Proceedings of the OAGM&ARW Joint Workshop 2017 DOI: 10.3217/978-3-85125-524-9-30

Generative Adversarial Network based Synthesis for Supervised Medical Image Segmentation*

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

Abstract— **Modern deep learning methods achieve state-of-the-art results in many computer vision tasks. While these methods perform well when trained on large datasets, deep learning methods suffer from overfitting and lack of generalization given smaller datasets. Especially in medical image analysis, acquisition of both imaging data and corresponding ground-truth annotations (e.g. pixel-wise segmentation masks) as required for supervised tasks, is time consuming and costly, since experts are needed to manually annotate data. In this work we study this problem by proposing a new variant of Generative Adversarial Networks (GANs), which, in addition to synthesized medical images, also generates segmentation masks for the use in supervised medical image analysis applications. We evaluate our approach on a lung segmentation task involving thorax X-ray images, and show that GANs have the potential to be used for synthesizing training data in this specific application.**

I. INTRODUCTION

Modern machine learning methods based on deep neural network architectures require large amounts of training data to achieve the best possible results. For standard computer vision problems, large datasets, such as MNIST [12], CIFAR10 [10], or ImageNet [23], containing millions of images, are publicly available. In the medical field, datasets are typically smaller by several orders of magnitude, as the acquisition process of medical images is costly and time consuming. Furthermore, ethical concerns make it harder to publicly release and share datasets.

Finding methods to improve performance when training deep learning methods on small datasets is an area of active research. Recent work in the medical imaging domain has shown that it is possible to improve performance with small datasets by putting application specific prior knowledge into a deep neural network [17]. Another approach has been made popular by the U-Net [21] architecture for biomedical image segmentation, which demonstrated how strong data augmentation can be used to deal with low amounts of training data in deep network architectures. Even though data augmentation is simple to implement and achieves good results, it is only able to produce fixed variations of any given dataset, requiring the augmentation to fit the given dataset.

Transfer learning approaches such as [19] show that training on large datasets (e.g. ImageNet) followed by fine-tuning on a small dataset achieves state-of-the-art results for datasets consisting of natural images. For medical imaging, the learned features from large natural image datasets may not be suitable, as the image features are very different compared to natural images. Furthermore, there is no straightforward way of transferring 2D features to 3D features, which poses a limitation when working with 3D medical images. Due to the difference in features between medical and natural images, another approach is to use unsupervised feature extractors (e.g. Autoencoders [27]) which are trained on medical images only. Nevertheless, transferring weights learned by these unsupervised methods requires the target network architecture to be close to the source architecture, which is rarely the case.

The requirement for large amounts of training data also popularized image generation methods in deep learning contexts. Recently, research has shown that Generative Adversarial Networks (GANs) [4] can be used for a large variety of applications such as image-to-image translation [6] or unsupervised representation learning [18]. GANs have also been successfully used for unsupervised domain adaptation [8] of multi-modal medical imaging data, demonstrating their potential for use with small medical imaging datasets.

Our goal was to use GANs in a completely different way, by using the high quality of the generated images to augment our small set of training data. We propose a novel modification to GANs, which generates new, synthetic images as well as the corresponding segmentation masks from random noise. This allows us to use the synthetic data as training data for a supervised segmentation task. We show that this architecture manages to produce convincing segmentation masks for the generated images. We evaluate the generated images in two different scenarios on an image segmentation task and show that training on purely generated images achieves results comparable to training on real images for very small datasets.

II. RELATED WORK

A. Training Data Augmentation

Training data augmentation is a commonly used method to reduce the effects of overfitting with small training datasets as well as improve the generalization of the trained network. Most machine learning frameworks allow for simple augmentation such as rotation, translation or intensity shifts of training data. AlexNet [11] was one of the first convolutional neural network (CNN) architectures to implement online data augmentation with successful results. However, data augmentation only achieves good results if the augmentation can actually occur in the data, and is relevant to the required application. For medical imaging, elastic deformations [21]

