 DOI: 10.3217/978-3-85125-603-1-07

Generative Adversarial Networks to Synthetically Augment Data for Deep Learning based Image Segmentation*

Thomas Neff[1], Christian Payer[1], Darko Štern[2,1], Martin Urschler[2,1]

Abstract— **In recent years, deep learning based methods achieved state-of-the-art performance in many computer vision tasks. However, these methods are typically supervised, and require large amounts of annotated data to train. Acquisition of annotated data can be a costly endeavor, especially for methods requiring pixel-wise annotations such as image segmentation. To circumvent these costs and train on smaller datasets, data augmentation is commonly used to synthetically generate additional training data. A major downside of standard data augmentation methods is that they require knowledge of the underlying task in order to perform well, and introduce additional hyperparameters into the deep learning setup. With the goal to alleviate these issues, we evaluate a data augmentation strategy utilizing Generative Adversarial Networks (GANs). While GANs have shown potential for image synthesis when trained on large datasets, their potential given small, annotated datasets (as is common in e.g. medical image analysis) has not been analyzed in much detail yet. We want to evaluate if GAN-based data augmentation using state-of-the-art methods, such as the Wasserstein GAN with gradient penalty, is a viable strategy for small datasets. We extensively evaluate our method on two image segmentation tasks: medical image segmentation of the left lung of the SCR Lung Database and semantic segmentation of the Cityscapes dataset. For the medical segmentation task, we show that our GAN-based augmentation performs as well as standard data augmentation, and training on purely synthetic data outperforms previously reported results. For the more challenging Cityscapes evaluation, we report that our GAN-based augmentation scheme is competitive with standard data augmentation methods.**

I. INTRODUCTION

Modern machine learning methods currently revolutionize our daily life. Especially *deep learning* [9] based methods consistently show improvements in the state-of-the-art every year, and for many *computer vision* tasks they even surpass human performance [6]. However, what most of these methods have in common is that they are *supervised*, therefore requiring *annotated* data for training. Furthermore, most deep learning methods benefit from a large amount of data to train, often in the range of hundreds of thousands of images. To circumvent the issue of insufficient annotated training data, common approaches such as *transfer learning* [3], *domain adaptation* [3] and *data augmentation* [13] can be followed. While transfer learning and domain adaptation are very popular, they are not as easily applicable for tasks where no large public datasets or pre-trained network parameters of a close domain are available, e.g. in medical image analysis. For this reason, we focus on data augmentation to deal with small amounts of data, in particular data augmentation using images synthesized from a generative model.

While data augmentation is most commonly done by using simple transformations, more sophisticated approaches for synthesizing additional training data have been proposed as well. For example, it has been shown that by rendering photorealistic, synthetic images and performing a set of transformations on those rendered images, they can be used to train an object detector with good performance [14]. Similarly, in the medical domain, it has been shown that by training a deep neural network on high-quality rendered 3D images from other computer vision tasks and fine-tuning it towards medical data, the general network performance can be improved when data is scarce [12]. This shows that data augmentation by using a generative model can improve the training of deep learning methods.

Recently introduced by Goodfellow et al., Generative Adversarial Networks (GANs) provide an attractive method of learning a generative model by training a deep neural network [4]. GANs have demonstrated potential in tasks such as state-of-the-art image generation [7], or synthetic data generation [16], [10]. The idea of using GANs in the context of data augmentation also saw some advancements in research, for example with the *SimGAN* [16] architecture. The main idea of SimGAN is to render synthetic images with corresponding labels (e.g. images of human eyes with their corresponding gaze direction) and refine those synthetic images with a *refiner*-GAN. This GAN uses the information from real, unlabeled images of the same domain while preserving the label information of the rendered images to generate realistic, refined images, which can further be used as training data for a supervised deep network. However, while GANs show impressive results when trained on large datasets, it is still a topic of active research how GANs behave when trained on a small amount of data, as most GAN-related research focuses on large datasets.

For this work, we will focus on data augmentation methods in the context of *image segmentation* tasks. As segmentation is a pixel-wise problem, the acquisition of annotated segmentation masks is even more time consuming, compared to e.g. classification tasks, as a human annotator has to label every pixel manually, which makes automated annotation methods highly desirable. Building upon the architecture we proposed in 2017 [10], we present a GAN-based data augmentation strategy incorporating state-of-the-art methods

*This work was supported by the Austrian Science Fund (FWF): P 28078-N33.

[1]Thomas Neff, Christian Payer, Darko Štern and Martin Urschler are with the Institute of Computer Graphics and Vision, Graz University of Technology, Austria thomas.neff@student.tugraz.at

[2]Darko Štern and Martin Urschler are with the Ludwig Boltzmann Institute for Clinical Forensic Imaging, Graz, Austria martin.urschler@cfi.lbg.ac.at

of GAN optimization, such as the Wasserstein GAN with gradient penalty (WGAN-GP) [5]. Our goal is to evaluate the segmentation performance of GAN-based data augmentation compared to standard data augmentation methods, especially in the case of small datasets. We perform experiments on two segmentation tasks, one from medical imaging, i.e. X-ray lung segmentation of the *SCR Lung Database* [17], and another, more challenging one, from computer vision, i.e. urban scene understanding of the *Cityscapes* [2] dataset. Additionally, we compare the segmentation performance when training with different ratios of real and generated data, to further evaluate the impact of GAN samples on the training process.

