Project Report on

Coarse-to-Fine Annotation

3D Computer Vision

Heidelberg Collaboratory for Image Processing (HCI)

By

Achita Prasertwaree (3585472)

Niels Bracher (3366659)

Thomas Ehret (3332970)

Ruprecht-Karls-Universität Heidelberg

Summer Semester 2019

Project Committee:

Prof. Dr. Carsten Rother, Lecturer

Radek Mackowiak, Project Supervisor

Abstract

High annotation quality of ground truth segmentation maps is of crucial importance for training neural networks to accurately segment multi-class scenes. While the acquisition of coarse segmentation ground truth via crowd-sourcing is comparatively cheap, collecting a large set of fine-labeled segmentation maps as required for example in the field of autonomous driving can quickly become prohibitively expensive. Using FASSEG [4] as a toy dataset we investigated the possibility of neural networks to learn a style transfer from coarse-labeling to fine-labeling. Taking pixel-wise accuracy of the coarse segmentation maps relative to the fine segmentation maps as a baseline for comparison, we found that a U-Net trained on the original RGB pictures concatenated with the coarse segmentation map as an “initial guess” generates 5 % more accurate segmentation maps. In a second semi-supervised model we then explored the idea to keep training a style transfer neural network on coarse labels even when running out of fine ground truth. To that end, we held out 60% of the available FASSEG fine ground truth to train a discriminator, replacing the missing fine ground truth in a second training stage. It turned out that the discriminator was not up to the task of capturing the difference between coarse and fine segmentation maps.

Table of Contents

Abstract **1**

1.Introduction **3**

1.1 Coarse-to-Fine Annotation 3

1.2 Proposed Model 3

2.1 Dataset - FASSEG 4

2.2 Model 5

2.2.1 Generative Adversarial Networks (GANs) 5

2.2.2 Conditional GANs (cGANs) 6

2.2.3 Deep Convolutional GANs (DCGANs) 7

2.2.4 U-Net 7

2.3 Loss 8

2.3.1 Weighted pixel-wise cross-entropy 8

2.3.2 Mean Squared Error (MSE) 9

2.4 Optimizer 9

2.4.1 Adaptive Moment Estimation (Adam) 9

2.5 Metric 10

2.5.1 Pixel Accuracy 10

3. Experiment **10**

3.1 Preprocessing 10

3.2 Training 11

3.3 Testing 15

4. Conclusion **16**

5. Future work **16**

6. References **17**

Coarse-to-Fine Annotation

1.Introduction

1.1 Proposed Model

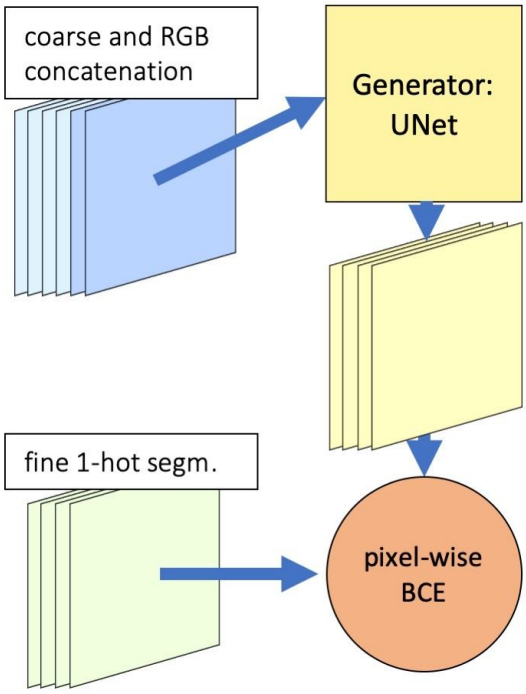

Figure 1: Training U-Net without the discriminator for style transfer.

We propose two models for coarse-to-fine transfer. As depicted in **figure 1** the first model is based upon the convolutional neural network U-Net [6], which receives as input a concatenation of the original RGB image and six one-hot maps computed from the corresponding coarse segmentation map. The idea behind concatenating these one-hot maps is to provide U-Net with a good “initial guess” for the fine segmentation map it should learn to infer.

