Augmenting Histological Images with Adversarial Attacks

Nikita Lockshin[1], Alexander Khvostikov[1] and Andrey Krylov[1]

[1]Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University, Leninskie Gory, 1, building 52, Moscow, 119991, Russia

Abstract

Neural networks have shown to be vulnerable against adversarial attacks - images with carefully crafted adversarial noise that is imperceptible to the human eye. In medical imaging tasks this can be a major threat for making predictions based on deep neural network solutions. In this paper we propose a pipeline for augmenting a small histological image dataset using State-of-the-Art data generation methods and demonstrate an increase in accuracy of a neural classifier trained on the augmented dataset when faced with adversarial images. When trained on the non-augmented dataset, the neural network achieves an accuracy of 55.24 on the test set with added adversarial noise, and an accuracy of 97.40 on the same test set when trained on the augmented dataset.

Keywords

Adversarial Attacks, Deep Learning, Image Classification, Histology, Tissue Recognition.

1. Introduction

Some machine learning models, deep neural networks in particular, have been shown to be vulnerable to *adversarial attacks*, which means they make incorrect predictions after adding an imperceptible noise to the input image [1, 2, 3, 4] (Fig. 1). Currently, many adversarial attack and defense methods have been developed [1, 5, 6]. Most adversarial defense methods either make modifications to the model, for example, defensive distillation [6], or make assumptions about possible attacks [5].

Currently, one of the most effective adversarial attack methods is *AdvGAN* [7]. This method has placed first on the MNIST Adversarial Examples Challenge. The main advantages of AdvGAN are high attack effectiveness and the small amplitude of the generated noise. This method is based on the Generative Adversarial Network framework [8], in which the generator is trained to produce adversarial noise for an input image.

Currently, some medical imaging tasks, such as histological image classification, are solved using neural networks [9], which are vulnerable to adversarial attacks. Hence, making predictions using neural networks in medical imaging can be dangerous. In this paper, we demonstrate the effectiveness of AdvGAN against a neural histological image classifier, and propose a pipeline for augmentating the train dataset in order to make the classifier robust to attacks of this type.

GraphiCon 2022: 32nd International Conference on Computer Graphics and Vision, September 19–22, 2022, Ryazan State Radio Engineering University named after V.F. Utkin, Ryazan, Russia

lockshin1999@mail.ru (N. Lockshin); khvostikov@cs.msu.ru (A. Khvostikov); kryl@cs.msu.ru (A. Krylov)

0000-0001-7777-7035 (N. Lockshin); 0000-0002-4217-7141 (A. Khvostikov); 0000-0001-9910-4501 (A. Krylov)