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

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

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

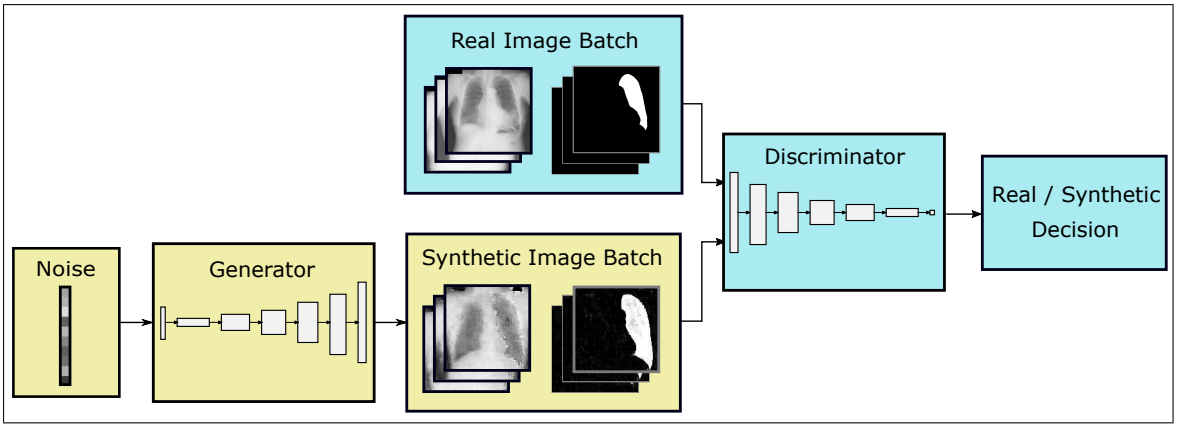

Fig. 1. Proposed GAN architecture incorporating the segmentation mask in the real and synthetic image batches

are especially useful for biomedical segmentation, as they can provide realistic variations of the input data, similar to natural variations.

B. Transfer Learning

Transfer learning aims to improve the learning of a target task in a target domain, given the learned knowledge of a source task in a source domain [16]. Applied to neural networks, it describes the process of training a source network on a source dataset, followed by transferring the learned features to train a different target network on a target dataset [28]. In the context of small datasets, this can be applied in different ways. It is possible to train on a large dataset, e.g. ImageNet, remove the final layer of the network architecture and fine-tune to a smaller target dataset [19]. A different approach is taken by using Autoencoders, which compress a given image to a vector representation and reconstruct the image from this compressed representation. As an example, denoising Autoencoders [27] have been used to extract robust features with great success. However, transferring Autoencoder features typically requires a target network architecture very similar to the source architecture, which is rarely the case.

C. Image Generation

A novel approach to tackle the issue of small datasets for training deep learning methods is to synthesize new training data via image generation methods. Recent research has shown that it is possible to render realistic images using 3D models to alleviate the problem of small datasets [22]. This has the advantage of being able to create an unlimited amount of training data of various scenarios, as long as the images are realistic enough. Rendered images have also recently been used to improve the performance of anatomical landmark detection in medical applications by learning on a dataset of rendered 3D models and fine-tuning on medical data [20]. The disadvantage of using rendered images is that the virtual model and scene parameters need to be explicitly defined and tuned towards the application, which is time consuming.

Generative Adversarial Networks [4] represent a different approach to image generation. A generator and a discriminator network are trained to compete against each other. The goal of the discriminator is to decide if any given image is real or synthetic. The generator generates synthetic images in the hope of fooling the discriminator. Since the generator never directly sees the training data and only receives its gradients from the discriminator decision, GANs are also resistant to overfitting [3]. However, the training process of GANs is very sensitive to changes in hyperparameters. The problem of finding the Nash Equilibrium between the generator and the discriminator generally leads to an unstable training process, but recent architectures such as DCGAN [18] and WassersteinGAN [2] improved on this substantially.

III. METHOD AND ARCHITECTURE

Standard GANs either exclusively learn to generate images [4], or learn to perform image transformations [6]. However, in order to use the generated images for other supervised deep learning tasks, like image segmentation, it is also necessary to have a ground-truth solution for any given input image.