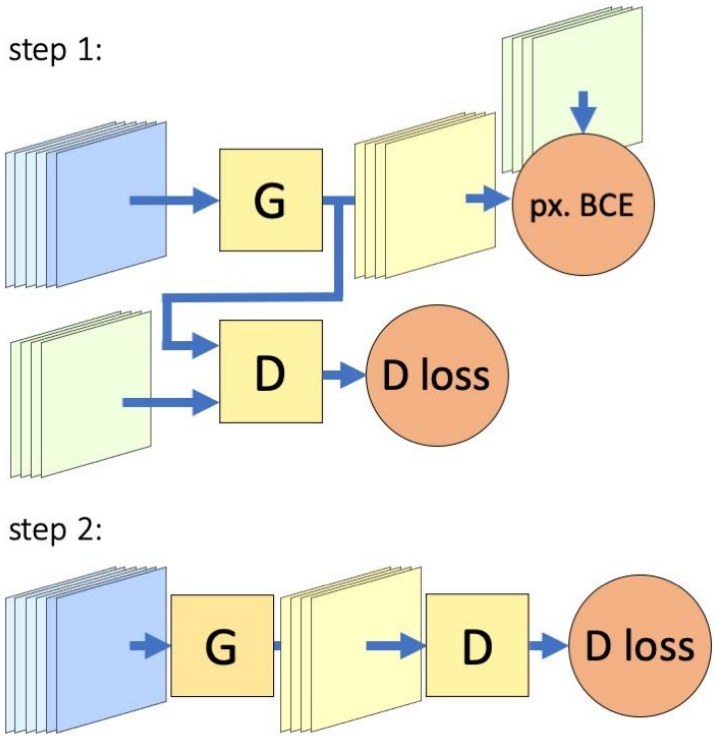

Figure 2: cGAN-inspired style transfer architecture.

The first step of our second model, as shown in **figure 2,** is inspired by the cGAN architecture so far that we used the original RGB image as the condition and we trained two neural networks against each other. Imitating the oftentimes encountered scenario in which much fewer images are finely annotated than coarsely annotated, we intentionally held out 60 % of our training ground truth. The idea of **step 1** is to train both generator and discriminator in a competitive minimax game until we run out of training images with fine ground truth segmentation maps. In **step 2** of the training procedure, we keep training on the held-out training images without ground truth, relying solely on a loss computed from the discriminator trained in **step 1**.

2. Background

2.1 Dataset - FASSEG

The FASSEG (FAce Semantic SEGmentation) dataset is composed of multi-class semantic segmentations divided into three subsets namely “frontal01”, “frontal02” and “frontal03”. For our purpose, we used the subsets “frontal01” and “frontal02” containing 70 RGB images of faces, each with a coarse and a fine segmentation map. The faces got labeled with six classes, namely "background", "hair", "skin", "eyes", "nose" and "mouth". The images were all of the height 512 pixels while the width varied.

2.2 Model

2.2.1 Generative Adversarial Networks (GANs)

GAN [3], as shown in **figure 3**, is a generative model introduced by Ian Goodfellow in 2014. It consists of two networks, a generator and a discriminator. The purpose of the generator is to generate an image attempting to fool the discriminator. Conversely, the goal of the discriminator is trying to tell the genuity of each photo. Both are trained simultaneously and compete with each other in the minimax game, as shown in **figure 4-5.**

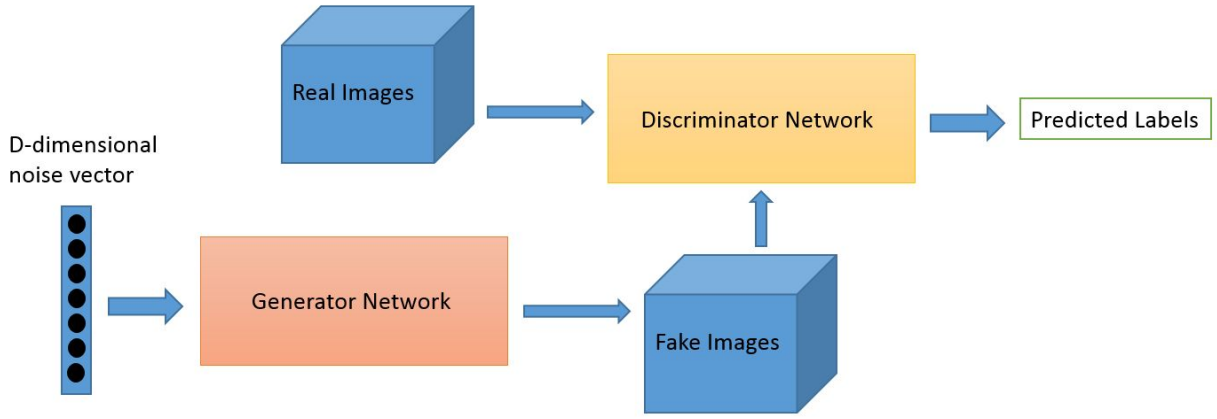

Figure 3. GANs architecture[1].

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Figure 4: The minimax equation [3].

$$\max_D V(D) = \underbrace{\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]}_{\text{recognize real images better}} + \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}_{\text{recognize generated images better}}$$

$$\min_G V(G) = \underbrace{\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}_{\text{Optimize G that can fool the discriminator the most.}}$$

Figure 5: Descriptive meaning for the minimax equation[2].

[1] https://www.oreilly.com/learning/generative-adversarial-networks-for-beginners, 12.10.19

[2] https://medium.com/deep-math-machine-learning-ai/ch-14-general-adversarial-networks-gans-with-math-1318faf46b43, 12.10.19

Here are some steps of training GANs

1. The generated picture from the generator is fed into the discriminator, which distinguishes both real images and fake images and outputs the probability. In the training of the discriminator, the generator parameters should remain constant.
2. Then, we train the generator with the output from the discriminator. During the training, we turn off the discriminator.