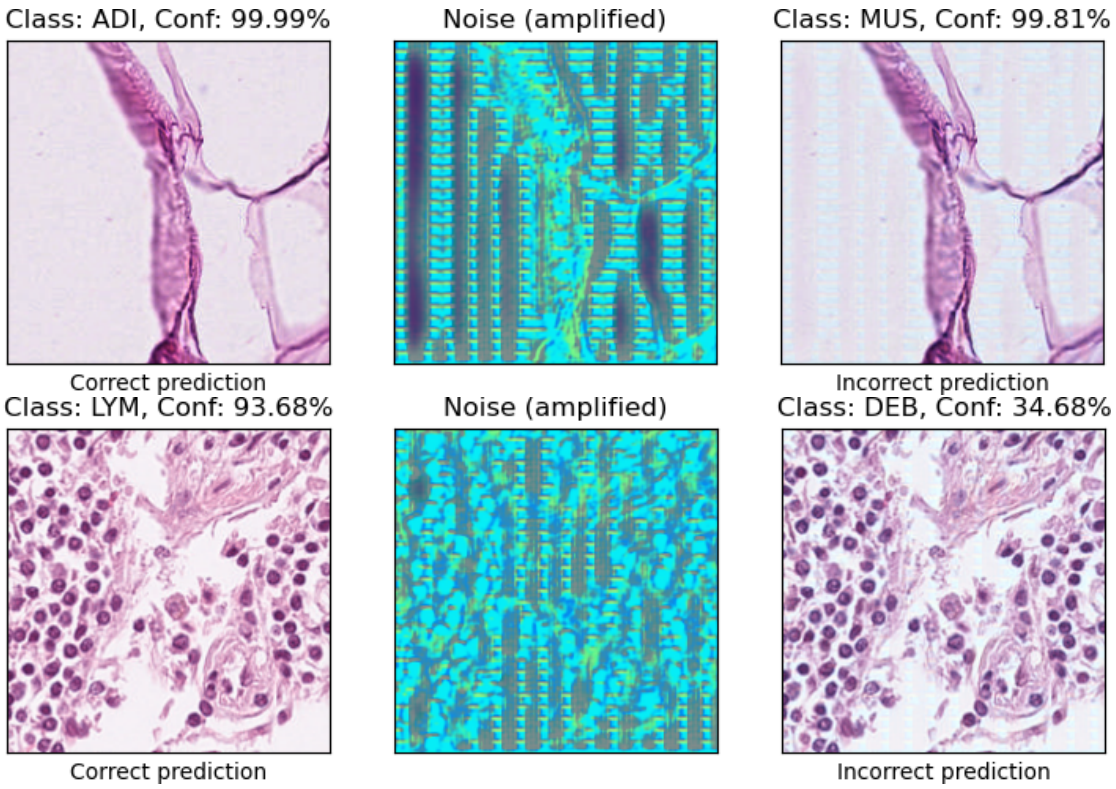

Figure 1: Examples of successful adversarial attacks generated by AdvGAN

2. Used data and task formulation

In this work we use a balanced subset of the *NCT-CRC-HE-100K* dataset [10], with the subset consisting of 22500 labeled histological images. The choice to use a small subset was made in order to decrease time required to train the proposed pipeline. The images are non-overlapping patches from hematoxylin & eosin (H&E) stained histological images of human colorectal cancer and normal tissue.

Each image has a resolution of 224×224 and is assigned one out of 9 classes: *adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR), colorectal adenocarcinoma epithelium (TUM).* These images were manually extracted from 86 H&E stained human cancer tissue slides from formalin-fixed paraffin-embedded (FFPE) samples from the NCT Biobank (National Center for Tumor Diseases, Heidelberg, Germany) and the UMM pathology archive (University Medical Center Mannheim, Mannheim, Germany). Tissue samples contained CRC primary tumor slides and tumor tissue from CRC liver metastases; normal tissue classes were augmented with non-tumorous regions from gastrectomy specimen to increase variability. Examples of images for each class are shown in Fig. 2.

The subset is split into 18000 train images and 4500 test images. The goal of this work is to

