Photo-to-Drawing Style Transfer

David Mašek

ČVUT - FIT

masekda4@fit.cvut.cz

December 28, 2021

1. Introduction

The goal of the semestral work is to convert photos to drawings (sketches). This type of task is often called style transfer. A popular example of similar work is the conversion of photos to impressionist-style paintings. My goal is to explore feasibility of domain transfer to hand-drawn sketches and train a model which handles the task as well as possible. I was especially interested how well you could transform a photo you might take outside such as landscape or city street.

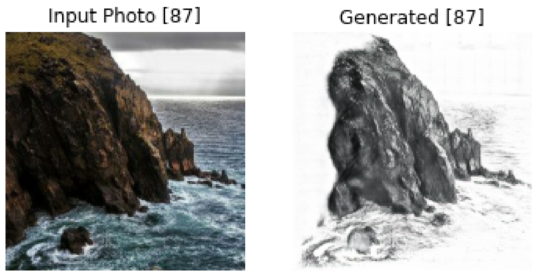


Figure 1: Example of generated image

2. Dataset

For the model training I've used two main datasets. The first one is called ImageNet-Sketch[8] and consists of 50000 sketch images, 50 images for each of the 1000 ImageNet classes. The other one consists of nearly 7000 photos mainly of landscapes [9].

Additionaly I used the monet2photo[9] dataset with about 1000 Monet paintings and AnimalFaces[1] dataset with over 16 000 photos of animal faces.

All images were resized to 256x256 pixels and scaled to $\langle -1; 1 \rangle$ range. Images were all converted to have three channels (rgb) for consistency. I have also experimented with converting images to grayscale, which will be discussed later.

3. Related Work

A popular approach for style transfer are Generative Adversarial Networks (GANs). This approach uses two Neural Networks (NNs) - a generator and a discriminator. The generator tries to produce required output based on some input (in our case hand-drawn image based on a photograph). The discriminator is given real samples (real photographs) and generated samples (from the generator) and has to decide which samples are real and which are generated. The idea is that by learning to fool the discriminator the generator learns to produce desired (real-looking) images.

We can categorize image translation methods to two categories based on wherever they need paired data (inputs and reference outputs for them). Approaches requiring paired data may be more straightforward as we have clear target, but have the drawback of requiring reference dataset, which might be very hard to obtain.

Since we do not have paired data (inputs and reference outputs) we are limited only to models which work with unpaired datasets. I also think these models are much more interesting as it is generally difficult to get the paired datasets.

The most relevant paper for my work is CycleGAN[9]. CycleGAN is a model that aims to solve the image-to-image translation problem without requiring paired input-output images, using cycle-consistent adversarial networks. The main idea behind the model is that there are two generators (and discriminators) and the model learns both translation from the source domain to target domain and the reverse translation. The model is then penalized if the transformation to target domain and back differs from the original image.

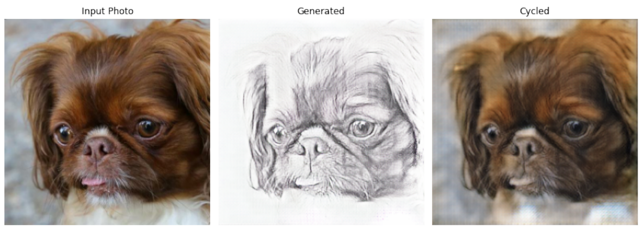


Figure 2: Illustration of cyclic generation

4. Methods

The main part of the work was using the CycleGAN architecture. The model was implemented and tested on the selected datasets. Further, I have done some experiments to try to improve the model quality.

The generator I used is based on the UNET[7] architecture. The discriminator is a Convolutional Neural Network (CNN).

I have looked into other methods than CycleGAN but none proved useful in my case. Two interesting papers which focus on sketching are Im2Pencil[5] and ArtPDGAN[4], but both require paired data. I have briefly experimented with FUNIT[6] and UGATIT[2], however both were unfeasible to train due to their hardware requirements. Another paper which studies photo-sketching is [3], however the result do not look very good for our use case.