Finally, the two neural networks should have a similar "skill-level" after the zero-sum game, not overwhelming each other.

However, in the training progress, GANs might suffer from problems such as non-convergence, mode collapse, vanishing gradient etc. These problems triggered various developments for GANs such as cGANs, conditioning the representation of the images and DCGANs, improving the infrastructure of vanilla GANs.

2.2.2 Conditional GANs (cGANs)

cGANs [5] come to guarantee the particularity of the output from the generator and also the discriminator, by employing a one-hot vector as the condition (c.f. **figure 6**).

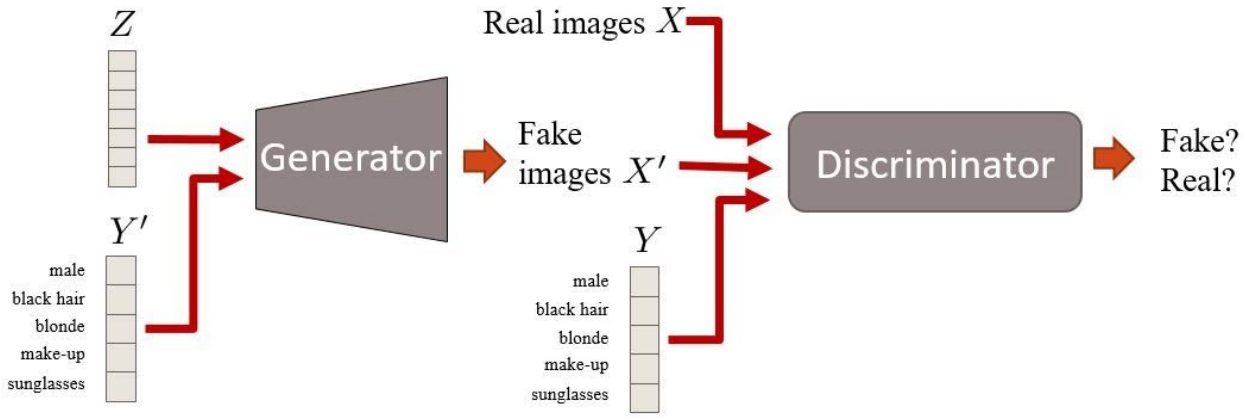

Figure 6: cGANs architecture[3].

2.2.3 Deep Convolutional GANs (DCGANs)

Firstly, CNN is a powerful model, commonly used in supervised learning. One of the outstanding features of CNNs is their translation invariance property that becomes possible through their convolutional layers. This is important when doing semantic segmentation. DCGANs [1] use these advantages from CNNs, building up their generator and discriminator.

2.2.4 U-Net

U-Net [6], as shown in **figure 7**, is the deep neural network, originally used for biomedical image segmentation. The model consists of a contracting path and an expansive path. The

[3] http://guimperarnau.com/blog/2017/03/Fantastic-GANs-and-where-to-find-them, 12.10.19

usage of the skip-connection, or a concatenation, is to recover the information from downsampling while upsampling the data. Moreover, we could learn the upsampling by transposed convolution, originated from the traditional convolution.