We propose a modification to the standard GAN architecture, which forces the generator to create segmentation masks in addition to the generated images. The discriminator then has to decide whether an observed image-segmentation-pair is real or synthetic. This forces both the discriminator and generator to implicitly learn about the structure of the ground-truth, making the resulting generated data useful for training in a supervised setup. While it is known that using ground-truth labels in the discriminator improves the image quality [24], this is the first time, to our knowledge, that the ground-truth is used to directly generate new image-segmentation-pairs. Fig. 1 illustrates this architecture.

As the foundation for our proposed architecture, we use the DCGAN [18] architecture, which has shown to achieve good results while having increased training stability in many different applications, compared to the previous GAN architectures. DCGAN uses a convolutional generator and discriminator, makes use of batch normalization, and replaces

all pooling layers with convolutions. The generator takes a noise vector z as input and feeds it through multiple fractionally strided convolutions in a fully convolutional manner to generate synthetic images $G(z)$. The discriminator receives both real images x and synthetic images $G(z)$, feeds them through a fully convolutional classification network which classifies any given image as either real, i.e. $D = 1$, or synthetic, i.e. $D = 0$. The discriminator uses the cross entropy loss function

$$l_D = \frac{1}{m}\sum_{i=1}^{m}\left[log\left(D\left(G\left(z^{(i)}\right)\right)\right) + log\left(1 - D\left(x^{(i)}\right)\right)\right], \quad (1)$$

where the mini-batch size m describes the number of training inputs for stochastic gradient descent [15], i denotes the current index in the mini-batch, $x^{(i)}$ is the real image, $z^{(i)}$ is the noise vector sample, D is the discriminator output and G is the generator output. The generator loss

$$l_G = \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \quad (2)$$

only takes the discriminator output of the generated images $D(G(z))$ into account.

By minimizing l_G, the generator is trained to generate images $G(z)$ which look real, i.e. $D(G(z)) \approx 1$, while by minimizing l_D, the discriminator is trained to correctly classify real and synthetic images, i.e. $D(x) \approx 1$ and $D(G(z)) \approx 0$. Therefore, generator and discriminator play against each other, as the generator creates synthetic images which fool the discriminator into believing they are real, while the discriminator attempts to classify real and synthetic images correctly every time.

In order to implement the additional segmentation mask generation, the DCGAN architecture was modified to use 2-channel images, where the first channel corresponds to the image, and the second channel corresponds to the segmentation mask. The discriminator network then simply classifies image-segmentation-pairs instead of images only. The GAN therefore creates synthetic image-segmentation-pairs, which we then further use for the supervised training of a segmentation task. For most GAN setups, this change is simple to implement, as no change in the training process is necessary, making this adaptation very flexible.

IV. EVALUATION

A. Materials

We evaluate our proposed method using a 3-fold cross-validation setup on the SCR Lung Database [26], which is composed of the JSRT Lung Database [25] with corresponding ground-truth segmentation masks. The cross-validation splits are set up so that all 247 images are tested once, using 82 test images, and randomly picking 20 validation images and 145 training images from the remaining images. The images are downscaled to a resolution of 128x128, on which all evaluations are performed. In order to demonstrate possible strengths and limitations of the GAN for even smaller datasets, we evaluate different scenarios on the full dataset, as well as on a reduced dataset. For the reduced dataset, the cross-validation setup for test and validation data is the same as for the full dataset, only the amount of training data is reduced to 30 images by randomly picking them from the training images of the full dataset. For the quantitative evaluation, we chose to perform image segmentation using the U-Net [21] fully convolutional network architecture.