5. Experiments

The first experiment was with the generator architecture. The original architecture can be seen as composed of down-scaling convolution blocks and up-scaling deconvolution blocks, with skip connections between corresponding (for example first and last) up and down layers. I have tried tried replacing some of the latter down blocks with convolution blocks that keep the size. Overall the results were worse than the original. Precise architectures are available in the source code.

The second experiment was to first preprocess images to grayscale. As mentioned above the CycleGAN has to also be able to convert drawings to images. This task might be hard because the color information is missing from the sketches and often cannot even be inferred reasonably (for example clothes or houses might have different colors). The idea was, that converting the images to grayscale might make the task easier and produce better results.

Lastly I have experimented with the influence of the input and target datasets. For this I have collected two additional datasets as described in previous section.

6. Results

The model is able to generate good looking sketches from some images after a relatively short training. However overall the results are not very reliable. The first problem with the generated images is the appearance of artifacts, see figure 3.

The second problem is that often large portions

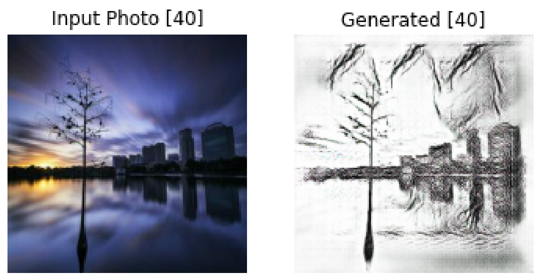


Figure 3: Example of artifacts

of the generated image are simply left empty, see figure 4. This can sometimes improve the result and make it feel more “artistic”, however it may also lead to the image simply missing important parts.

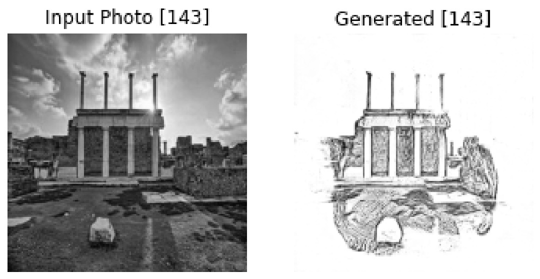


Figure 4: Example of image with empty space

Figure 5 shows loss for various runs, which are described in appendix A. Contrary to expectations I did not observe any difference with or without the conversion to grayscale in preprocessing. This could be explained by the model being able to generate “good enough” color pictures. See figure 2, where the overall color of the image matches the original (if we do not focus on details). However other explanations are possible and it might be interesting to explore this further.

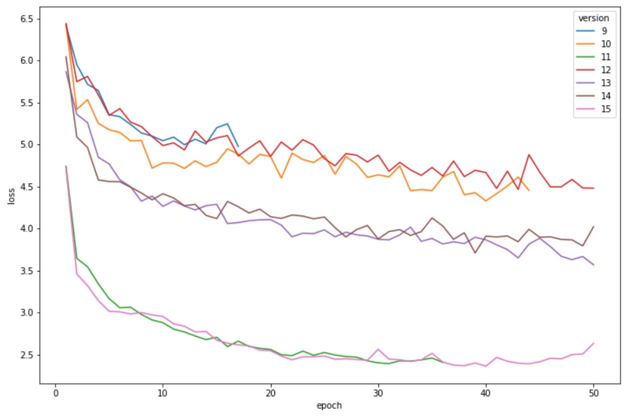


Figure 5: Loss for different runs

The loss reached on different runs (displayed on figure 5) can naturally be seen is falling into three groups. As can be seen on figure 6 the main difference is in used datasets. When using the "Monet" dataset as target the model achieves the best results, regardless of input dataset. When using the "sketch" dataset as target the model has more problems to learn. In this setting the model achieves better on the "animals" (source) dataset.

The results on different datasets ended as expected. The "Monet" dataset usually has some "texture" which can be relatively easily copied to target image to achieve good results (at least at the first sight). The "animal-faces" dataset is closer to the "sketches" dataset, includes only one object and often provides clear lines.