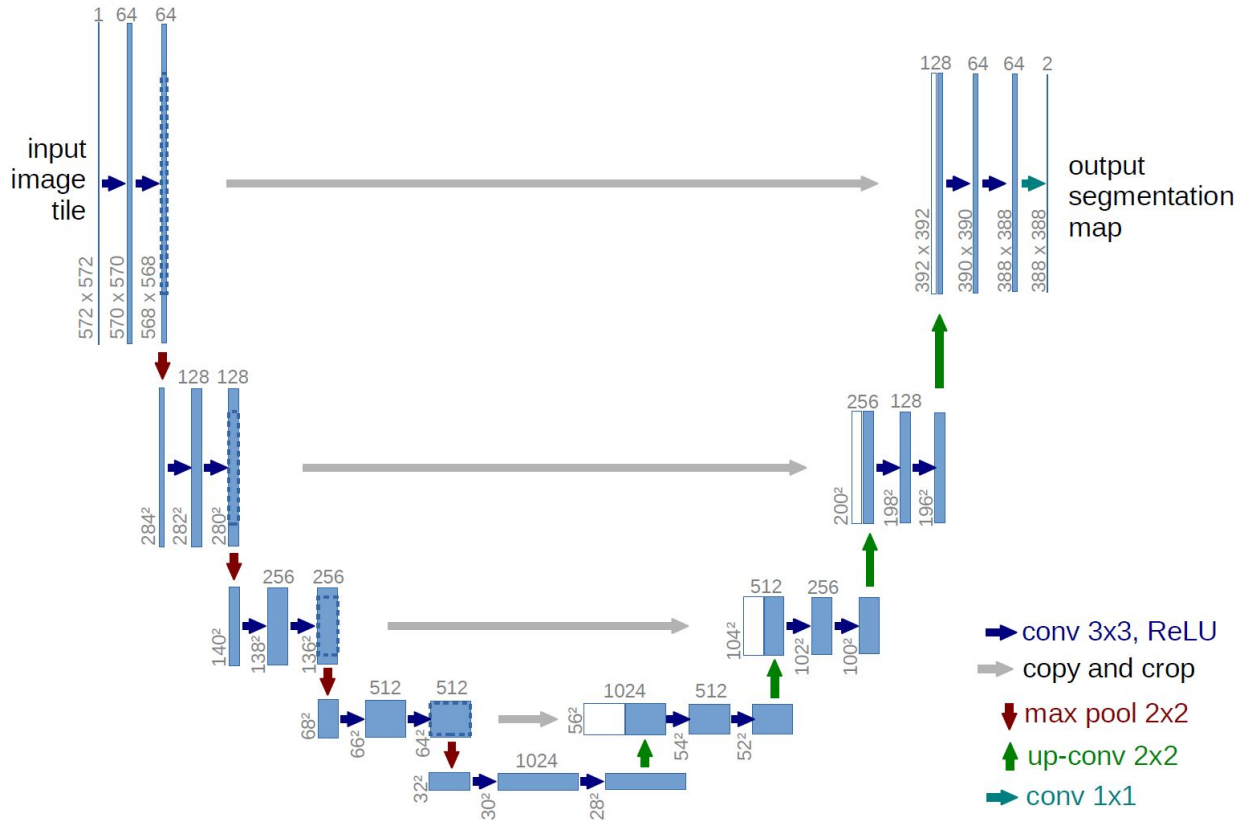

Figure 7: The U-Net Architecture. The number of channels is labeled above each box [6].

2.3 Loss

2.3.1 Weighted pixel-wise cross-entropy

The loss is computed pixel-wise from the softmax values of the prediction and the one-hot encoded target with the equation illustrated in **figure 8**. Weighting the loss for each output channel is done to counteract the unbalanced representation of classes in our dataset and is also used in the original U-Net model.

$$px.\,BCE = -\sum_{pixels}\sum_{classes} weight_{class} * y_{true}\log(softmax(y_{pred}))/(Image\ Area * Batchsize)$$

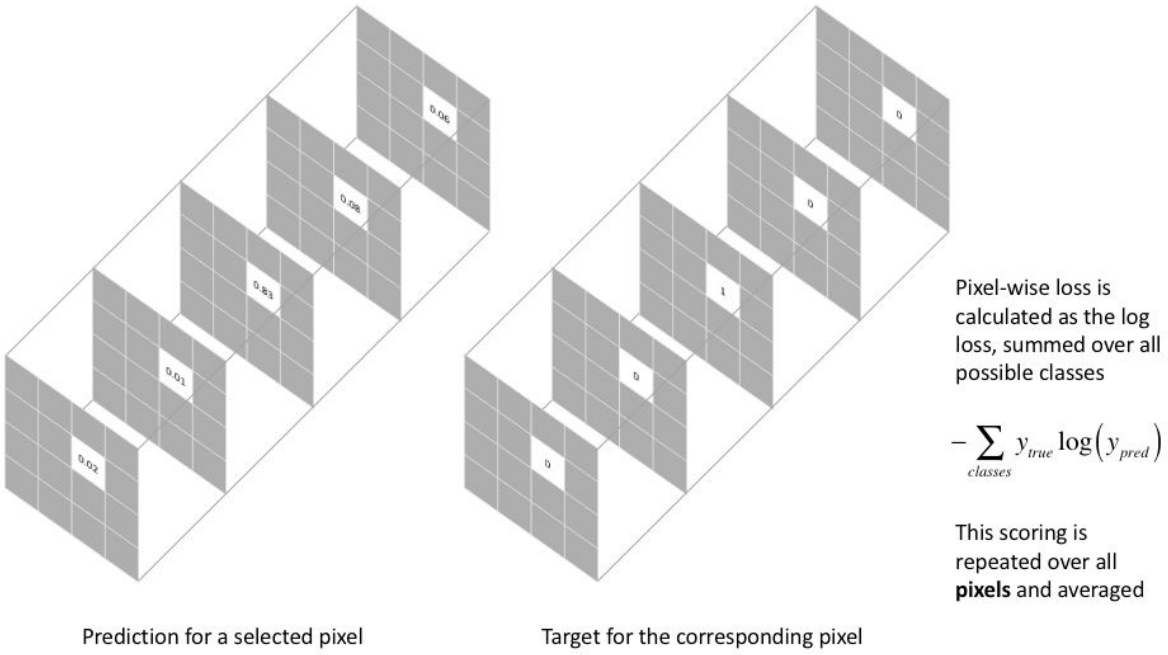

Figure 8: Pixel-wise cross entropy and visualization[4].

