Skin Lesion Analysis Towards Melanoma Detection: A Generative Approach

Jan Riedo
Student MSc Biomedical Engineering
Advanced Topics in Machine Learning
University of Bern, Switzerland
jan.riedo@students.unibe.ch

Abstract—Malignant melanoma is an aggressive form of skin cancer. As it is, even for professionals, hard to distinguish from benign seborrheic keratosis, an automated approach is sought-after. The International Skin Imaging Collaboration (ISIC) set up a challenge to classify three different types of skin lesions and provided a small dataset for training. Our approach to this problem is to employ an artificial image generator consisting of a sequence of a variational autoencoder (VAE) and a deep convolutional generative adversarial network (DCGAN). Furthermore, an augmentation strategy for dataset enhancing was applied. For classification, different residual networks were tested. The best accuracy of 66.4% was achieved with a wide residual network (WRN).

Index Terms—skin lesion, variational autoencoder, VAE, deep convolutional generative adversarial network, DCGAN, ResNet, WRN

1 Introduction

Skin cancer (melanoma and non-melanoma) is the most common malignancy in Caucasians [1], [2]. Furthermore, Switzerland is among the states with the highest incidence of malignant melanoma skin cancer in Europe [3]. Non-melanoma skin cancer is malignant too, but unlikely to spread to other parts of the body. Malignant melanoma skin cancer however, is a highly aggressive cancer that tends to spread to other parts of the body and may be fatal if not treated early.

Machine learning (ML) is one way to improve early stage detection, as it can be made accessible to the majority of the global population. ML is used in various fields of medical image processing and analysis with increasing success [4]. The main challenge hereby is the lack of consistent imaging modalities for training data generation. Additionally, most medical image recording processes involve expensive and very sophisticated imaging procedures such as computed tomography (CT) or magnetic resonance imaging (MRI) scanning. An advantage of ML in dermatology is that the data (images) is accessible straightforward with normal photographic equipment. This enables a fast progression of ML in this field with promising results [5]. Furthermore, modern mobile phones are able to take images of sufficient quality to use them for store-and-forward teledermatology or automated on-smartphone diagnosis [6].

One approach to overcome the lack of large, coherent datasets is the augmentation of the original data or the generation of artificial data [7]. Variational autoencoders (VAEs) [8] and generative adversarial networks (GANs) [9], [10] are two sorts of networks which can be applied for this purpose. Both are used in this work and are explained in the proceeding sections. One special form of a GAN is a deep convolutional GAN (DCGAN) which was adapted for this project. Our approach is inspired by a work from apple researchers [11], which showed that GANs can be used as refinement networks for model data.

The International Society for Digital Imaging of the Skin (ISDIS) was founded in 1992 and aims at evolving and promoting new digital skin imaging technologies through its journal "Skin Research and Technology". The International Skin Imaging Collaboration: Melanoma Project (ISIC Project) is one of ISDIS projects to facilitate the application of digital skin imaging to help reduce melanoma mortality. Since 2016, the ISIC Project set up a challenge every year (Skin Lesion Analysis Towards Melanoma Detection), which invites researchers around the world to compete in skin lesion segmentation, attribute detection and lesion classification.

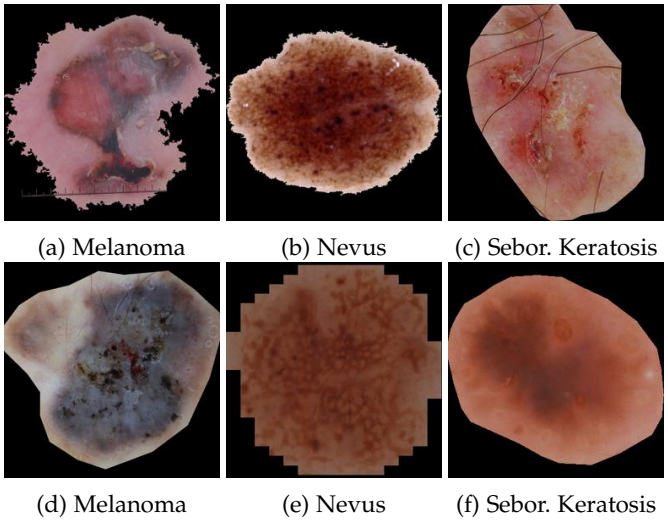
The classification task of the ISIC 2017 challenge [12] targeted at distinguishing images of melanoma, seborrheic keratosis and nevus. Seborrheic keratosis [13] is a non-cancerous benign lesion form, which shows very similar appearance as the dangerous malignant melanoma. Nevus is a nonspecific medical term for a visible, chronic lesion of the skin [14]. The challenge is set up as a two step binary classifier. First a melanoma versus rest classification, second seborrheic keratosis versus rest classification. As we did not actually participate in the challenge, we decided to make a direct classifier for all three classes.