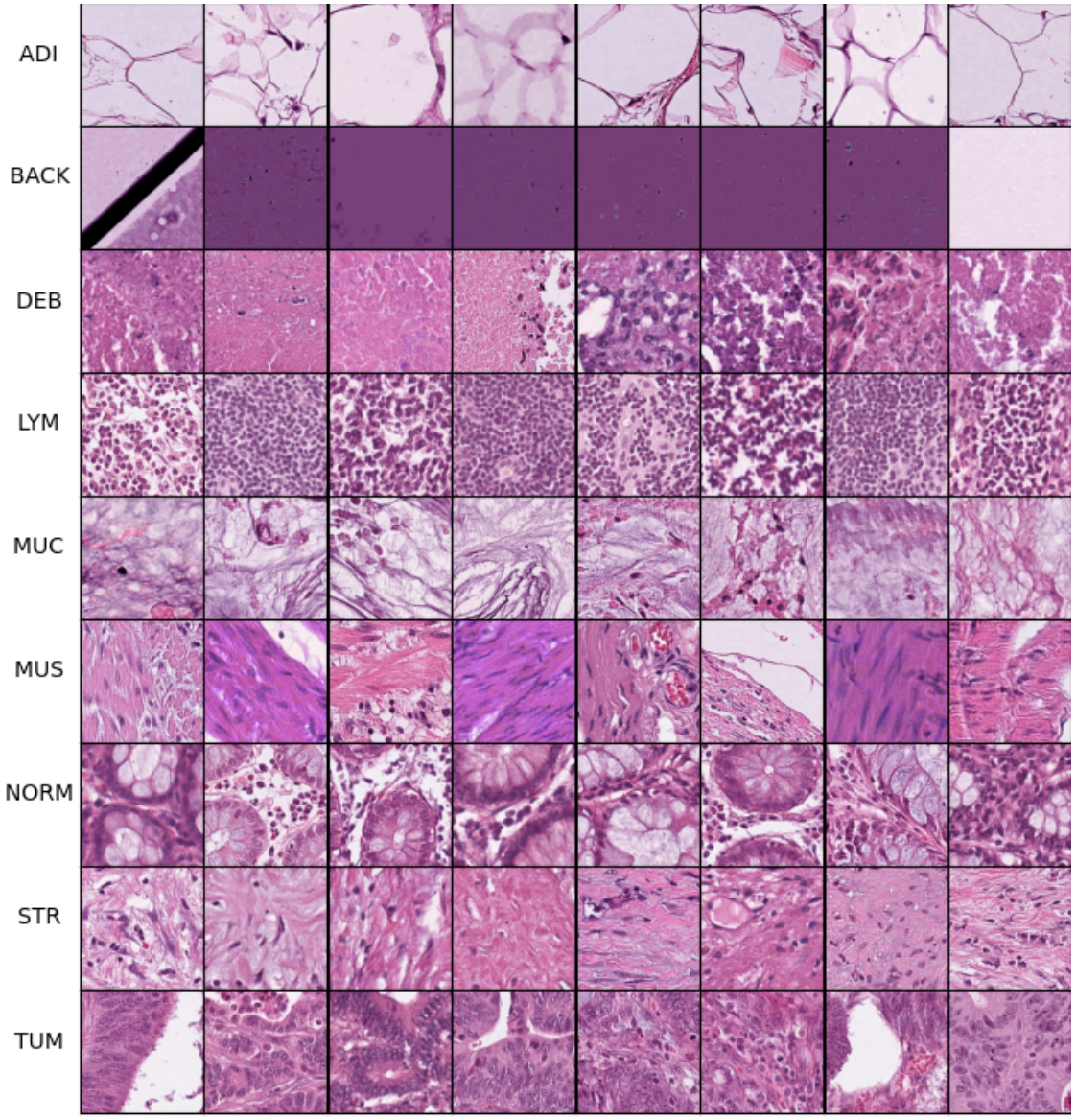

Figure 2: Examples of images for each class in the used subset of the NCT-CRC-HE-100K dataset [10]

design and implement an augmentation method using adversarial attacks in order to increase the robustness of a neural classifier trained on the augmented dataset against adversarial attacks generated by AdvGAN.

3. Proposed methodology

In this paper we propose a new augmentation method designed for histological image classification consisting of several steps:

1. Performing augmentation by generating new examples using the well known StyleGAN2 architecture [11].
2. Applying superresolution technique in order to scale the generated images from 128×128 to 256×256 using the *SRGAN* method [12].
3. Performing augmentation of the resulting dataset using AdvGAN [7] to generate adversarial examples.

3.1. Augmentation using StyleGAN2

StyleGAN2 is a well known method used for data generation from a learned joint distribution $P(X, y)$, where X is an object in the data, in our case it is a histological image of size $128 \times 128 \times 3$, and y is a class label. The image size of 128 was chosen in accordance with the StyleGAN2 architecture. After training the StyleGAN2 generator accepts a random vector $z \sim \mathcal{N}_{512}(0, 1)$ and returns a synthetic image of size $128 \times 128 \times 3$. The decrease of image resolution from 224 to 128 was done to decrease the training time necessary for the method to achieve satisfying quality, as well as memory consumption.

During training, the loss function takes the form of the *Vanilla GAN Loss* [8], given by the following equation:

$$l_{GAN} = \mathbb{E}_{x \sim p_{data}(x)} \log D(x) + \mathbb{E}_{z \sim \mathcal{N}_{512}(0,1)} \log(1 - D(G(z))), \tag{1}$$

where D is the discriminator in the GAN framework, G is the generator, x is a data sample (in our case - a histological image of size $128 \times 128 \times 3$), z is a random vector of length 512 sampled from a normal distribution with mean 0 and variance 1. The generator G aims to minimize the loss, whereas the discriminator D aims to maximize it.