2.3.2 Mean Squared Error (MSE)

MSE is a common loss used in Machine Learning. The loss is computed from the difference between the prediction and ground truth. Then, the value is squared and averaged, respectively. The loss ensures that the trained model will not produce a huge error of outlier predictions. The equation is written in **figure 9.**

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Figure 9: MSE loss.

2.4 Optimizer

2.4.1 Adaptive Moment Estimation (Adam)

Adam is a combination of two optimizers, which are competent in dealing with sparse gradients and non-stationary. It is a method that computes an adaptive learning rate. The step size is approximately bounded by the step-size hyperparameter, which can be considered as a

[4] https://www.jeremyjordan.me/semantic-segmentation/, 12.10.19

forgetting factor for the respective momentum. Adam is scale-invariant, efficient in computation and requires little memory.

2.5 Metric

2.5.1 Pixel Accuracy

The accuracy is a metric used in semantic segmentation, to report the correctness of each pixel, which can be represented as:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 10: Pixel accuracy.

where TP, TN, FP, FN stands for true positive, true negative, false positive and false negative, respectively.

3. Experiment

3.1 Preprocessing

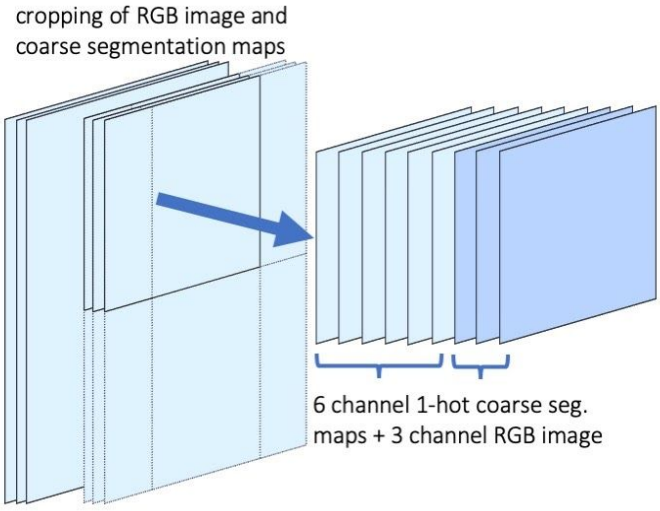

Figure 11: The preprocessing.

To ensure that all the images we feed to the conditional U-Net are of the same shape, some steps of preprocessing were necessary. First, we decided that the input should have the shape 256x256x9. Therefore, four corner crops were applied to the RGB images as well as to the segmentation maps imitating a sliding window. An overlap happens to occur in the horizontal direction since the width of the images is smaller than 512 px. Afterward, the segmentation maps got one-hot coded and concatenated with the normalized RGB image.

Because our dataset is unbalanced in terms of the proportion of classes in one image, we decided to correct for that using a weighted pixel-wise batch cross-entropy loss, where the weights are the inverse proportions of the individual classes averaged over all segmentation maps.

3.2 Training

We used Google Colab as our working platform with PyTorch as our development framework, supplying us with a variety of functions typically used in semantic segmentation. We used 50/70 images of the FASSEG dataset for training the first model and tested U-Net's capability of style transfer on the remaining 20 images. In the training process, we used the Adam optimizer with a learning rate of 2e-4. Training U-Net in our first model **without** a discriminator for 1000 epochs gave rise to **figure 12**.

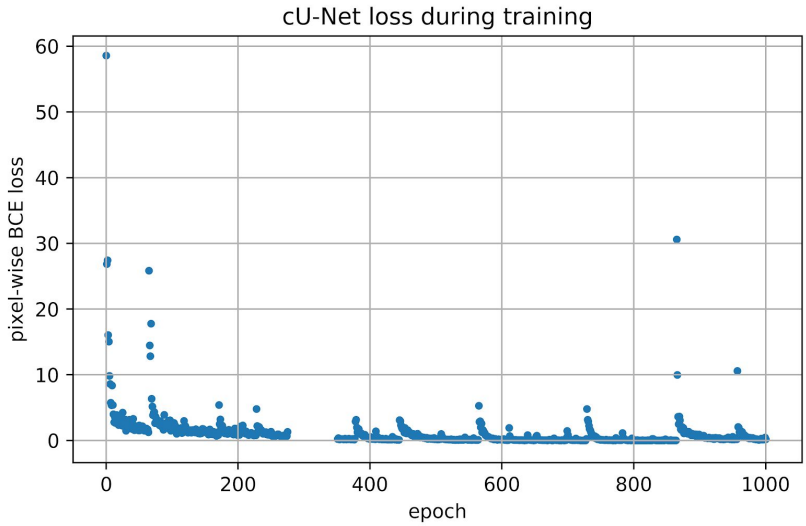

Figure 12: Pixel-wise BCE loss of the conditional U-Net.

Two things immediately strike the eye. First, the pixel-wise BCE loss decays very quickly over the first 60 epochs and continued training up to epoch 1000 does not lead to a sizable further decrease in loss. Second, the loss jumps to disproportionately high values several times during training. A possible explanation for these two observations is that we might have chosen a too large learning rate preventing optimization from exploring local minima more thoroughly.