II. Related Work

A. Data Augmentation

Data augmentation is the process of generating additional training data from the available existing data [3]. Typically, this is done by using annotation-preserving transformations on the input data, such as randomly rotating, translating or deforming the image. Through the random nature of data augmentation, it can be used to potentially generate an 'infinite' amount of training data by augmenting the already existing data. For medical image analysis, data augmentation such as elastic deformation has been used with much success in combination with convolutional neural networks, as demonstrated by the *U-Net* [13] architecture for medical image segmentation. Although data augmentation is an effective way of dealing with the issue of small amounts of training data, it is not universally applicable, as prior knowledge of target domain and task is required to find a good data augmentation. Furthermore, the parametrization of data augmentation methods introduces another set of important hyperparameters, which can have a significant impact on the error made by the deep learning method.

B. Generative Adversarial Networks

Due to their end-to-end nature, good generated image quality and compatibility with modern deep learning techniques, GANs are the current state-of-the-art for generative models. A GAN consists of two subnetworks: the *generator* G, and the *discriminator* D, which play against each other in a two-player minimax game [4]. The generator synthesizes data from an input noise vector $\mathbf{z} \sim p_z$. The discriminator is a standard classification network, which receives real data $\mathbf{x} \sim p_R$, as well as data from the generator $G(\mathbf{z})$ as input. The goal of the discriminator is to perfectly classify each input image as either *real* or *synthetic*, while the goal of the generator is to synthesize images as close as possible to the real data, which leads to the discriminator misclassifying real and generated data [4].

As GAN optimization requires finding a Nash Equilibrium [4] between the generator and discriminator, GANs have historically been very unstable to train, which led to a lot of research focused on improving their training stability and generated image quality. Especially WGAN-GP [5] enjoys large popularity, due to being stable across different datasets and network architectures, as well as producing high quality images. The main idea behind WGAN is to use the Wasserstein-1 distance as its optimization criterion, which intuitively computes the cost of the optimal transport plan to transform the real data distribution p_R to the generator distribution p_G. Further improving on WGAN, WGAN-GP approximates the Wasserstein-1 by using a soft penalty on the gradient norm of the discriminator/critic. For this gradient norm, Gulrajani et al. [5] sample uniformly from a distribution $p_{\hat{\mathbf{x}}}$ that is defined along the lines between pairs of samples from the real data distribution p_R and the model distribution p_G, leading to the following optimization function:

$$L = \underbrace{\mathbb{E}_{\mathbf{x}\sim p_R}\left[D(\mathbf{x})\right] - \mathbb{E}_{\mathbf{z}\sim p_z}\left[D\left(G(\mathbf{z})\right)\right]}_{\text{WGAN critic loss}} + \underbrace{\lambda\, \mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}}\left[\left(||\nabla_{\hat{\mathbf{x}}} D(\hat{\mathbf{x}})||_2 - 1\right)^2\right]}_{\text{WGAN-GP gradient penalty}}, \tag{1}$$

where λ describes the gradient penalty coefficient and $\mathbb{E}$ denotes the expectation operator.

III. Method and Implementation

The standard GAN definition only allows for the generation of images, without respective labels. Therefore, for the generated data to be used for data augmentation, the conventional GAN formulation needs to be modified to also generate corresponding labels. For this modification, we build upon our previously proposed GAN architecture, which jointly generates images and their corresponding segmentation masks, for direct use of training data augmentation [10]. Compared to the standard GAN formulation, this architectural adaptation is a simple change in the network architecture, and can therefore be used with any GAN training scheme, such as WGAN-GP.

The main idea of this architecture is to fuse the image and segmentation mask to create an *image-segmentation pair*. This is done by concatenating both images along the channel axis. When training the GAN, the generator is now modified to generate image-segmentation pairs, instead of just images. The discriminator follows a similar principle, and now takes image-segmentation pairs as input, and its goal is to correctly decide if any given image-segmentation pair is real or synthetic. Therefore, the first convolutional layer of the discriminator needs to be modified to accept inputs, where the number of channels is equal to the number of channels of the image-segmentation pair.