To measure the quality of the generated images we use the *Fréchet Inception Distance (FID)* [13], given by

$$d^2((m, C), (m_w, C_w)) = \|m - m_w\|_2^2 + Tr(C + C_w - 2(CC_w)^{1/2}), \tag{2}$$

where m, m_w are the means of the learned and ground truth distributions respectively, and C, C_w are the respective covariance matrices.

The FID metric acts as a distance between distributions. To calculate m, m_w, C and C_w we use the outputs of the last linear layer of the $Inception - V3$ [14] neural network pretrained on the Imagenet dataset, with real and synthetic images given as input. The output of the linear layer is a vector of length 2048.

3.2. Super-resolution

Currently, one of the best and most widely used methods for single image super-resolution is *Super-resolution using Generative adversarial networks (SRGAN)* [12]. SRGAN is a GAN in which the generator attempts to upscale a single image passed as input.

The quality of the superresolution during training is controlled by a linear combination of the following equations:

$$l = \alpha l^{SR}_{VGG_i} + \beta l^{SR}_{Gen}, \tag{3}$$

$$l^{SR}_{VGG_i} = \frac{1}{W_i H_i} \sum_{x=1}^{W_i} \sum_{y=1}^{H_i} (\phi_i(I^{HR})_{x,y} - \phi_i(G(I^{LR}))_{x,y})^2, \tag{4}$$

$$l^{SR}_{GAN} = \sum_{n=1}^{N} -\log D(G(I^{LR})), \tag{5}$$

where I^{HR}, I^{LR}, I^{SR} are, respectively, the image from the train partition of the dataset of size 224×224 scaled to 256×256, the same image of size 224×224 downscaled to 128×128, the result generated from the downscaled image by SRGAN of size 256×256, W_i, H_i - the width and the height of the feature map ϕ_i of a pretrained *VGG-19* [15] network, N is the batch size. The discriminator D during training attempts to maximize the left part of (1).

To clarify our choice of image sizes, we followed an available implementation of SRGAN in which the scaling factor r was set to 2. In accordance with the StyleGAN2 architecture, the downscaled image resolution was chosen as 128×128, and then upscaled by the factor r. After generation of synthetic 256×256 images, each image is downscaled to 224×224 and added to the full dataset. This augmentation method saved an enormous amount of time, since SRGAN did not require a lot of training time, and no code modification to the original SRGAN had been done.

After training, we use SRGAN to scale the small images generated by StyleGAN2 of size 128×128 to 256×256.

3.3. Generating adversarial attacks using AdvGAN

AdvGAN is a GAN in which the generator accepts an image of size $224 \times 224 \times 3$ as input, and produces a perturbation of the same size, such that when added to the input image, would cause a target neural network classifier to misclassify the resulting image. The architecture of this method is shown in Fig. 3.

Here, *Threshold Loss* is given by:

$$l_{threshold} = \sum_{x=1}^{W} \sum_{y=1}^{H} \sum_{z=1}^{C} \max(|I_{x,y,z}| - thr, 0)^2, \tag{6}$$

where I is an image of size $H \times W \times C$ passed as input to the generator G, thr is the threshold for the maximum absolute value of the adversarial noise. In our work $H = W = 224$, $C = 3$.

The loss functional *Vanilla GAN loss* is given by (1). *Negated Cross Entropy Loss* is defined as:

$$l_{nce} = \frac{1}{l_{ce}}, \tag{7}$$

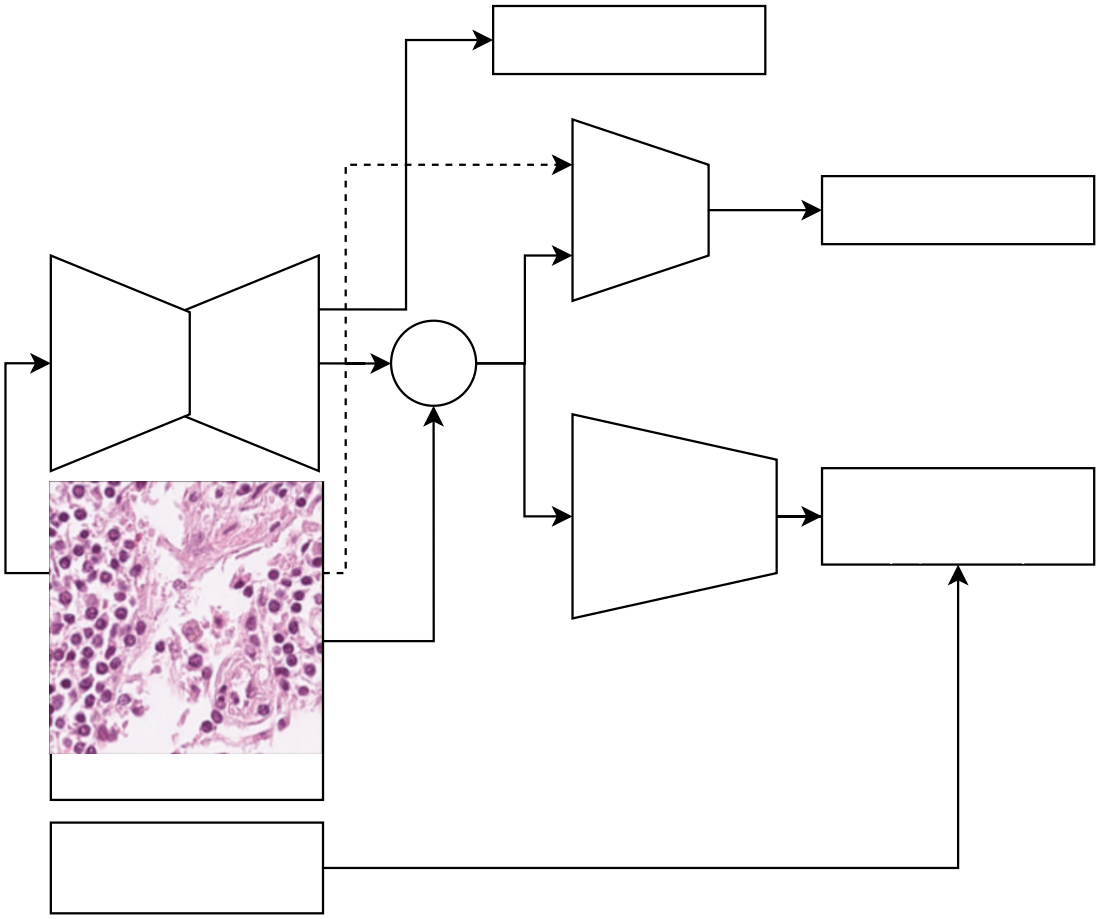

Figure 3: AdvGAN architecture. G is the adversarial noise generator which accepts an input image and returns adversarial noise for that image, D is the discriminator in the GAN framework

$$l_{ce} = -\frac{1}{N}\sum_{n=1}^{N}\log\frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C}\exp(x_{n,c})}, \tag{8}$$

where x_n is the output vector of size C, returned by the neural classifier and corresponding to the image with index n in a batch of size N. y_n is the class index of the image with index n. In summary, the overall loss functional is defined as:

$$l = \gamma l_{nce} + \theta l_{GAN} + \zeta l_{threshold}, \tag{9}$$

where γ, θ, ζ are the hyperparameters that control the importance of each separate function.

4. Testing and results

4.1. StyleGan2 generation results

In our work, we trained StyleGAN2 for 280000 batch iterations, with each batch containing 32 randomly chosen images of the train partition of the dataset. Training was done on a single NVIDIA RTX A6000 GPU and took approximately 7 days and 21 hours. At the end of the training session, the calculated FID (given by (2)) between the generated images and the train partition of the dataset was equal to 16.77. Examples of generated images are shown in Fig. 4.

After completing the training procedure, 18000 synthetic images of size $128 \times 128 \times 3$ were generated.

4.2. SRGAN generation results

In our work, we have trained SRGAN for 6000 batch iterations, with each batch containing 32 randomly chosen images from the train partition.Training was done on a single NVIDIA RTX A6000 GPU and took approximately 21 hours. After training, the *PSNR* between images in the train partition and the images scaled by SRGAN was equal to 30.7 on average. *SSIM* was equal 0.93 on average. Examples of scaled images can be found in Fig. 5. This result allows to generate high quality synthetic images without spending a lot of time training a high-resolution image generator.

4.3. AdvGAN generation results

We have trained this method on 68000 batch iterations, with each batch containing 16 randomly chosen images from the train partition. In our work, we have set parameters γ, θ and ζ to 10, 1 and 1 respectively. We have tested the resulting adversarial attacks on three neural classifiers, all of which were *ResNet34* [16] networks. The test results are demonstrated in table 1. Each of the classifiers was given its own train dataset. Results for the classifier trained on the vanilla dataset containing 18000 real images are shown in the first row. Results for the classifier trained on the combination of 18000 real images and 18000 fake images are shown in the second row. And the results for the classifier trained on the previous combination with applied AdvGAN adversarial attacks to each image are shown in the third row. Additionally, we have tested the classifiers on the test set with applied FGSM attacks. The attacks were generated using the classifier trained on the initial dataset, as would a potential adversary attempt without any knowledge of synthetic data in the training set. Since the test dataset is balanced, we used the classic accuracy metric which shows the proportion of correctly classified images accross the whole dataset.

As a result of the augmentation pipelane, the dataset size has increased by 4 times using high quality synthetic images. The test accuracy of the neural classifier trained on the resulting dataset for the same amount of epochs has increased, moreover, by adding adversarial images to the dataset, the classifier showed remarkable results on the AdvGAN adversarial attack test set, showing only a 1.18% drop in performance.

The achieved results demonstrate that the proposed augmentation pipeline not only makes the target classifier more robust to various adversarial attacks, but also improves its performance

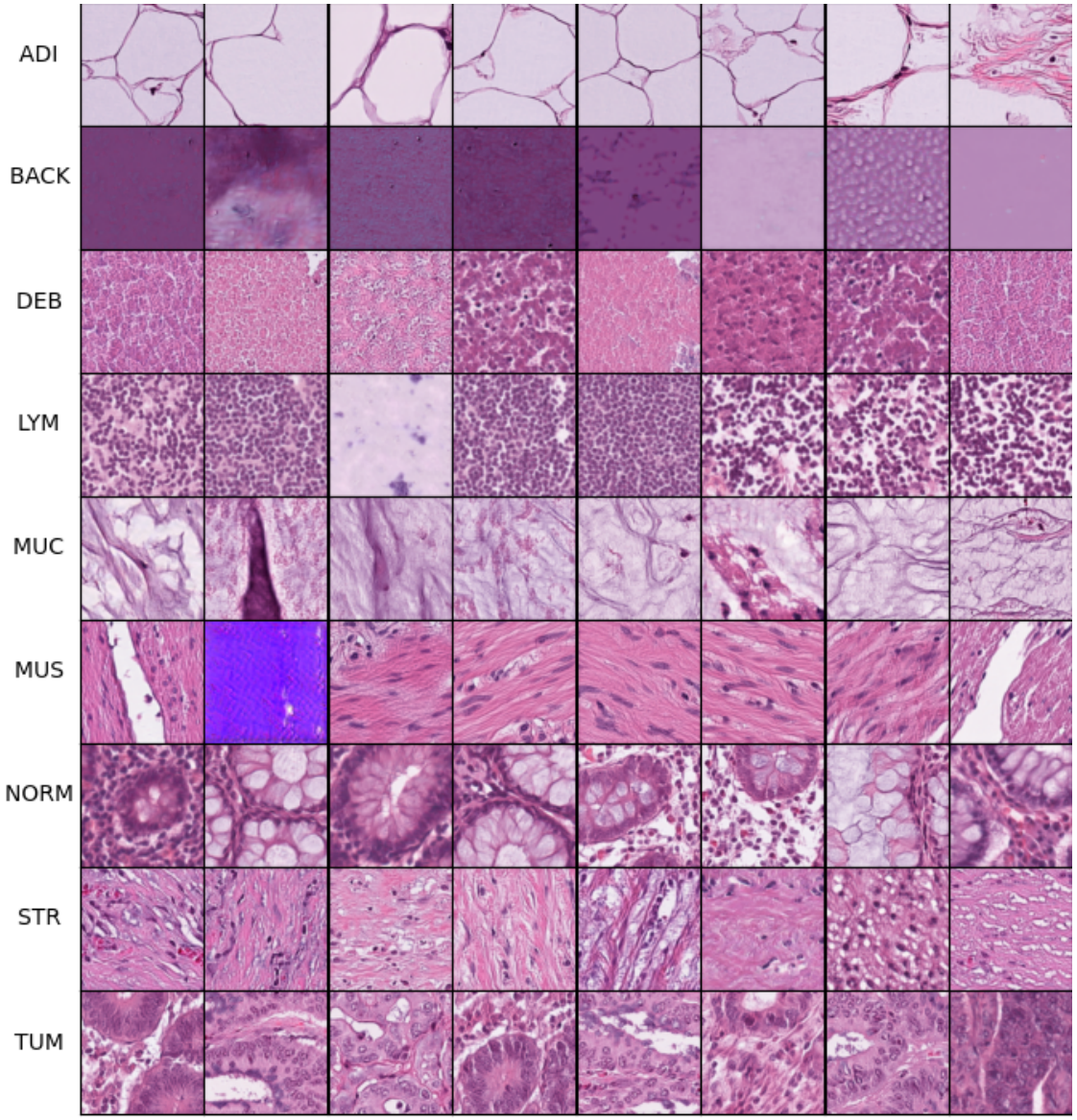

Figure 4: Examples of synthetic images generated by StyleGAN2 for each class (ADI, BACK, DEB, LYM, MUC, MUS, NORM, STR, TUM

in general.

Adversarial attack examples are shown in Fig. 1.

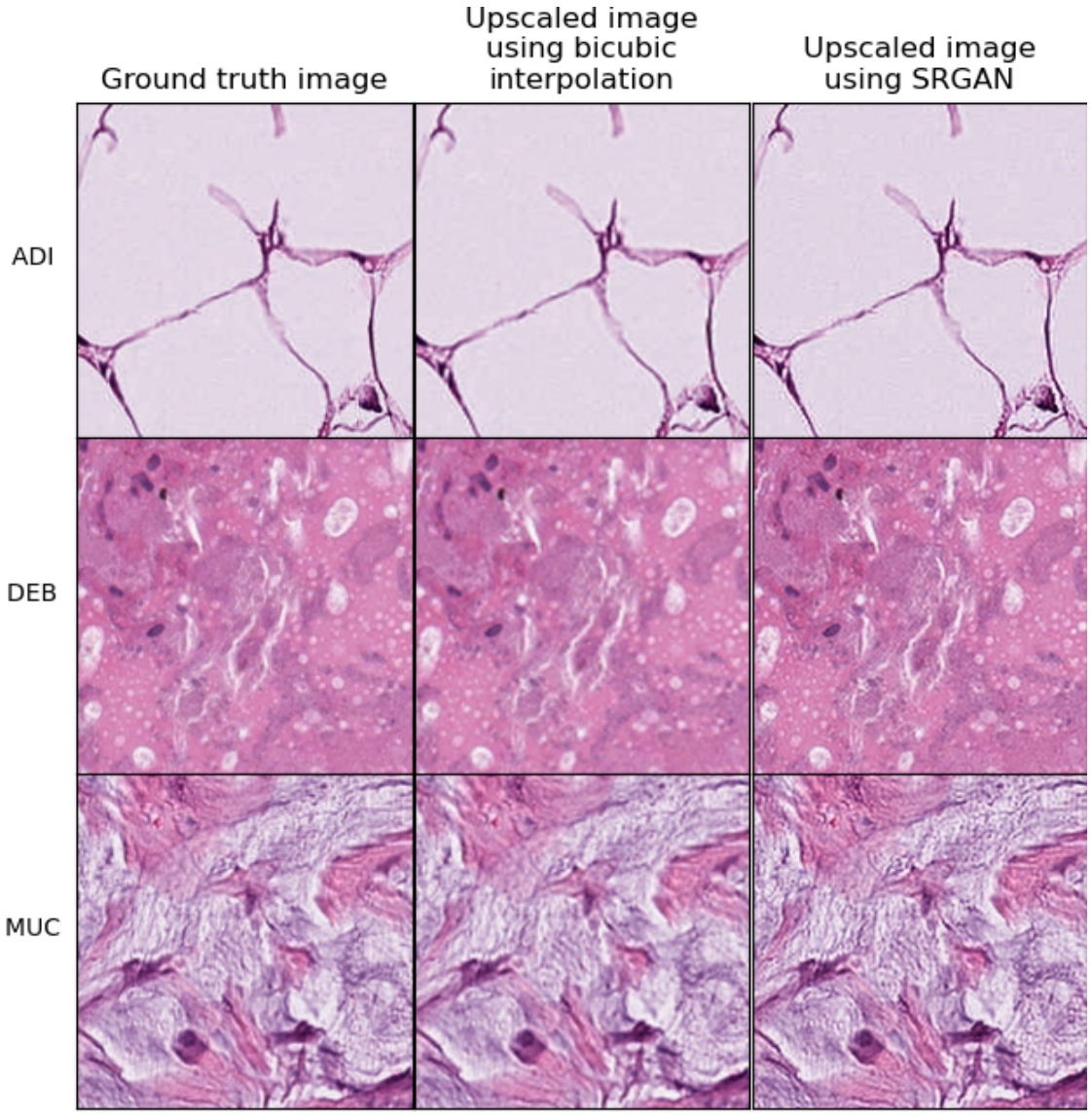

Figure 5: Examples of upscaled images belonging to several classes (ADI, DEB, MUC) using SRGAN and bicubic interpolation

5. Implementation details

All experiments were conducted using the *Python3* programming language. The implementation for the CNN architectures, training and evaluating procedures was done using the open source software library for machine learning *PyTorch*.

Table 1
AdvGAN testing results. The second column shows accuracy on the test dataset with AdvGAN applied to each image, the third columns shows accuracy on the test dataset with FGSM with $\epsilon = 0.1$ applied to each image

Train dataset	Test dataset accuracy, %	AdvGAN, %	FGSM, %
18000 train images	95.67	55.24	68.76
36000 images (18000 train images and 18000 generated by StyleGAN2)	97.56	63.42	83.54
72000 images (36000 previous images with added adversarial attacks)	**98.58**	**97.40**	**90.38**

6. Conclusion

In this paper, we have implemented a pipeline for augmenting a small dataset of histological images with adversarial attacks. We have demonstrated the effectiveness of the proposed pipeline on the available test set. The train dataset has increased by 4 times with high quality synthetic histological images, and the neural classifier trained on the resulting dataset has shown an increase in quality not only on the adversarial test set, but on the test set with no additional noise added to the images. The main drawbacks are the amount of used data and the quality of the adversarial attacks. Since we haven't used the entire available training set (only 18000 out of 100000 labeled images), future research should test the augmentation pipeline on larger amounts of data. Moreover, the AdvGAN adversarial attacks are not perfect, and can easily be spotted by the human eye, as can be seen in Fig. 1. Future research should focus on making the adversarial attacks more imperceptible without sacrificing effectiveness in fooling neural networks.

Acknowledgments

The work was funded by RSCF according to the research project 22-21-00081.

References

[1] I. J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, arXiv preprint arXiv:1412.6572 (2014).
[2] J. Su, D. V. Vargas, K. Sakurai, One pixel attack for fooling deep neural networks, IEEE Transactions on Evolutionary Computation 23 (2019) 828–841.
[3] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 ieee symposium on security and privacy (sp), IEEE, 2017, pp. 39–57.
[4] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.
[5] B. Liang, H. Li, M. Su, X. Li, W. Shi, X. Wang, Detecting adversarial image examples in

deep neural networks with adaptive noise reduction, IEEE Transactions on Dependable and Secure Computing 18 (2018) 72–85.

[6] N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami, Distillation as a defense to adversarial perturbations against deep neural networks, in: 2016 IEEE Symposium on Security and Privacy (SP), IEEE, 2016, pp. 582–597.

[7] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, D. Song, Generating adversarial examples with adversarial networks, arXiv preprint arXiv:1801.02610 (2018).

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[9] A. Khvostikov, A. Krylov, I. Mikhailov, P. Malkov, N. Danilova, Tissue type recognition in whole slide histological images (2021).

[10] J. N. Kather, N. Halama, A. Marx, 100,000 histological images of human colorectal cancer and healthy tissue, Zenodo10 5281 (2018).

[11] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, T. Aila, Analyzing and improving the image quality of stylegan, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 8110–8119.

[12] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[13] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local nash equilibrium, Advances in neural information processing systems 30 (2017).

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

[15] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[16] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.