The second model **with** a discriminator was trained for 500 epochs in step 1 and for 500 epochs in step 2. To simulate an unbalanced dataset in terms of uneven coarse and fine segmentation maps (c.f. dataset Cityscapes), 20 images with coarse and fine segmentation were used to train the generator and especially the discriminator in step 1. The sum of the pixel-wise batch cross-entropy loss and the discriminator loss as a result of training the generator as well as the

MSE loss as a result of training the discriminator are displayed in **figure 13.** Both the loss jumps and the quick decrease as described for model 1 are also present for model 2.

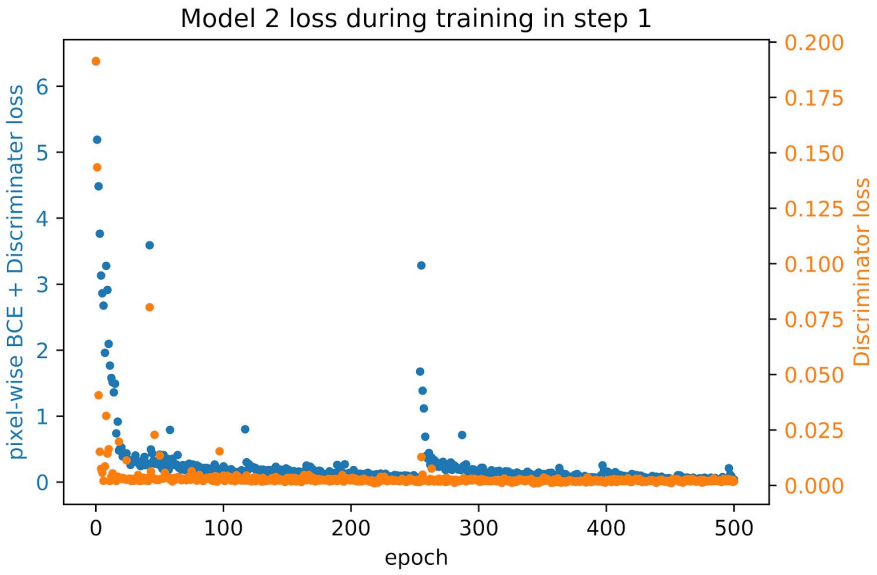

Figure 13: The loss of discriminator and generator of model2 during step 1.

Afterward in step 2 the fine segmentation maps were neglected and the generator was trained with the remaining 30 images and coarse segmentation maps of our training set. Using the MSE to compute the loss of the discriminator resulted in **figure 14**.

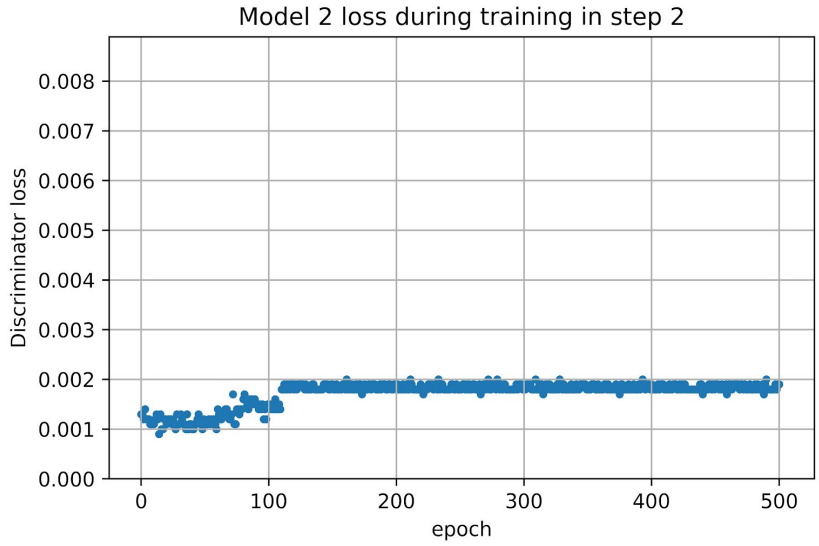

Figure 14: The loss of generator of model 2 during step 2.

While in the optimal case the loss would exponentially decrease, here it increases until it converges around 0.002. It seems in this case that the discriminator has not learned enough to replace the pixel-wise loss and train the generator without having ground truth fine segmentation maps.
To compare our results of using the conditional U-Net, also an ordinary U-Net was used to segment the RGB images from the training set. The loss converges much slower compared to model 1 and model 2, while the loss peaks during training are still remaining as described for model 1.