B. Experimental Setup

For our proposed GAN architecture, we adapted the DCGAN [18] TensorFlow [1] implementation tf-dcgan[1]. We modified the architecture to include support for the generation of segmentation masks and increased the image resolution to 128x128. The higher resolution made it necessary to increase the number of generator and discriminator feature maps. We also used a random noise vector z of higher dimension as the generator input. The noise vector dimension was fixed at 400, using uniform noise in the range of $[-1,1]$. Generator feature map sizes were set to $[512,256,128,128,128]$, discriminator feature map sizes were set to $[128,128,256,512,512]$. As suggested in [18], the convolutional kernel sizes were kept at 5. The weights of all convolutional layers were initialized randomly using a normal distribution with zero mean and a standard deviation of 0.05. The input data was scaled to be in the range of $[-1,1]$. The used optimizer was Adam [9] with a learning rate of 0.0004 and an exponential decay rate for the first and second moment estimates of $\beta_1 = 0.5$, $\beta_2 = 0.999$. The training was done using a mini-batch size of 128. The network was trained for 12000 mini-batches in total, as after 12000 mini-batches the overall quality of the generated images $G(z)$ was high for all cross-validation folds. Samples were generated every 200 mini-batches of training. To slightly reduce the impact of Mode Collapse [3], where the generator learns to map several different noise vector inputs z to the same output image $G(z)$, the resulting GAN images were checked for similarity by using a perceptual image hash, which removes images that are almost identical in a batch of samples. Training the GAN took approximately 24 hours per cross-validation fold on an Intel i7-6700HQ CPU @ 2.60 GHz and an NVidia GTX980M GPU with 8 GB of GPU memory.

For the quantitative segmentation results, we used a U-Net architecture of depth 4, replacing max pooling with average pooling for downsampling. This U-Net was implemented using Caffe [7]. Although data augmentation is used to great effect and is also described as a strength of the U-Net [21], we decided not to use it in any of our experiments, in order to specifically evaluate the impact the synthetic GAN samples have on the training process and the resulting segmentation masks. All convolution kernel sizes were set to 3, with feature map sizes of 64 and weights initialized using the MSRA [5] method. We used the Nesterov [14] optimizer at a learning rate of 0.00001 for the segmentation task, with a momentum of 0.99 and a weight decay of 0.0005. The

[1] https://github.com/sugyan/tf-dcgan

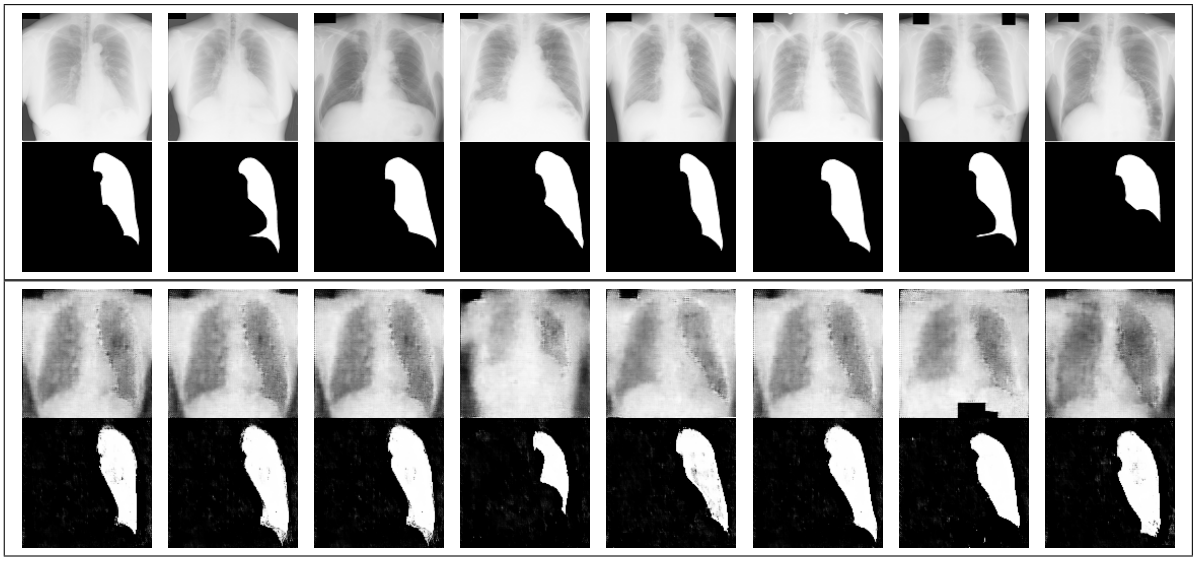

Fig. 2. Sample images and segmentation masks from the real training data (top) compared to synthetic data created by the GAN trained on the full training set (bottom)