The target of this project is to train a residual network on the skin lesion classification task of the ISIC 2017 challenge and enhance its accuracy by enlarging the rather small dataset. The proposed approach is generating images, similar to the real ones, with a sequence of a VAE and a DCGAN. Furthermore, the impact of additional augmented images (based on the original ones) is assessed.

2 METHODS/INFRASTRUCTURE

2.1 Dataset

The dataset with real skin lesion images was provided by the ISIC archive within the framework of the ISIC 2017 challenge (http://challenge2017.isic-archive.com). The training set consisted of 1700 train images (Fig. 2) with labels (318 melanoma, 1166 nevus, 216 seborrheic keratosis). Another 300 images were used for validation. The test set contained 750 images (147 melanoma, 471 nevus, 132 seborrheic keratosis). Furthermore, segmentation masks were provided for all images, which were applied prior to processing. Moreover, the images of very different resolution and quality were scaled and cropped to 256 x 256 pixels.



(a) Melanoma (b) Nevus (c) Sebor. Keratosis

(d) Melanoma (e) Nevus (f) Sebor. Keratosis

Fig. 2: Scaled and cropped real images with applied segmentation masks. Very inhomogeneous masks with highly different level of border detail. Hairs are a natural artifact spread over all classes.

2.2 Image Generation

A sequence of a variational autoencoder (VAE) (model image generation) and a deep convolutional adversarial network (DCGAN) (refinement) was used for data generation. Each class of skin lesion was trained independently from the others. A schematic of the overall approach is shown in Fig. 1. Models of skin lesions were sampled from the VAE's encoder output (μ_f, σ_f) and refined with the DCGAN.

2.2.1 Variational Autoencoder

A variational autoencoder (VAE) [8] has the same structure as a conventional autoencoder. However, instead of just learning a function representing the data in a compressed form like autoencoders, VAEs learn the parameters of a probability distribution representing the data. In Fig. 1 the VAE is located on the left side with its latent variables $z \sim N(\mu_f, \sigma_f)$.

To achieve the goal set for the VAE of learning the parameters of a multivariate Gaussian distribution, which describes the features of the original images, the objective function is written as follows:

$$\log P(x) - D_{\mathrm{KL}}(Q(z|x)|P(z|x)) \quad (1)$$

$$= E_{z\sim Q}(\log P(x|z)) - D_{\mathrm{KL}}(Q(z|x)|P(z)) \quad (2)$$

Where the encoder 1 output $Q(z|x)$ (Fig. 1) is constrained to follow a multivariate normal distribution $N(z|0,1)$. The posterior $P(z|x)$ is approximated by a simpler distribution $Q(z|x)$ under the Kullback-Leibler divergence measure D_{KL} [15], [16] Eq. 7. D_{KL} represents a penalty for the encoder 1 in case the latent variables z don't follow a multivariate Gaussian distribution.

For the purpose of not only having the ability of sampling from the latent variables in order to create model images, a Gaussian criterion Eq. 6 is applied directly on the pixel space, represented by the distribution μ_p, σ_p. This ensures that the output of the decoder 1 (Fig. 1) is represented as pixel mean μ_p and pixel standard deviation σ_p.