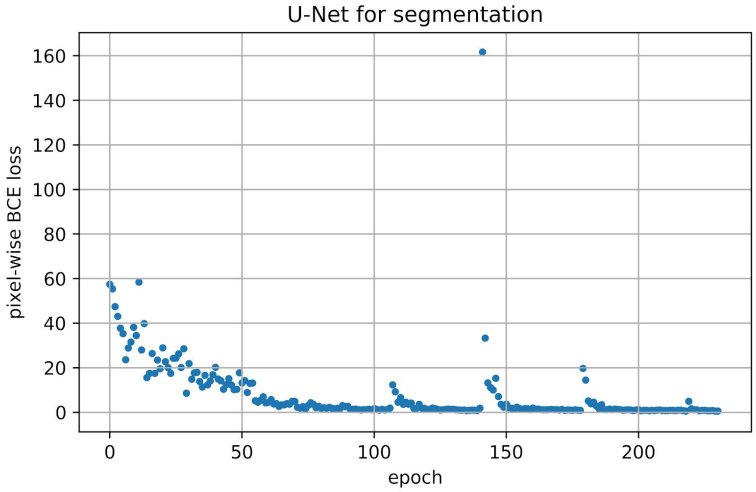

Figure 15: The loss of U-Net for segmentation.

Results from the trained models generating sampled images from the test dataset are shown below. The crosses in **figure 19** are produced when stacking the cropped images back together. Nevertheless, the quality of the generated images reflects the reached accuracy (**table 1**).

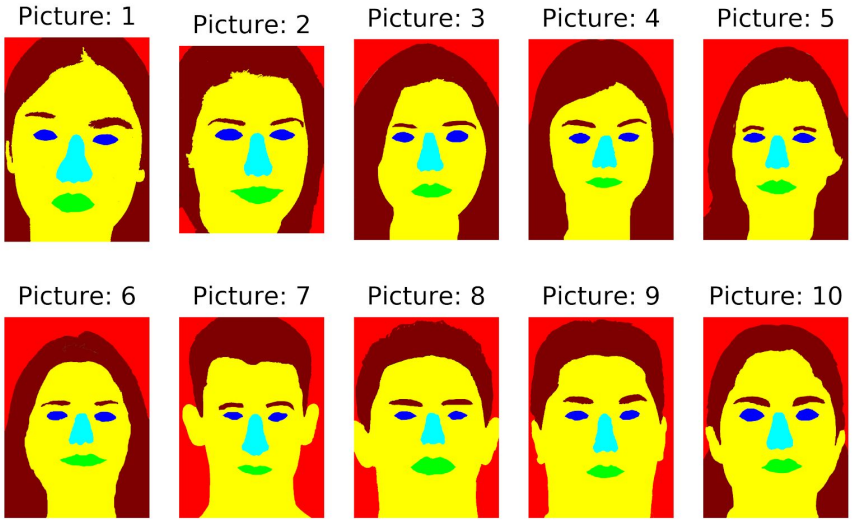

Figure 16: Ground truth fine segmentation maps from test dataset.

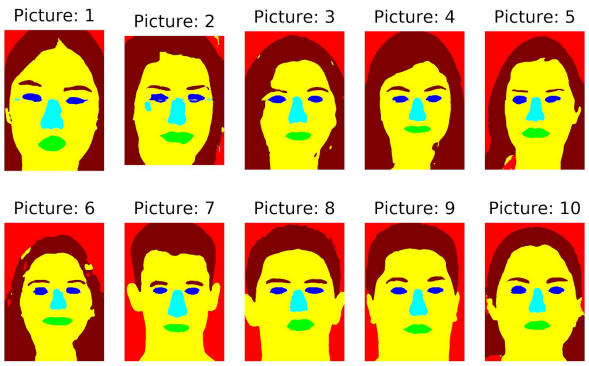

Figure 17: Generated images with conditional U-Net (model 1).

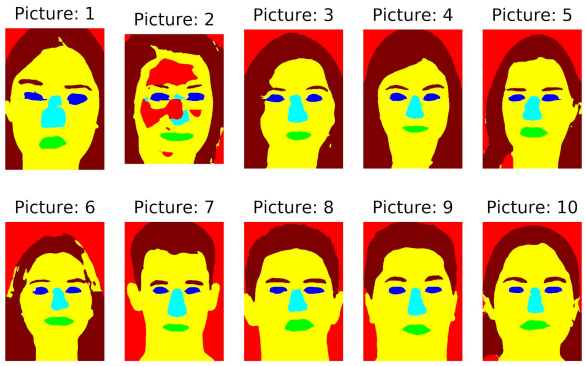

Figure 18: Generated images with Model 2 (step 1).

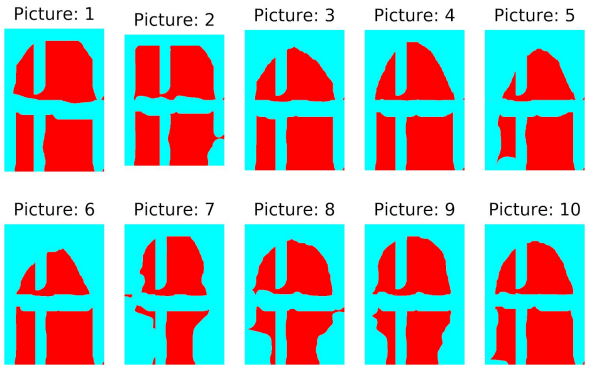

Figure 19: Generated images with Model 2 (step 2).