mini-batch size was set to 16. The network was trained until the average of the validation error over the last 10 epochs started to increase. The input data was scaled to be in the range of $[-1, 1]$. Since the generated GAN images and segmentation masks are in the value range of $[-1, 1]$, the resulting segmentation image needs to be post-processed to arrive at a binary segmentation mask, which can then be used as an input for the U-Net. To achieve this post-processing, a threshold, largest component and hole filling filter were applied to the generated GAN segmentation masks before they were fed into the U-Net. The threshold was set at the pixel value of 150, and the hole-filling algorithm used is based on geodesic morphology as described in Chapter 6 of [13]. We tested the segmentation performance when using only real training data, a mix of real and synthetic data, as well as only synthetic data. For the synthetic data, we generated a batch of 120 images and segmentation masks from the fully trained GAN. For evaluating the segmentation results, we used the Dice coefficient and Hausdorff distance metrics. Training the U-Net took approximately 3 hours per experiment on the same machine as described above.

C. Results

For the full dataset, Fig. 2 illustrates generated images and segmentation masks from the fully trained GAN, compared to real images and segmentation masks. The quantitative evaluation results for the full dataset can be seen in Table I.

For the reduced dataset, the quantitative evaluation results are shown in Table II.

V. DISCUSSION AND CONCLUSION

Small datasets pose large issues for deep learning methods, leading to overfitting and lack of generalization. We propose an adaptation of Generative Adversarial Networks, where the generator network is trained to generate artificial images in addition to their corresponding segmentation masks. While the qualitative results shown look very promising, they also heavily depend on the amount of training the GAN receives. Fig. 2 shows that using a fully trained GAN to create segmentation data in addition to image data still leads to high quality images. The segmentation also matches the generated images very well, suggesting that both the generator and discriminator are forced to learn the structure of the segmentation as well. However, it can also be seen that small noise artefacts appear in the region of the left lung of the image. These artefacts do not appear if the GAN

TABLE I
QUANTITATIVE RESULTS OF SEGMENTATION USING THE FULL TRAINING SET

U-Net training data		Evaluation metrics			
# Real	# Synthetic	Dice (mean)	Dice (stddev)	Hausdorff (mean)	Hausdorff (stddev)
145	0	**0.9608**	0.0101	**6.1229**	5.0183
145	120	0.9537	0.0121	6.3147	4.8708
0	120	0.9172	0.0283	9.3564	6.0651

TABLE II
QUANTITATIVE RESULTS OF SEGMENTATION USING THE REDUCED TRAINING SET

U-Net training data		Evaluation metrics			
# Real	# Synthetic	Dice (mean)	Dice (stddev)	Hausdorff (mean)	Hausdorff (stddev)
30	0	**0.9464**	0.0158	7.6384	6.0395
30	120	0.9394	0.0133	**7.2885**	5.1007
0	120	0.9312	0.0199	7.6091	5.5654

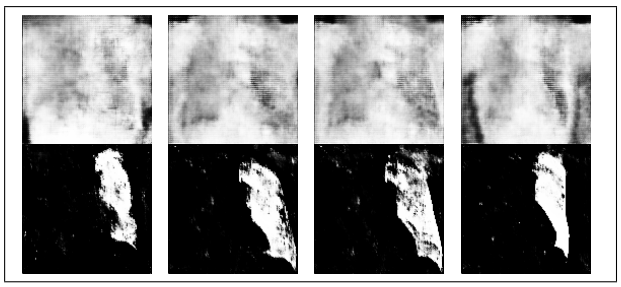

Fig. 3. Sample images and segmentation masks generated by the GAN trained on the full dataset if the training is stopped too early

is trained without generating segmentation masks. We also experience a mild form of Mode Collapse [3], as some of the generated images look very similar. While the images obtained by the fully trained GAN shown in Fig. 2 have a high quality, Fig. 3 illustrates that, if the training time for the GAN is too short, generated images are unusable for later supervised training, as the image quality is too low. Finding a suitable stopping point for GAN training is still a hot topic of current research, as a lower GAN loss during training typically does not indicate higher image quality of the generated images. However, recent modifications to the GAN learning process show that it is possible to correlate the GAN loss with image quality [2], which enables the possibility of stopping the GAN training once the loss is under a certain threshold.