A batchsize of 5 was employed for training. Adam optimizer [17] was applied with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and a learning rate of 10^{-4}. A maximum of 400 epochs were used for training. To prevent overfitting, a regularization in form of an early stopping [18] has been implemented. The termination condition was set to no change in loss over a period of 5 epochs. The default weight initialization was used.

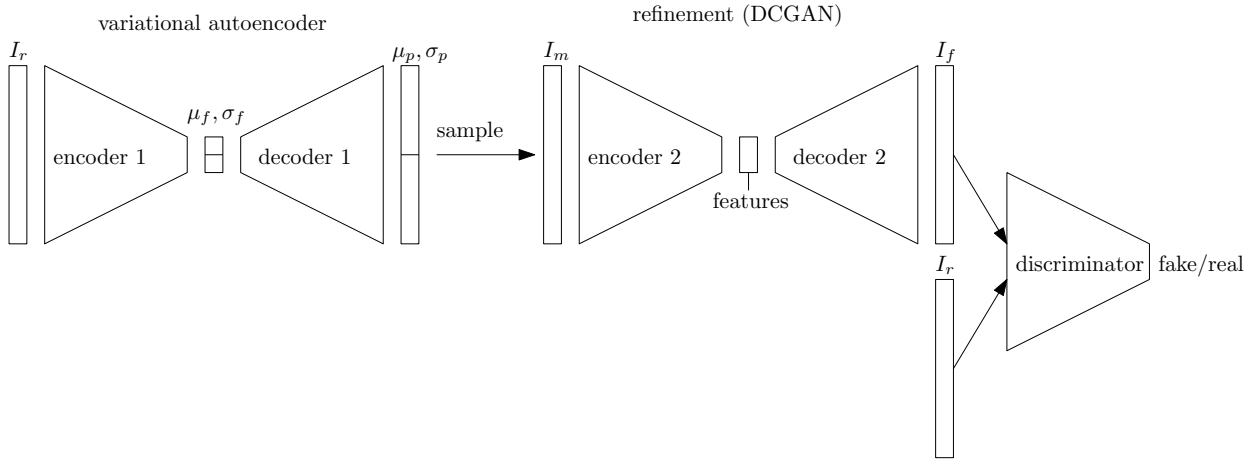


Fig. 1: Overview of whole image generation setup. I_r: real images, I_f: fake images, I_m: model images, μ_f: feature mean, σ_f: feature standard deviation, μ_p: pixel mean, σ_p: pixel standard deviation.

2.2.2 Deep Convolutional Generative Adversarial Network

A Deep Convolutional Generative Adversarial Network (DCGAN) consists of two different networks, a generator G and a discriminator D [9]. In Fig. 1 decoder 2 represents the G network and discriminator the D network. Instead of sampling from a random variable, as proposed in the original paper [19], we employ the encoder 2 in order to get features out of the model images generated by the VAE. In this paper we refer to the encoder 2 - decoder 2 combination as G, rather than only the decoder 2.

The generator will try to produce fake images as real as possible in order to make the discriminator incapable of distinguishing fake from real images. However, the goal of the discriminator is to label all input images correctly as real or fake. The crucial part of training a GAN is the balance between training D and G. Usually, D converges much faster, which disables the generator from further progress. This two-player minimax game can be described as follows with the value function $V(G, D)$:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \quad (3)$$

A batchsize of 32 was employed for all training processes. Learning rates for G and D were set equally to 10^{-4}. Different learning rates (lower for D in order to stop it from too fast convergence) lead to a very unstable behavior. Adam optimizer [17] was applied with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. A first weight initialization was set to a standard normal distribution with zero mean and a standard deviation of 0.04. Further initialization processes are described at the end of this subsection.

The discriminator training strategy (see Fig. 3) consists of two parts; First, real images are given as input to the discriminator and the output is compared to the labels set to one, which results in the real loss L_r, calculated as a binary cross entropy (BCE) [20] Eq. 5. Second, model images are fed into encoder 2 and the output of decoder 2 is fed into the discriminator, whose output is compared to the labels set to zero, which computes the fake loss L_f, again in form of a BCE loss Eq. 5. Backpropagation is carried out on the summation of the real and fake loss and D is optimized.

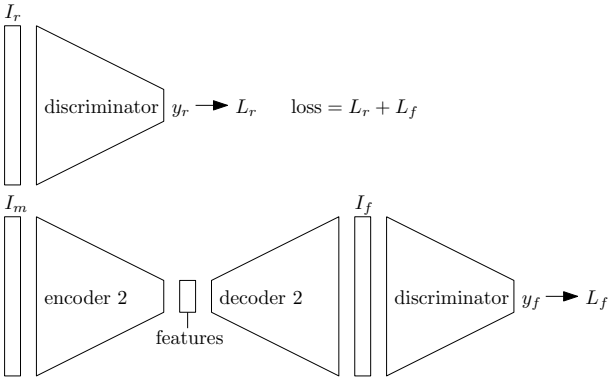


Fig. 3: Scheme of the arrangement for the discriminator training. I_r: real images, I_f: fake images, I_m: model images, y_r: real labels (= 1), y_f: fake labels (= 0), L_r: real loss, L_f: fake loss.

In detail, the discriminator consists of 7 convolution layers, with each, except the first and last with a batch normalization. Additionally, activation functions in form of leaky ReLUs with negative slope 0.2 are applied after each, except for the last one. The last activation function is a sigmoid and the number of output channels is one.

The two sequential nets of the generator, encoder 2 and decoder 2 in Fig. 4 are structured as follows: The encoder 2 is almost identical with the discriminator, except for the feature vector being of size 100. The decoder 2 consists of 7 deconvolution layers with each, except for the last one followed by a batch normalization and a ReLU activation function. The last activation layer is a tanh function and the output size is 3 x 256 x 256. The implementation of the encoder 2 is based on a DCGAN implementation [1] made for celebrity image generation.

The generator training (see Fig. 4), similar to the discriminator training consists of two steps; First, the model images are fed into encoder 2 and the output of decoder 2 is fed into the discriminator, whose output is compared to the labels set to one, which computes the fake loss L_f in form of a BCE loss Eq. 5. Second, encoder 2 and decoder 2 are employed as autoencoder, to assure the similarity of the fake images to the real ones. The input are the model images, and the output is compared to the real images, giving the autoencoder loss L_a, calculated as mean squared error (MSE) Eq. 4. The autoencoder loss is taken into account with a factor γ set to 0.8 and the fake loss with factor (1-γ). Backpropagation is carried out on this summation of losses in order to optimize the encoder 2 and decoder 2.

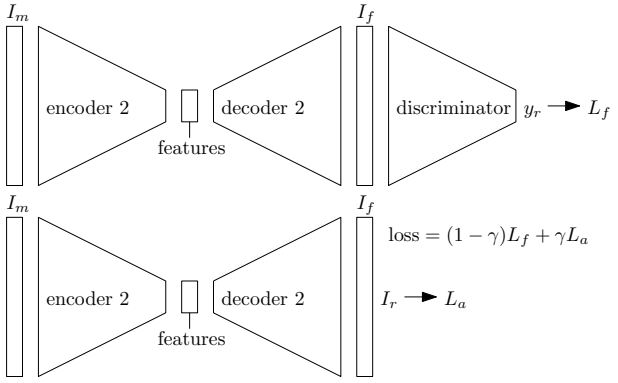


Fig. 4: Scheme of the arrangement for the generator training. I_r: real images, I_f: fake images, I_m: model images, y_r: real labels (= 1), L_f: fake loss, L_a: autoencoder loss, γ: factor for autoencoder loss (= 0.8).

The overall training strategy consists of a weight initialization and a real training part. For weight initialization, the DCGAN network is trained on real images for 8'000 epochs. For each epoch, G is trained four times, while D only once, in order to better balance out the discriminator and generator loss. Furthermore, G was pretrained for 2'000 epochs on the real images. As the model images are similar to the real ones, a total of 2'000 epochs with the model images (again, G was trained 4 times per epoch) was sufficient enough to obtain

1. https://github.com/znxlwm/pytorch-MNIST-CelebA-GAN-DCGAN

realistic images. The success of the strategy was evaluated by qualitative visual assessment of the output images, rather than quantitatively by the loss values.

2.3 Image augmentation

A simple image augmentation was carried out to enlarge the training dataset size. Three different augmentation strategies were applied on all training data, thus quadrupling the dataset size. The three following strategies consist of one or more sequentially applied operations:

- Gaussian-distributed additive noise with $\mu = 0$ and $\sigma^2 \leq 0.01$
- Random zoom, crop and rotate; Zoom factor k: $1.1 \leq k \leq 1.6$, cropping to original image size. Random rotation with maximum angle $0 \leq \alpha \leq 359$, dimensions are preserved.
- Gaussian filter (blur) and horizontal flip; Gaussian filter kernel size 3, normalized with $\sigma \leq 0.2$

2.4 Image Classification

The image classification itself is carried out with different, non-pretrained residual networks. Five deep residual networks (ResNet) [21] with different numbers of blocks (see Fig. 5) are trained (14, 18, 34, 50, 101). The ResNet architecture solves the problem of vanishing gradients and it can be seen as an ensemble of less deep networks [22]. Additionally, two wide residual networks [23] (WRN) with 16 blocks are trained. One with a widen factor of one and the other with widen factor of three. The WRN may be two times faster and improve performance compared to the original ResNet.
All ResNets have a similar structure, with the main difference being the number of blocks used. For the smaller three networks with 14, 18 and 34 blocks, basic building blocks are used as visualized in Fig. 5 on the left. For the deeper two networks with 50 and 101 blocks, the "bottlenecks" visualized on the right in Fig. 5 are implemented. The input and output are the same for all sizes of networks, with the input consisting of a convolution, batch norm, ReLU and a max pooling layer. The output consists of an average pooling layer and a linear layer with three output channels corresponding to the three classes of skin lesions.
The WRN architecture is based on the one of the ResNet with different sequence within the building blocks, which is batch normalization, ReLU and convolution.

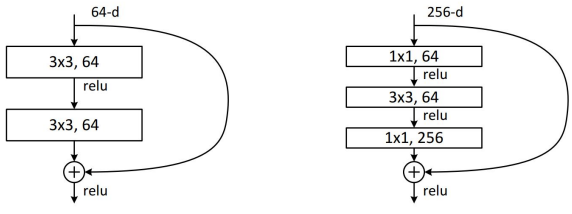


Fig. 5: Main components of ResNet [21]; Left: Basic building block, Right: "bottleneck" building block [24].

A batchsize of 30, and 100 epochs were applied to train all networks. The Adam optimizer [17] was used with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a learning rate of 10^{-4}. Additionally, for all ResNets, a different version of the momentum update has been tested. The Nesterov momentum [25] (Nadam) is known to work slightly better than the standard momentum.

2.5 Performance Measures

Mean squared error (MSE) loss Eq. 4 can be used to compare two images. It was used for generator training of the DCGAN.

$$MSE(x, y) = \frac{1}{N}\sum_{n=0}^{N}(x_n - y_n)^2 \tag{4}$$

Binary cross entropy (BCE) loss Eq. 5 is used to compare an output x_n between zero and one with labels y_n in the same range. It was used for generator and discriminator training of the DCGAN.

$$BCE(x, y) = \frac{1}{N}\sum_{n=0}^{N} y_n \log x_n + (1 - y_n)\log(1 - x_n)[20] \tag{5}$$

The VAE training required two different loss functions: A Gaussian criterion for reconstruction loss calculation Eq. 6, and the Kullback-Leibler divergence measure Eq. 7.

$$RL = \frac{1}{2}\log\sigma^2 + 0.5(2\pi) + \frac{0.5(x - \mu)^2}{\sigma^2} \tag{6}$$

$$D_{\mathrm{KL}}(\mu, \sigma^2) = -\frac{1}{2}\sum(1 + \log\sigma^2 - \mu^2 - \sigma^2)[15][16] \tag{7}$$

The classifier training was carried out with a cross-entropy loss function which is used to compare the output x as probabilities for each class with the ground truth c of which class it actually is.

$$CE(x, c) = -x[c] + \log(\sum_{j} e^{x[j]}) \tag{8}$$

As measure for classification performance, the top-1 accuracy is reported. To gain a deeper insight towards possible misclassifications, the confusion matrix is computed and visualized.

2.6 Infrastructure

The code is implemented in Python 3.6[2] with PyTorch v0.5.0 for windows [26] using CUDA for GPU access and scikit-image for image augmentation [27]. Computations were carried out either on a Linux system with a NVIDIA Tesla K20c GPU with 5 GB memory or on a Windows system with a NVIDIA GTX 1080Ti Aero with 11 GB memory.

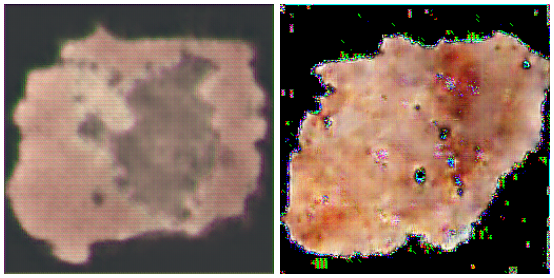
3 Results

3.1 Image Generation

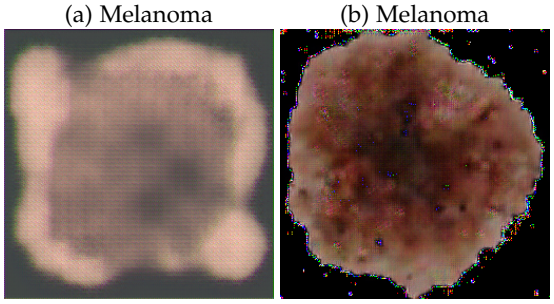
A total of 2016 images (384 melanoma, 1166 nevus, 216 seborrheic keratosis) were generated (see Fig. 6), each class proportional to the original images available in this class. This resulted in a total training set size for original and generated images of 3716. The quality of the DCGAN output-images did not strongly correlate to the computed losses.

2. Python Software Foundation. Python Language Reference, version 3.6. Available at http://www.python.org

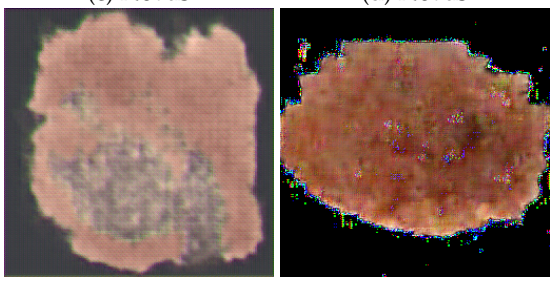
Every 100 epochs, a freshly generated image was saved to inspect it visually and assess its quality. Termination of training was therefore not fixed by epochs, rather manually, when the quality of the images was satisfactory. However, the losses were used to determine if a network (G or D) was still learning and if the balance between G and D is kept.



(a) Melanoma (b) Melanoma



(c) Nevus (d) Nevus



(e) Sebor. Keratosis (f) Sebor. Keratosis

Fig. 6: Left: Model images generated with the VAE. Right: Generated images after refinement with DCGAN

3.2 Image augmentation

A total of 5100 images were added to the training set, resulting in a training set size of 6800 images. The augmentation applied was carried out with rather weak perturbations, leading to very similar images compared to the original dataset. The biggest change was induced by the zooming operation.

3.3 Image Classification

The highest accuracy of 66.4% on the test set was achieved with the WRN (16 blocks, widen factor 3), trained on the original and augmented images (see Fig. 7). The WRN-3-16 network achieves the highest scores for all three training sets. The accuracies obtained are: 66.0% when trained on the original and generated, and 65.5% when trained only on the original images. The best performance of 65.5% with a ResNet is trained on the original and generated data with 14 building blocks.

Fig. 7: Comparison of accuracy of the different networks, trained on the original images, the original and the augmented images, and the original and generated images.

The confusion matrices for overall top accuracy and best accuracy on a ResNet are visualized in Fig. 8 and 9 respectively. Both matrices show a similar pattern with the highest true positive score for nevus, followed by seborrheic keratosis and melanoma. Overall, all classifiers predicted for most skin lesions to be nevus, which is well visible in the middle column of the confusion matrices.

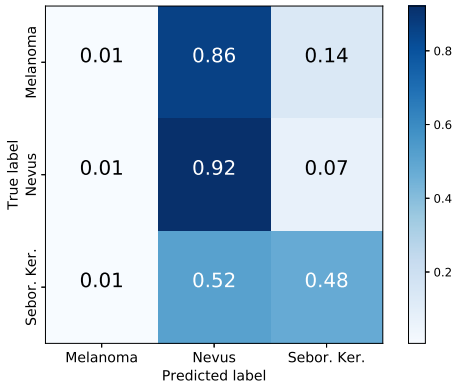


Fig. 8: Confusion matrix of WRN-3-16 trained on the original and augmented images. This setting achieved the highest overall performance.

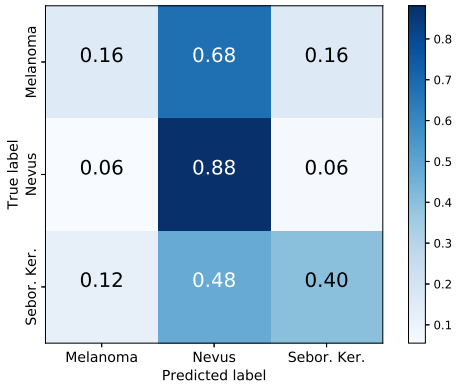


Fig. 9: Confusion matrix of ResNet-14 trained on the original and generated images. Highest performance of all ResNet tested in this work.

4 Discussion

All four network types (VAE, DCGAN, ResNet, WRN) showed satisfactory results. Both dataset enlarging strategies (generation and augmentation) were successful and lead to an increase in accuracy. The overall top accuracy (achieved with a WRN) was enhanced by 0.9%. The accuracy of the ResNets was improved by 2.3%.

The sampled images from the variational autoencoder captured a large variety of structures and appearances. Different latent feature space sizes were tested, with the best being 64. Early stopping was crucial to prevent the algorithm from producing very smoothed out images with no "black spots" on the actual skin lesion.

The main challenge of DCGAN training was, as is known, the balance between generator and discriminator training. Three measures were taken to face it. First; The generator was trained four times per epoch, whereas the discriminator only once. This implementation had the highest impact of all actions. Second; Generator pretraining as autoencoder for 2000 epochs was crucial as well. Without it, even after the first epoch, the discriminator had smaller losses of magnitude 10 and converged far to fast for the generator to react. Third; The use of the generator as autoencoder and constructing the total loss of 80% autoencoder loss and 20% actual DCGAN loss lead to more realistic images. Related to this was the manual termination criterion, where the image quality was assessed visually rather than concluding it from the loss.

Other elements were tested, but did not evoke the desired reaction. Different learning rates were implemented (a smaller learning rate for D than G), but lead to a very unstable training behavior with bad results. Additionally, tests were made where per epoch, only the network with the higher loss was updated. This resulted in only the generator being trained, but with no actual progress.

The artifacts in Fig. 6b, d, f were addressed with a larger feature space (between encoder 2 and decoder 2 in Fig. 4). Unfortunately, the quality of the fake images did not improve with this change. A brute force method would be to median filter the image as a postprocessing step, but the network should be able tho handle this itself.

Both dataset enlargement strategies lead to an increase in classifier accuracy. Although it is only a slight rise (0.9% on WRN and 2.3% on ResNet), it proves, that additional artificially generated and augmented data can help to improve a classifier. The Nadam optimizer did not have a significant better performance than te standard Adam optimizer.

The confusion matrices revealed, that the accuracies achieved are strongly based on the unbalanced presence of classes in the dataset. Nevus is the most abundant skin lesion type in the train as well as in the test set. This lead to good performance of the classifier on the nevus class (overfitting), but to a poor one on the other two. With respect to this, the ResNet-14 classifier (trained on the original and generated images) did perform better, with a better generalization than the overall best net WRN-3-16 (trained on the original and augmented dataset).

The best score of this challenge published [28] is not comparable to our results, as they approached the task with a two-step binary classifier.

5 Conclusion

We were able to show that dataset augmentation and artificial data generation can improve the accuracy of a classifier on a given (small) dataset. Our approach with a sequence of model-generating VAE and refining it with a DCGAN seems feasible. Further research should focus on removing image artifacts coming from the DCGAN and investigating in the impact on the classifier accuracy of the amount and distribution of artificial data used for training.

Acknowledgements

The author gratefully acknowledges the contribution made to this work by Michael Müller and Elias Rüfenacht, who implemented the image classifier, and Matthias Fontanellaz, who implemented the variational autoencoder.

References

[1] D. C. Whiteman, A. C. Green, and C. M. Olsen, "The growing burden of invasive melanoma: Projections of incidence rates and numbers of new cases in six susceptible populations through 2031," *Journal of Investigative Dermatology*, vol. 136, no. 6, pp. 1161–1171, jun 2016.

[2] E. de Vries and J. W. Coebergh, "Cutaneous malignant melanoma in europe," *European Journal of Cancer*, vol. 40, no. 16, pp. 2355–2366, nov 2004.

[3] E. Union, "Ecis - european cancer information system," 2018. [Online]. Available: https://ecis.jrc.ec.europa.eu

[4] M. de Bruijne, "Machine learning approaches in medical image analysis: From detection to diagnosis," *Medical Image Analysis*, vol. 33, pp. 94–97, oct 2016.

[5] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, jan 2017.

[6] C. Rat, S. Hild, J. R. Sérandour, A. Gaultier, G. Quereux, B. Dreno, and J.-M. Nguyen, "Use of smartphones for early detection of melanoma: Systematic review," *Journal of Medical Internet Research*, vol. 20, no. 4, p. e135, apr 2018.

[7] L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning."

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *The International Conference on Learning Representations (ICLR)*.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv*.

[10] Y. Hong, U. Hwang, J. Yoo, and S. Yoon, "How generative adversarial nets and its variants work: An overview of gan," *CoRR*, vol. abs/1711.05914, 2017.

[11] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training."

[12] N. C. F. Codella, D. Gutman, M. E. Celebi, B. Helba, M. A. Marchetti, S. W. Dusza, A. Kalloo, K. Liopyris, N. Mishra, H. Kittler, and A. Halpern, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic)."

[13] C. Hafner and T. Vogt, "Seborrheic keratosis," *Journal der Deutschen Dermatologischen Gesellschaft*, vol. 6, no. 8, pp. 664–677, aug 2008.

[14] R. Happle, "What is a nevus," *Dermatology*, vol. 191, no. 1, pp. 1–5, 1995.

[15] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, mar 1951.

[16] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2007*. IEEE, apr 2007.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[18] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, apr 2007.
[19] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," *ArXiv e-prints*, Nov. 2015.
[20] D. P. K. Reuven Y. Rubinstein, *The Cross-Entropy Method*. Springer-Verlag New York Inc., 2004.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778.
[22] A. Veit, M. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks."
[23] S. Zagoruyko and N. Komodakis, "Wide residual networks."
[24] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *Procedings of the British Machine Vision Conference 2011*. British Machine Vision Association, 2011.
[25] A. Botev, G. Lever, and D. Barber, "Nesterov's accelerated gradient and momentum as approximations to regularised update descent."
[26] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.
[27] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, jun 2014.
[28] K. Matsunaga, A. Hamada, A. Minagawa, and H. Koga, "Image classification of melanoma, nevus and seborrheic keratosis by deep neural network ensemble."