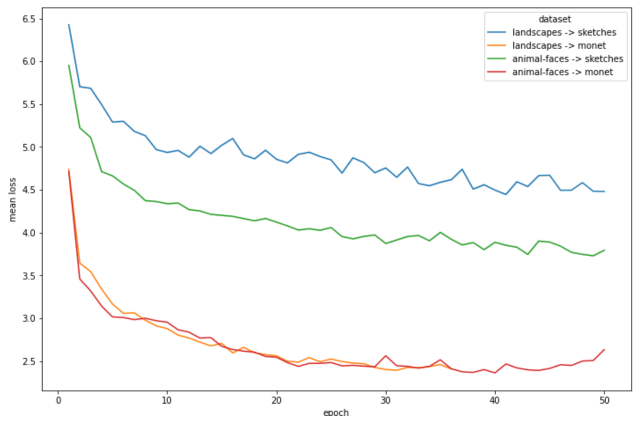


Figure 6: Loss by used datasets

See apendix B for more examples of generated images.

7. Conclusion

In this work I have explored the task of style-transfer with focus on transformation to sketches. I have shown that current state-of-the art models can produce, at least in some cases, promising results. I have found no difference between converting color and grayscale images. I have shown differences in results based on source and target datasets.

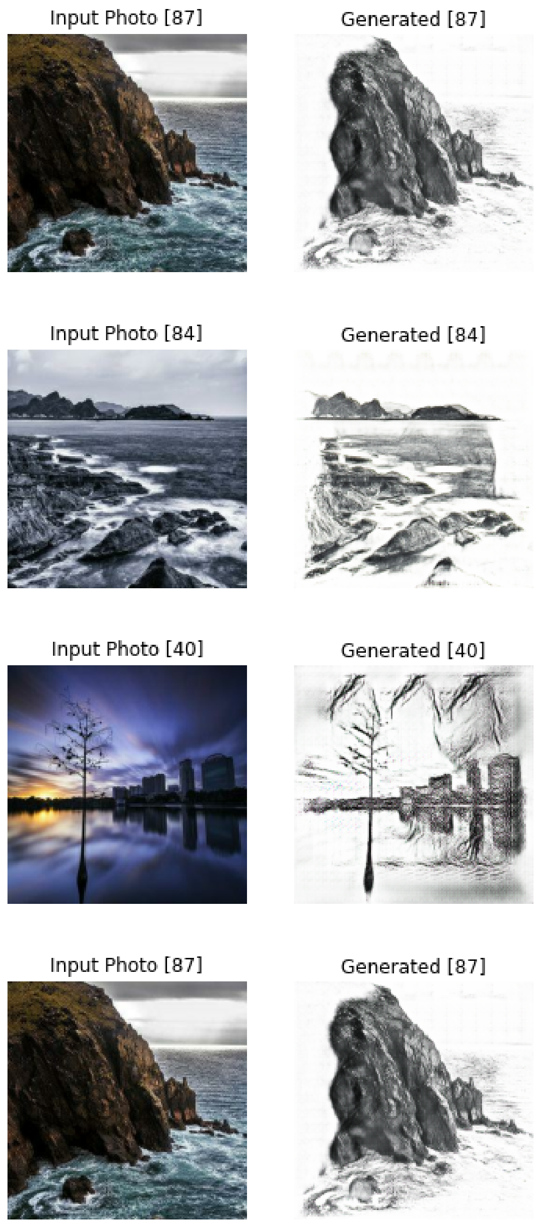
Overall I think the topic of style-transfer is very interesting and could find many applications in domains of art and content-generation.