All GANs in our experiments are trained using the WGAN-GP training scheme and loss function, as we found this to be the most robust method for training GANs, even across multiple datasets. For the gradient penalty hyperparameter, we used the default value suggested by Gulrajani et al., setting $\lambda = 10$. Additionally, compared to [10], we increase the image resolution in order to test how well the GAN is able to handle higher resolutions. Our code is based

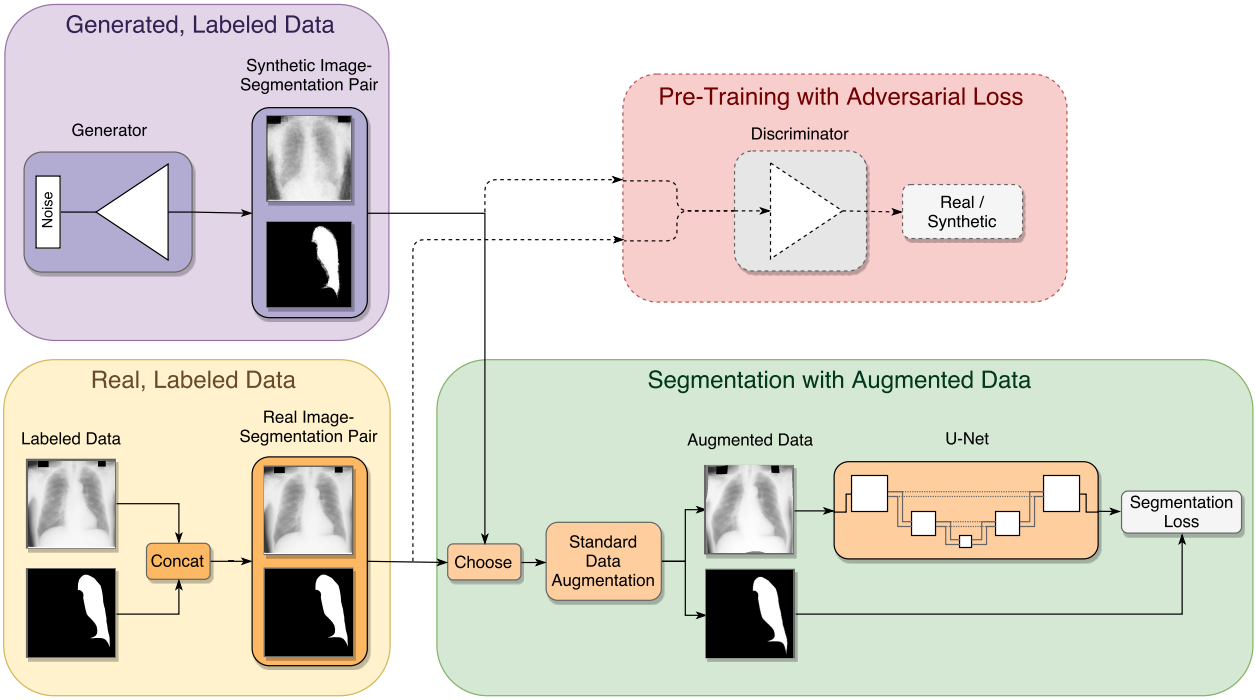

Fig. 1: Evaluation setup containing a pre-trained GAN generator producing image-segmentation pairs and a U-Net style segmentation network. The dashed lines and boxes illustrate the pre-training part of our setup.

on the code provided by the authors of WGAN-GP [5][1], using the TensorFlow [1] deep learning framework. For our evaluations, we use the following evaluation setup. First, we split the data into one or multiple training, validation, and test sets. Then, we train GANs for every training set of this dataset, until the generated image quality does not further improve. Finally, we take the fully-trained generator network, and use it directly as an input to a U-Net based [13] segmentation network. Compared to our previous evaluation [10], where a fixed amount of image-segmentation pairs was sampled, this on-the-fly generation allows for a much larger range of images to be sampled from the generator, better capturing the variation that the generator has learned from the data. For training the segmentation network, we use different ratios of real and generated data, and apply either no additional standard data augmentation, or a combination of standard data augmentation methods composed of *intensity shifts*, *intensity scaling*, *random translations*, *horizontal flipping* and *elastic deformations*. When mixing real and generated data, we exclusively use the specific GAN that was trained on the same real training data set, to keep all training sets separate. Our evaluation setup is illustrated in Figure 1.

IV. Implementation Details

Our GAN network architecture is based on DCGAN [11], only modified to generate image-segmentation pairs instead of just images. For the SCR Lung Database, our U-Net based segmentation network consists of 3 levels and uses a constant number of 64 filters for every convolutional layer. For the Cityscapes dataset, we use a U-Net based network with 4 levels and a constant number of 256 filters for every convolutional layer. We use Adam [8] as our optimizer for all networks, using a learning rate of $\eta = 0.0001$ and decay rates of $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for the first and second moments, respectively. All network weights were initialized using the *He* initializer [6]. We use ReLU activations in all layers, except for the final generator layers, which use tanh activations. In contrast to our GANs, we do not use Batch Normalization in our segmentation networks.