The results of the quantitative evaluation on the full dataset shown in Table I indicate that the GAN images are not sufficient to replace the real images in this case. Using a combination of real and synthetic images to train our segmentation network, the Dice score and Hausdorff distance results are comparable to the results obtained by training on real images only. When only synthetic images obtained by the GAN are used to train the segmentation network, the performance is worse. For the reduced dataset evaluation, the results shown in Table II are not as conclusive. The network with the best Dice score was trained exclusively on real images, while the network with the lowest Hausdorff distance was trained on a combination of real and synthetic images. A very interesting point, however, is that for the reduced dataset, the network trained exclusively on generated GAN images performed almost as well as the network trained on real images, showing significant potential of GANs for training data generation. It is also worth mentioning that the U-Net trained exclusively on generated GAN images from the reduced dataset performed better than the U-Net trained exclusively on generated GAN images from the full dataset. We suspect that this is because the GAN has an easier time to converge to generating high quality images for the reduced dataset compared to the full dataset, leading to better image quality of the generated images.

The quantitative results still have room for improvement. As a further outlook, it would be interesting to incorporate data augmentation in the GAN by using elastic deformations to induce variance in the GAN's training data, which may potentially lead to a greater variety of generated GAN images. Overall, we demonstrated that GANs have significant potential for synthesis of medical training data for supervised tasks by learning to generate segmentation masks in addition to artificial image data.

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*, ser. OSDI'16. Berkeley, CA, USA: USENIX Association, 2016, pp. 265–283. [Online]. Available: http://dl.acm.org/citation.cfm?id=3026877.3026899

[2] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," *ArXiv e-prints*, Jan. 2017.

[3] I. Goodfellow, "NIPS 2016 Tutorial: Generative Adversarial Networks," *ArXiv e-prints*, Dec. 2016.

[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*, 2014, pp. 2672–2680.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 1026–1034. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.123

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," *ArXiv e-prints*, Nov. 2016.

[7] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM '14. New York, NY, USA: ACM, 2014, pp. 675–678. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654889

[8] K. Kamnitsas, C. Baumgartner, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, A. Nori, A. Criminisi, D. Rueckert, and B. Glocker, "Unsupervised domain adaptation in brain lesion segmentation with adversarial networks," in *Information Processing in Medical Imaging (IPMI)*, June 2017. [Online]. Available: https://www.microsoft.com/en-us/research/publication/unsupervised-domain-adaptation-brain-lesion-segmentation-adversarial-networks/

[9] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference on Learning Representations*, vol. abs/1412.6980, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[10] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[12] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST database of handwritten digits," 1998.

[13] A. M. Mharib, A. R. Ramli, S. Mashohor, and R. B. Mahmood, "Survey on liver ct image segmentation methods," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 83–95, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10462-011-9220-3

[14] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O (1/k2)," in *Soviet Mathematics Doklady*, vol. 27, 1983, pp. 372–376.

[15] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015, http://neuralnetworksanddeeplearning.com.

[16] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[17] C. Payer, D. Štern, H. Bischof, and M. Urschler, "Regressing Heatmaps for Multiple Landmark Localization Using CNNs," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 230–238.

[18] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," in *International Conference on Learning Representations*, 2016.

[19] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 512–519. [Online]. Available: http://dx.doi.org/10.1109/CVPRW.2014.131

[20] 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.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: http://lmb.informatik.uni-freiburg.de//Publications/2015/RFB15a

[22] A. Rozantsev, V. Lepetit, and P. Fua, "On Rendering Synthetic Images for Training an Object Detector," *Computer Vision and Image Understanding*, vol. 137, pp. 24 – 37, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314214002446

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[24] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved Techniques for Training GANs," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf

[25] 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*, vol. 174, no. 1, pp. 71–74, Jan. 2000. [Online]. Available: http://dx.doi.org/10.2214/ajr.174.1.1740071

[26] 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.

[27] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: ACM, 2008, pp. 1096–1103. [Online]. Available: http://doi.acm.org/10.1145/1390156.1390294

[28] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3320–3328. [Online]. Available: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf