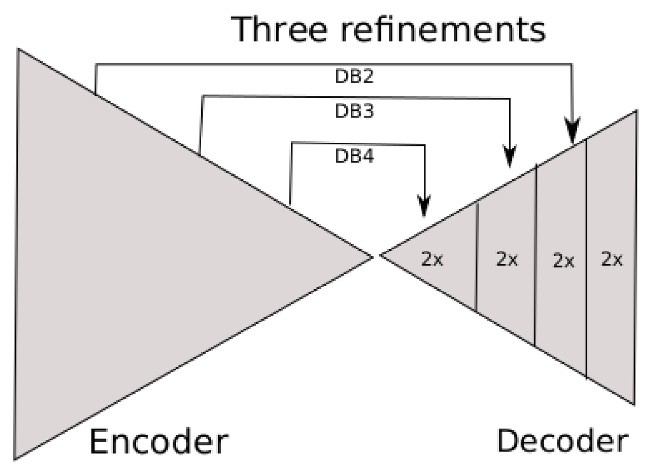


Figure 1: Encoder-Decoder network with skip connections

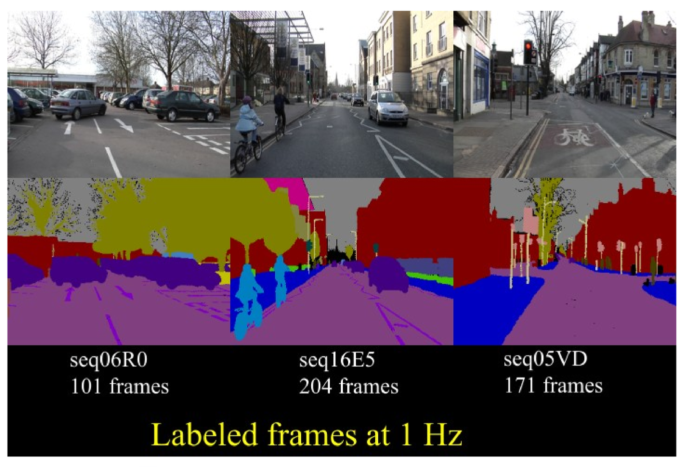


Figure 2: CamVid Frames [1]

In Figure 1 the used encoder-decoder neural network including the refinement modules with upsampling rate two and the corresponding skip connections is shown.
Besides in Figure 2 an example for the CamVid images with it's labeled frames is shown.

Baseline

The baseline that has been considered is the above mentioned network trained on CamVid without considering any augmentation.

The performance metric was the IoU (Intersection of Union) which measures the intersection over the union of labelled segments for each class and reports the average.
The IoU is shown in Figure 3 reached 0.55 after training the network 80 epochs.

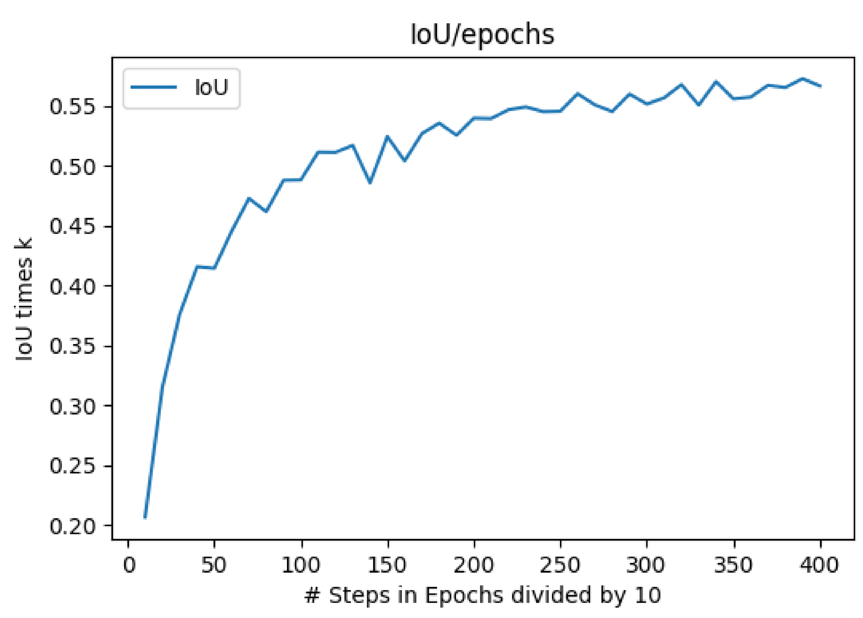


Figure 3: IoU Baseline without augmentation

This result can be considered as acceptable baseline in terms of training time and dataset difficulty.

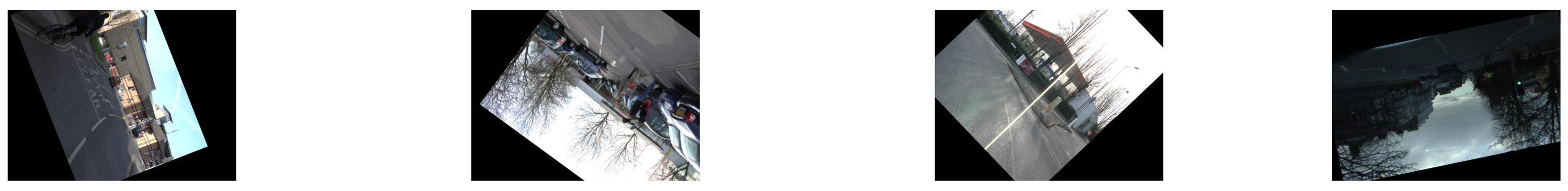
Augmentation

Data Augmentation artificially creates novel training images through different ways of image processing, such as Random Crop, Flipping or Scaling etc.
Two types of Augmentation techniques were used, Spatial and Intensity Augmentation. The following value intervals in the tables have been used for optimization.