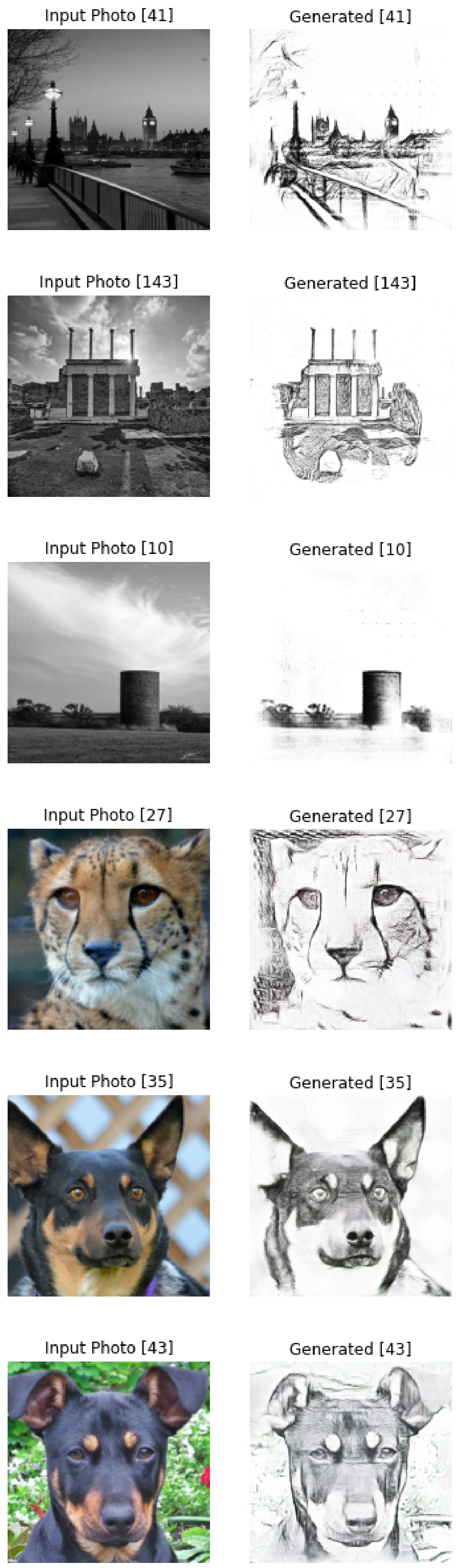
Source code including results and trained models is available at `https://www.kaggle.com/davidmasek/public-photo-to-drawing-style-transfer`.

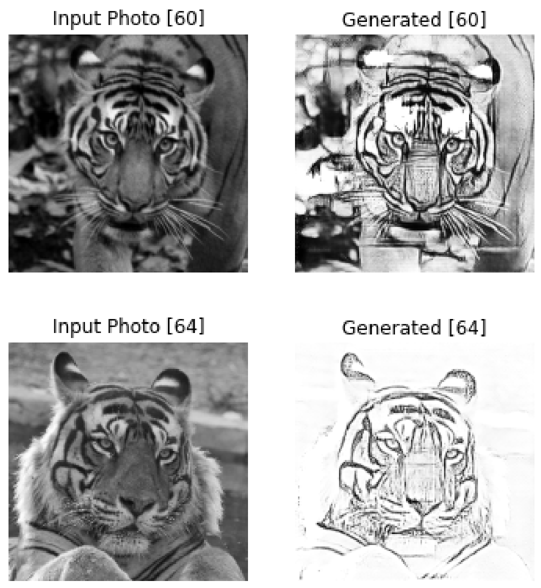
A. Overview of run configurations

run id	color	source dataset	target dataset
9	rgb	landscapes	sketches
10	grayscale	landscapes	sketches
11	rgb	landscapes	monet
12	rgb	landscapes	sketches
13	rgb	animal-faces	sketches
14	grayscale	animal-faces	sketches
15	rgb	animal-faces	monet

B. Gallery of generated images







References

[1] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[2] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. In *International Conference on Learning Representations*, 2020.

[3] Mengtian Li, Zhe Lin, Radomír Mˇ ech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *WACV*, 2019.

[4] SuChang Li, Kan Li, Ilyes Kacher, Yuichiro Taira, Bungo Yanatori, and Imari Sato. Art-pdgan: Creating artistic pencil drawing with key map using generative adversarial networks. In *International Conference on Computational Science*, pages 285–298. Springer, 2020.

[5] Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, and Ming-Hsuan Yang. Im2pencil: Controllable pencil illustration from photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[6] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10551–10560, 2019.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[8] Haohan Wang, Songwei Ge, Eric P Xing, and Zachary C Lipton. Learning robust global representations by penalizing local predictive power. *arXiv preprint arXiv:1905.13549*, 2019.

[9] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.