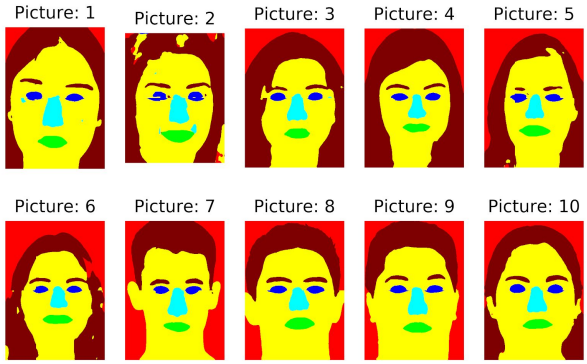

Figure 20: Generated images with U-Net for segmentation.

3.3 Testing

We chose pixel-wise accuracy as our metric over intersection over union (IoU). While IoU as an accuracy measure puts emphasis on getting the largest possible overlap of the respective classes, pixel-wise accuracy focuses more on getting right where classes are **not** present. For the task of style transfer from coarse to fine segmentation, this is what really matters, since the main difference between coarse and fine maps lies in the accuracy of the class boundaries. A reference accuracy was defined as the mean pixel-wise accuracy of the coarse segmentation maps on the fine segmentation maps averaged over the whole dataset.
It turned out that model 1 was the most accurate with a nearly 5 % better accuracy compared to the reference (c.f. **table 1**).

Model	Accuracy
Reference	89.9 %
Model 1	94.5 %
Model 2 (step 1)	91.4 %
Model 2 (step 2)	2.3 %
Just U-Net (segmentation)	92.8 %

Table 1: Pixel-wise accuracy of the two models.

The model 2 works efficiently in the step 1, with an accuracy improvement of 2% compared to the reference. However, when the training comes to stage 2, the model fails to learn and then completely breaks down.

4. Conclusion

In our first model we investigated the capability of a U-Net to learn a style transfer from coarse-to-fine labeling of segmentation maps taken from the toy dataset FASSEG. We found that the conditional U-Net reached a 5% better accuracy than our reference. This supports our first suggestion that providing an initial guess for the fine segmentation map in the form of a coarse label not only helps to speed up convergence during training but also allows to reach a better accuracy in a prescribed number of training epochs.
In a second semi-supervised model we then explored the idea to keep training a style transfer neural network on coarse labels when running out of fine ground truth. Our second approach to style transfer did not work out as we had hoped for. During the minimax game of step one we could not train the discriminator well enough to replace a ground truth-based loss and guide the generator in the second step.

5. Future work

As suggested in the original U-Net paper it would be worthwhile to investigate the effectiveness of data augmentation on FASSEG. Furthermore, preprocessing is important and we would like to try a proper weight initialization. Given that the FASSEG dataset is comparatively small, k-fold cross validation could be a promising option for hyperparameter selection. Unfortunately, the second step of the style transfer model does not achieve the desired accuracy. So, we plan to modify the discriminator, which we consider to be the main problem of the result, in several ways. Instead of just classifying a fake or real image, this could be done pixel-wise with a kind of fake/valid segmentation map to transfer more specific information to the discriminator during

backpropagation [2]. The next interesting thing to try would be to do a coarse-to-fine segmentation on the Cityscapes dataset. In fact, our second model was inspired by the difference in the number of finely-annotated (3000 maps) and coarsely-annotated (20000 maps) images found in Cityscapes. Clearly, accomplishing a successful style transfer from coarse-to-fine segmentation maps would improve the ground truth quality in a cost-effective way and thus enable an improved training of multi-class segmentation networks, e.g for autonomous driving.

6. References

[1] Alec Radford, Luke Metz, Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. 2016.
[2] Hamideh Kerdegari, Manzoor Razaak, Vasileios Argyriou, Paolo Remagnino. Semi-supervised GAN for Classification of Multispectral Imagery Acquired by UAVs. 2019.
[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. Generative Adversarial Networks. 2014.
[4] *Khalil Khan*, *Massimo Mauro*, *Riccardo Leonardi*, “Multi-class semantic segmentation of faces”, *IEEE International Conference on Image Processing (ICIP)*, 2015.
[5] Mehdi Mirza, Simon Osindero. Conditional Generative Adversarial Nets. 2014.
[6] Olaf Ronneberger, Philipp Fischer, Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. 2015.
[7] Pauline Luc, Camille Couprie, Soumith Chintala, Jakob Verbeek. Semantic Segmentation using Adversarial Networks. 2016.
[8] Yadan Luo, Ziwei Wang, Zi Huang, Yang Yang, Cong Zhao. Coarse-to-Fine Annotation Enrichment for Semantic Segmentation Learning. 2018.