For every segmentation network, we train for at least 3000 iterations. During training, we keep track of the minimal validation loss and its iteration number, as well as the network parameters at the validation loss minimum. Every time a new validation loss minimum is found, we train for at least 3000 more iterations. In practice, this resulted in a good compromise of network performance and training time. The final metric for our evaluation is the segmentation performance of the segmentation network on the unseen test data. For the evaluation on the test set, and therefore the final segmentation performance, we upsample the output of our segmentation networks using bicubic upsampling to the target resolution. Afterwards, we compute the Dice coefficient and the Hausdorff distance (for the SCR Lung Database), as well as the mean Intersection-over-Union (mIoU) (for the Cityscapes dataset) as our evaluation metrics. All evaluations were done on an NVIDIA Tesla K80 with

[1]GitHub: Improved Training of Wasserstein GANs, `https://github.com/igul222/improved_wgan_training`, Accessed: 14.03.2018

12GB of GPU memory, although all networks were designed for an NVIDIA GTX980M with 8GB of GPU memory. For both evaluations, all input images were intensity-normalized to a range of $[-1, 1]$.

V. Evaluation

A. Lung Segmentation of the SCR Lung Database

1) Dataset Description: The *SCR Lung Database* [17] is a dataset consisting of 247 chest X-ray images, taken from the JSRT database [15]. Its image resolution is $[2048 \times 2048]$ at a physical resolution of 0.175 mm per pixel in each dimension, and it contains groundtruth segmentation masks for 5 objects: both lungs, the heart and both clavicles. For our evaluation, we chose the task of segmenting the left lung from the image.

2) Evaluation Setup: All images are downsampled to a resolution of $[256 \times 256]$ before we use them for training in order to fit all our networks into GPU memory while still being able to use a large enough minibatch size for stable training. We shuffle the dataset randomly and split it into 3 folds, each containing 135 training images, 30 validation images and 82/83 test images, chosen such that all images are contained exactly once in the set of test images. For the final evaluation of the segmentation performance, we report performance as the average Dice score and Hausdorff distance over all folds.

As the first step of our evaluation, we train our modified GAN for each of the 3 folds of training data, resulting in 3 fully-trained GANs. As it is difficult to determine a quantifiable stopping criterion for the training of GANs, every GAN was trained for a fixed number of 10000 iterations, which took approximately 24 hours per GAN. The raw image-segmentation pairs from the generator are in the intensity range of $[-1, 1]$. Therefore, when training our segmentation network using generated images, we threshold all segmentation masks at 0 when computing the segmentation loss.

For the main part of our evaluation of the SCR Lung Database, we train multiple segmentation networks for every fold, using an exhaustive set of combinations of real and generated data as well as with and without standard data augmentation. We evaluated different combinations of standard data augmentation on one fold of the cross-validation set to find suitable augmentation parameters for the final comparison. To speed up this parameter search, we fixed elastic deformation at 10 pixels for each control point. The results for different augmentation methods are shown in Table I, and the combination of parameters listed in bold are used as our standard data augmentation method for the final evaluation. Important to note is that this parameter search was done on only a single fold of the validation set, therefore those results are not comparable to our final quantitative results which are averaged over all folds. Before computing our final segmentation performance metrics, we extract only the largest connected component. As the left lung is only a single connected component in all our images, this reduces false predictions of the resulting segmentation masks. Training each segmentation network took approximately 8 hours on our setup.

TABLE I: Comparison of augmentation parameters for the SCR Lung Database. For further evaluation, we use the augmentation parameters listed in **bold** as our standard data augmentation.

Augmentation Parameters				Validation Performance
Intensity shift around zero (stddev)	Intensity scaling around one (stddev)	Random translation around zero (stddev)	Elastic deformation around zero (stddev)	Dice (mean)
-	-	-	-	96.98%
-	-	-	10 px	97.06%
-	-	10 px	10 px	96.85%
-	0.05	-	10 px	97.03%
-	0.05	10 px	10 px	96.77%
0.05	-	-	10 px	96.40%
0.05	-	10 px	10 px	96.91%
0.05	0.05	-	10 px	96.63%
0.05	**0.05**	**10 px**	**10 px**	**97.12%**

3) Results: Example images from our GANs trained on the SCR Lung Database can be seen in Figure 2. The final segmentation performance, averaged across all folds, is shown in Table II. In order to better compare GAN-based data augmentation and standard data augmentation, we also present examples of resulting segmentation masks for two of our networks: the network trained on a mix of real and generated data without data augmentation (GANs-based augmentation), and the network trained solely on real data with standard data augmentation. Some of the best resulting examples, as well as the example showing the worst performance are shown in Figure 3.

B. Semantic Segmentation of the Cityscapes Dataset

1) Dataset Description: For our second evaluation, we chose the task of semantic segmentation using the *Cityscapes* [2] dataset. Cityscapes is a challenging dataset for semantic urban scene understanding, which aims to capture the complexity of real-world urban scenes. For 30 object classes divided into 8 groups, pixel-level and instance-level segmentation masks are provided for every image. The base resolution of all images is $[2048 \times 1024 \times 3]$. This dataset consists of 2975 training images and 500 validation images with finely annotated segmentation masks, with an online submission system used to evaluate performance on the test set, for which the groundtruth segmentation masks are not known. Since this segmentation problem is much more challenging compared to the lung segmentation problem of the SCR Lung Database, we decided to only do segmentation of the 8 object groups (*'categories'*) defined in the Cityscapes dataset, and not on the individual classes.

2) Evaluation Setup: We created our own data split from the given training and validation sets. We used all 500 images from the Cityscapes validation set as our test set. For our internal validation set, we randomly selected 400 images from the Cityscapes training set. Finally, our training set consisted of the remaining 2575 images from the Cityscapes training set. Due to the much larger amount of data compared to the SCR Lung Database and the time consuming nature

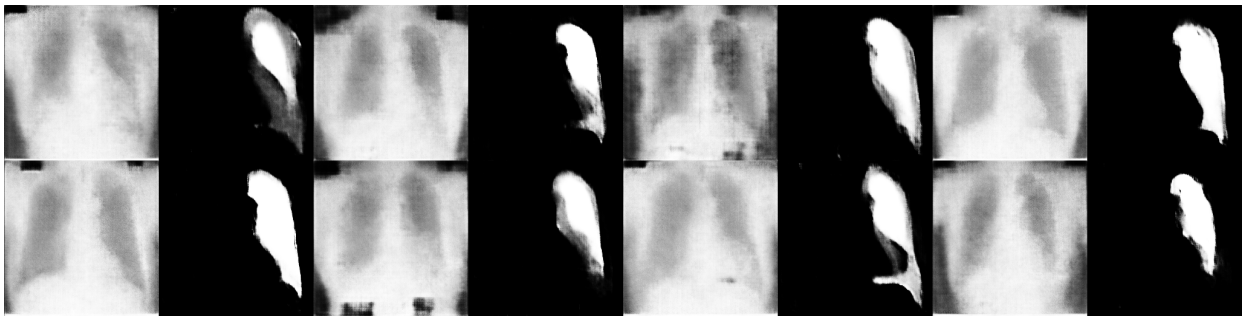

Fig. 2: Example images from our GAN trained on the SCR Lung Database. Odd columns show generated images, while even columns show the respective generated segmentation masks.

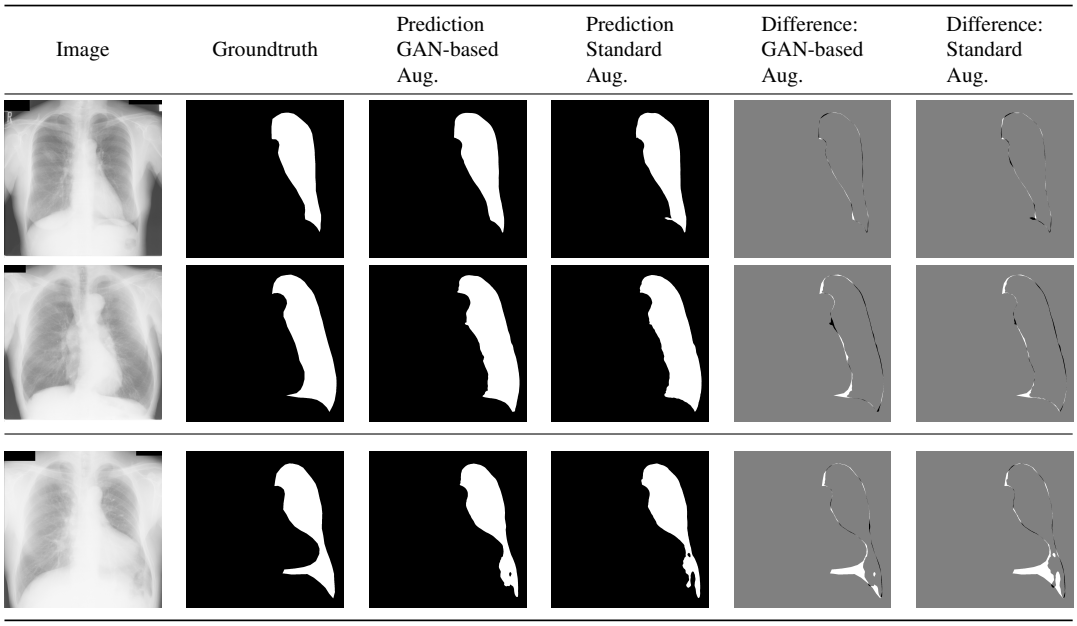

Fig. 3: Comparison of segmentation masks from fully trained segmentation networks between standard data augmentation and GAN-based data augmentation for the SCR Lung Database. Rows 1 and 2 show good results, while row 3 shows the worst performing test example. Columns 3 and 5 show results from our segmentation network trained using GAN-based augmentation with a mix of real and generated data, while Columns 4 and 6 show the results of our segmentation network trained using standard data augmentation.

TABLE II: Segmentation performance comparison between training on real data, generated data, and mixed data, using either no additional data augmentation, or standard data augmentation (see Table I), evaluated on our test set of the SCR Lung Database. Since the previous work of Neff et al. [10] was not tested on full resolution, the Hausdorff distance was omitted from these results, as it is not an accurate comparison.

Network ID	# real pairs in minibatch	# generated pairs in minibatch	Aug.?	Dice (mean)	Dice (stddev)	Hausdorff (mean)	Hausdorff (stddev)
16-0-Aug	16	0	yes	97.65%	1.65%	1.2057 mm	0.3131 mm
16-0-NoAug	16	0	no	97.42%	1.66%	1.2626 mm	0.3440 mm
8-8-Aug	8	8	yes	97.65%	2.28%	**1.1722 mm**	0.3313 mm
8-8-NoAug	8	8	no	**97.68%**	1.47%	1.2106 mm	0.3154 mm
0-16-Aug	0	16	yes	96.55%	1.63%	1.2651 mm	0.3067 mm
0-16-NoAug	0	16	no	96.32%	2.02%	1.3273 mm	0.3434 mm
Neff et al. [10]	16	0	no	96.08%	1.01%	-	-
	≈ 8	≈ 8	no	95.37%	1.21%	-	-
	0	16	no	91.72%	2.83%	-	-

TABLE III: Comparison of augmentation parameters for the Cityscapes dataset. For further evaluation, we use the augmentation parameters listed in **bold** as our standard data augmentation.

Augmentation Parameters				Validation Performance
Intensity shift around zero (stddev)	Intensity scaling around one (stddev)	Random translation around zero (stddev)	Horizontal flipping	mIoU
0.10	0.10	10 px	no	69.89%
0.05	0.05	5 px	no	74.21%
-	-	-	no	78.04%
0.05	0.05	-	yes	79.50%
-	**-**	**-**	**yes**	**80.42**%

of our evaluation, we only evaluate on this single fold of data. For all networks in this evaluation, we downscaled the resolution of all input images to $[256 \times 128 \times 3]$ to be able to fit our generator network and our segmentation network into memory at the same time, while still keeping a sufficiently large minibatch size for training stability. The Cityscapes dataset contains a lot of small, thin structures (e.g. objects such as street lights and traffic signs), that get reduced to just a few pixels in size when downsampling to such an extent. This is especially apparent for the *'Human'* and *'Object'* categories, which contain the smallest objects in the dataset, such as pedestrians and street lights.

Before training our segmentation networks, we train our modified GAN on the Cityscapes dataset for 10000 iterations, as the image quality did not improve further after that. For the standard data augmentation, we experimented using a set of multiple different augmentation methods, and chose the best one based on the performance on the validation set. The validation results for different augmentation methods are shown in Table III, and the combination of parameters listed in bold are used as our standard data augmentation method for further training.

We threshold the output segmentation images of the generator to get discrete segmentation masks.

For our final evaluation of the Cityscapes dataset, we train our segmentation network on different ratios of real and generated data, similar to the previous experiment, using either no augmentation or standard data augmentation. Each segmentation network took approximately 24 hours to train until convergence.

3) Results: Our final segmentation performance for all different evaluation setups of the Cityscapes dataset is shown in Table IV. Additionally, we show the resulting mIoU for every category for the four best performing networks in Figure 4. Since the significant amount of downsampling of the input images results in small objects vanishing or being reduced to single-pixel size, and therefore not being useful for training, we also present mIoU results excluding the 'Human' and 'Object' categories in Table IV. Similar to our evaluation of the SCR Lung Database, in Figure 5 we show an example of a resulting segmentation mask to better compare GAN-based augmentation to standard augmentation.

TABLE IV: Segmentation performance comparison between training on real data, generated data, and mixed data, using either no additional data augmentation, or standard data augmentation, evaluated on our test set of Cityscapes.

Network ID	# real pairs in minibatch	# generated pairs in minibatch	Aug.?	mIoU	mIoU excluding 'Human' and 'Object'
8-0-Aug	8	0	yes	**78.59**%	**88.94**%
8-0-NoAug	8	0	no	76.16%	87.67%
4-4-Aug	4	4	yes	75.48%	87.32%
4-4-NoAug	4	4	no	76.30%	87.86%
0-8-Aug	0	8	yes	47.05%	65.35%
0-8-NoAug	0	8	no	46.32%	64.26%

VI. Discussion and Conclusion

Our main focus of this work was to perform a comparison between standard data augmentation and GAN-based data augmentation. For the first comparison, we chose to perform medical image segmentation of the SCR Lung Database, as this allows us to directly compare to previously reported results. Figure 2 shows that for the SCR Lung Database, our GAN manages to generate high-quality images with corresponding segmentation masks that fit the generated image well. Compared to our previously published GAN examples of this dataset shown in [10], the generated samples are of much higher quality and more closely resemble the training data. We also do not experience mode collapse of our generated samples compared to these previous results, as the resulting samples show similar variety to the training set the generator was trained on. Looking at the segmentation performance shown in Table II, we can see that the Dice scores and Hausdorff distances are very close between all networks, only showing a significant gap for the networks trained on strictly synthetic data and for our previous results in [10]. Augmenting with synthetic images from a trained GAN does not decrease the segmentation performance, and networks trained with a mix of synthetic and real images stay competitive with networks trained on strictly real data, using standard data augmentation. Even though the difference is small, the best result (Dice score, standard deviation of Dice) of our evaluation was achieved using our GAN-based augmentation, i.e. using a network trained on mixed real and synthetic data. This suggests that GAN-based augmentation might be a viable augmentation strategy in the future, especially if GAN research further improves on the quality and variety of generated images. Furthermore, it is very interesting to see that our network trained on purely synthetic data achieves better results compared to the network trained on real data of our previous work [10]. This is mostly due to the higher quality of the synthetic images sampled from our GAN on-the-fly. This shows that our GAN has managed to learn enough about the underlying training data distribution to produce valuable images for training segmentation networks.

Looking at the samples produced from our GAN-

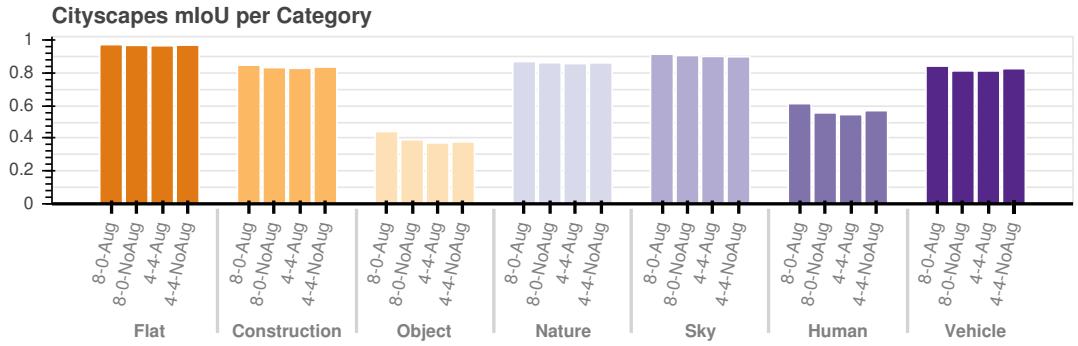

Fig. 4: mIoU for every category of the Cityscapes dataset. For visual clarity, we omit the worst performing networks and only report results for the four best networks, identified by their network ID shown in Table IV.

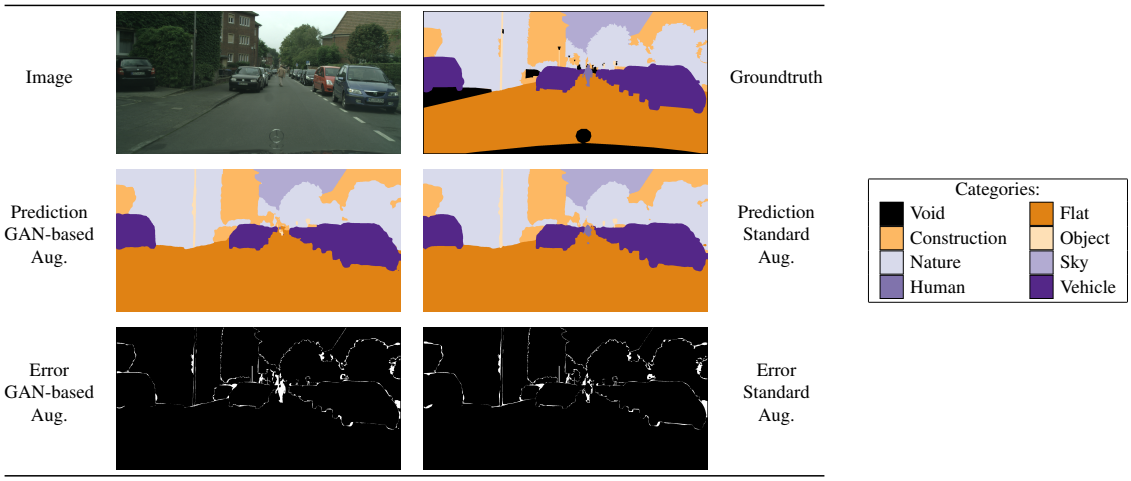

Fig. 5: Comparison of segmentation masks from fully trained segmentation networks between standard data augmentation and GAN-based data augmentation for the Cityscapes dataset for the best performing test image.

augmented network and our network trained with standard augmentation in Figure 3, we can see that the segmentation quality is also equally good. For some test images, the network trained with GAN-based data augmentation produces better segmentation masks, while for others, the network trained with standard data augmentation achieves higher quality results. Since the Dice scores and Hausdorff distances are almost identical, and we cannot determine significant differences in image quality, it seems that the lung segmentation problem for this dataset is already very well modeled by the U-Net. Additional augmentation does not provide any more benefits, but also does not have a negative impact on the results either. However, GAN-based augmentation also does not lead to worse performance in this case, which was not the case in our evaluation presented in [10], suggesting that the higher quality GAN images from WGAN-GP improved the overall augmentation method, and our GAN managed to better capture the distribution of our training data.