Spatial Augmentation	
Random Crop	0 - 16 pixel
Horizontal Flip	50% of images
Vertical Flip	20% of images
Image Scaling (x and y-axis)	(0.7,1.4)[4]
Translation (x and y)	(-0.2, 0.2)
Rotation	-90,90

Intensity Augmentation	
Gaussian Blur	σ: 0-3.0
Additive Gaussian Noise	0-0.02
Salt and Pepper	0-0.5%

The following images show spatial augmentation examples for the CamVid dataset. This photos are produced with Imgaug [2]. Like in the neural network the augmentation methods shown below are mixed up.



Likewise the below images with Intensity Augmentation are produced using the same API.

Results

The learning rate has been optimized on the baseline, best results achieved with 0.0002 for Adam Optimizer. In Figure 4 the IoU for Spatial augmentation is shown, which reaches a higher IoU as Intensity Augmentation in Figure 5. One reason could be that the images with Insensity Augmentation are heavily augmented compared to the spatial ones. Another reason could be the exploitation approach in hyperparameter search, which has the risk to get stuck in a local minimum.

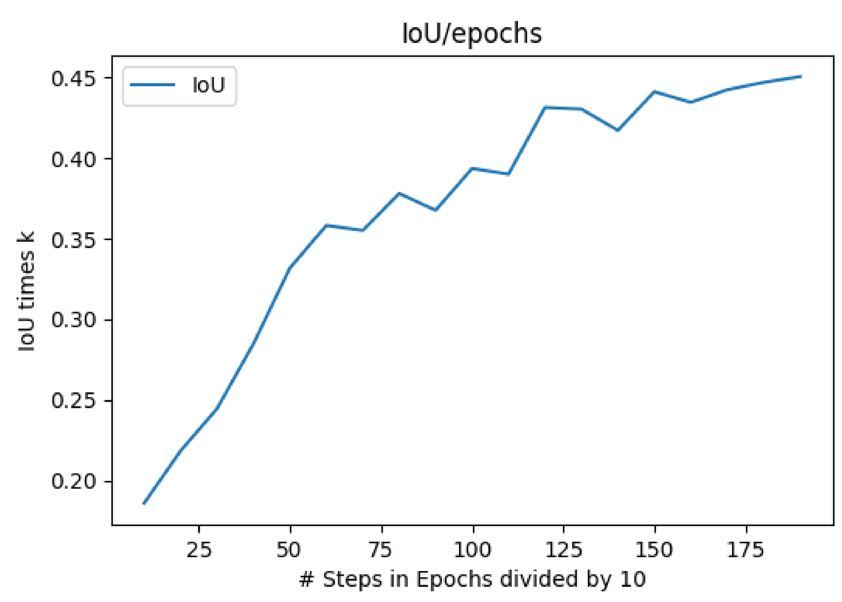


Figure 4: Best results for Spatial Augmentation

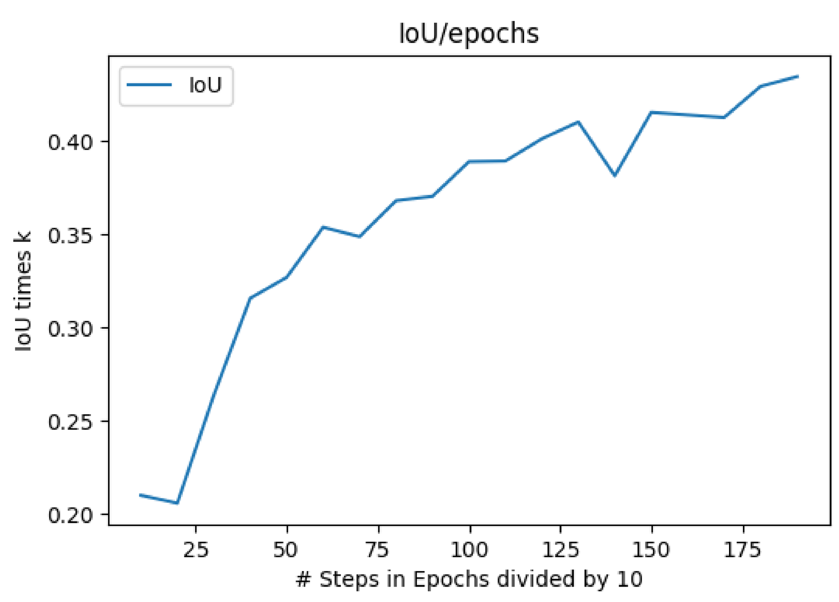


Figure 5: Best results for Intensity Augmentation

Spatial Augmentation	
Random Crop	27% of image
Flipping	25% of images
Scaling	74% (upscaling)
Rotation	19.97°

Intensity Augmentation	
Gaussian Blur	σ: 2.86
Additive Gaussian Noise	0.015
Salt and Pepper	0.48%

In future to get better results it would be possible to optimize the hyperparameter of the augmentation methods with Bayesian Optimization, Hyperband [3] or to use more epochs for training.

References

[1] http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/. Accessed: 2019-01-30.

[2] https://github.com/aleju/imgaug-doc/blob/master/source/installation.rst. Accessed: 2019-02-01.

[3] https://github.com/zygmuntz/hyperband. Accessed: 2019-02-01.

[4] T. B. Gabriel L. Oliveira, Wolfram Burgard. Efficient deep models for monocular road segmentation. 2016.