For the evaluation on the Cityscapes dataset, results are interesting as well. While our GAN managed to generate images of reasonable variety, the image quality is not as high, as the Cityscapes dataset is more complex, and therefore more difficult to learn for a generative model. Looking at the quantitative evaluation of the Cityscapes dataset, we found that the best standard data augmentation for this dataset and our segmentation network architecture was to just use horizontal flipping (see Table III). Using other combinations of intensity shift, intensity scaling, or random translation led to worse segmentation performance. For some settings, the segmentation performance was even worse than not using data augmentation at all. Similarly, we can observe that using horizontal flipping in combination with our GAN-based approach (*4-4-Aug*) leads to worse performance compared to just using GAN-based augmentation (*4-4-NoAug*). This illustrates an important point of data augmentation - the augmentation parameters require careful tuning to fit the dataset, as unsuitable data augmentation can have a negative effect by drastically reducing the segmentation performance. From our final segmentation performance shown in Table IV, we can see that the network trained on real data, using horizontal flipping for data augmentation (see Table III), achieved the best performance compared to all other networks. However, we can again observe that the network trained using GAN-

based augmentation without additional standard data augmentation achieves similar performance to the network that was trained on real data without augmentation. Especially interesting is that when using GAN-based augmentation without standard augmentation, the results are better than some of the results when using standard augmentation shown in Table III. This illustrates that our GAN has learned a reasonable representation of our training data, even though the generated samples are not of high quality.

Compared to the highscore database of the Cityscapes dataset[2], our baseline performance for the category mIoU is in line with the weaker results on the online database. This is mostly due to the large amount of downsampling we perform on the input images, as well as that we do not use pre-trained networks as all other competing methods do. Because one of our main goals was to evaluate how GAN-based data augmentation affected the results of training segmentation networks, we did not want to additionally pre-train our networks, as that would introduce another variable that significantly impacts training behavior of deep networks.

Performing large amounts of downsampling leads to a lower segmentation performance for small or thin structures, and borders between regions, as those fine details vanish when downsampling is applied. This effect can be seen by comparing the resulting segmentation masks of our networks, shown in Figure 5. Most of the errors of our results are in the border regions between classes, as the fine detail necessary to determine exact borders is lost during downsampling. We also observe the consequence of downsampling in Figure 4, where we show results for every category. While our networks consistently show weaker performance on the 'Object' and 'Human' categories, the other categories show good results, given that we only used a standard U-Net segmentation network architecture with additional data augmentation methods. Computing the mIoU over all categories but those two, we achieve much better scores, as can be seen in Table IV.

To conclude, we performed an extensive evaluation of the possibilities of using GANs for training data augmentation in image segmentation tasks. From our current results, we cannot conclude GAN-based augmentation has a positive or negative impact, but we believe that if a GAN was able to fully learn the training data distribution, the additional synthetic data could be highly useful as a regularizer for deep networks. Compared to standard data augmentation, GAN-based augmentation does not require extensive data analysis to find out optimal augmentation parameters. Especially in the Cityscapes evaluation, we saw how certain data augmentation parameters can lead to much worse performance, therefore an augmentation method that is learned from data would save a lot of effort in fine-tuning deep networks. The most straight-forward future improvement would be to increase the resolution and representation power of our GAN, leading to higher quality synthetic images. We are certain that such an improved generative model could be used as a data augmentation method to improve performance for supervised deep learning tasks.

[2]Cityscapes Pixel-Level Semantic Labeling Task Results, {https://www.cityscapes-dataset.com/benchmarks/\#pixel-level-results}, Accessed: 14.03.2018

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI)*. USENIX Association, 2016, pp. 265–283.

[2] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27 (NIPS)*. Curran Associates, Inc., 2014, pp. 2672–2680.

[5] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems 30 (NIPS)*. Curran Associates, Inc., 2017, pp. 5769–5779.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2015, pp. 1026–1034.

[7] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," *Proceedings of the Sixth International Conference on Learning Representations (ICLR)*, 2018.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the Third International Conference on Learning Representations (ICLR)*, 2015.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[10] T. Neff, C. Payer, D. Štern, and M. Urschler, "Generative Adversarial Network based synthesis for supervised medical image segmentation," in *Proceedings of the OAGM&ARW Joint Workshop*, 05 2017, pp. 140–145.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with Deep Convolutional Generative Adversarial Networks," in *Proceedings of the Fourth International Conference of Learning Representations (ICLR)*, 2016.

[12] G. Riegler, M. Urschler, M. Rüther, H. Bischof, and D. Štern, "Anatomical landmark detection in medical applications driven by synthetic data," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 85–89.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer, Cham, 2015, pp. 234–241.

[14] A. Rozantsev, V. Lepetit, and P. Fua, "On rendering synthetic images for training an object detector," *Computer Vision and Image Understanding (CVIU)*, vol. 137, pp. 24 – 37, 2015.

[15] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K.-i. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi, "Development of a digital image database for chest radiographs with and without a lung nodule," *American Journal of Roentgenology (AJR)*, vol. 174, no. 1, pp. 71–74, Jan. 2000.

[16] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2242–2251.

[17] B. van Ginneken, M. Stegmann, and M. Loog, "Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database," *Medical Image Analysis*, vol. 10, no. 1, pp. 19–40, 2006.