Modern Approach to Texture synthesis

July 2, 2024

Matteo Rampolla

Abstract

Texture synthesis is a fundamental task in the field of computer vision and graphics. Such task aims to generate a larger and more visually appealing texture starting from a smaller sampler of texture. My project tries to explore one of the most modern approaches to such a problem, by capturing and learning the essential characteristics of the input texture and trying to reproduce it in an output image of a bigger size. The methodology includes the design and training of a Generative adversarial Networks (GANs) that incorporate texture analysis and synthesis, emphasizing feature extraction, pattern recognition, and image generation processes.

1. Related Works

The first approaches to tasks of this type began to be produced around the 1996 by (Efros & Freeman, 2001). Their work has laid the groundwork for texture synthesis by introducing a non-parametric technique based on the importance of local similarity. Their algorithm worked by sampling the appearance of small neighborhoods around each pixel in a source texture, this work has been the foundation of many subsequent studies such as the one by Wei and Levoy (Wei & Levoy, 2001) that optimized the process of texture synthesis with the introduction of a similar but more efficient algorithm using tree-structured vector quantization. More works were developed in this field until the introduction of deep learning which gave a new and different approach to texture synthesis tasks. The most relevant starting point for this new type of approach has been the work of Gatys et al. in 2015 (Gatys et al., 2015). Their demonstration of how deep neural networks could capture and manipulate texture information opened up new avenues for research, leading to significant developments in both texture synthesis and style transfer. A more recent approach was introduced to use with a new type of architecture: Generative Adversarial Networks (GANs). One of the most relevant works has been TextureGAN by Xian et al. in 2017 (Xian et al., 2017) which showcased the potential of GANs in generating realistic and controllable textures. This approach highlighted the possibilities of using advanced machine-learning techniques to push the boundaries of texture synthesis further. These key contributions collectively represent the most relevant works in the field of texture synthesis, each marking a significant step forward in our ability to generate new textures from sample images

2. Methodology

The approach used in this homework tries to combine different techniques (parametric and non-parametric approach) that have been refined over the previous years and compare them together. As mentioned above, the main idea is to use a Generative Adversarial Network to create a generator capable of receiving as input a smaller texture and producing as output a larger texture with visually similar characteristics to the original one. From a more technical point of view, this means that the generator will be trained together with a discriminator, the role of the generator will be to create more and more convincing textures to fool the discriminator, which will instead have to distinguish the real image from the generated one.

Architecture. The Generator in question is a standard encoder-decoder architecture that was created based on the work of (Johnson et al., 2016) and is more than adequate as we need to capture large-scale non-stationary behavior. To do so they introduced a chain of residual blocks that allowed them to enlarge our receptive field to about the same size as our original input.

The discriminator adopted instead was taken from the work of (Isola et al., 2016). This fully convolutional network halves the spatial resolution of the input four times while doubling the number of channels. Everything will then be projected into a scalar (by using a 1x1 conv + sigmoid) and the resulting pattern is then classified as a real one or a fake one generated by our generator thanks to the use of a BCE.

Email: Matteo Rampolla <rampolla.1762214@studenti.uniroma1.it>.

Deep Learning and Applied AI 2023, Sapienza University of Rome, 2nd semester a.y. 2022/2023.

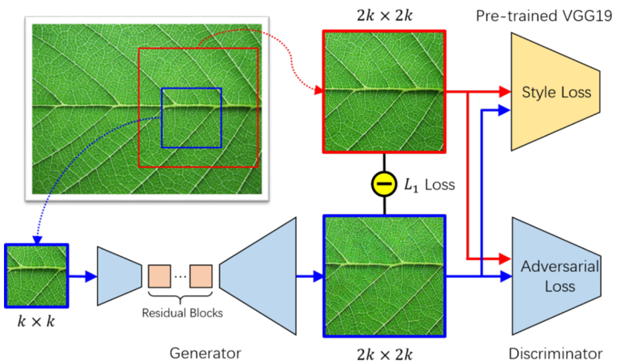

Figure 1. Methodology of training for our model, and loss combination used.

Figure 2. Results obtained with Non Parametric Approach

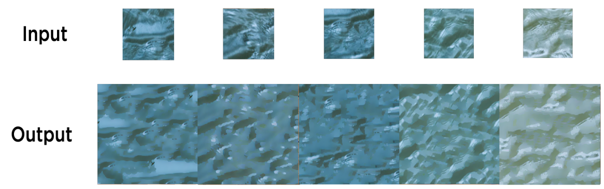

Figure 3. Results obtained with Parametric Approach

3. Training

The main training procedure was obtained from the methods shown in Fig 1 taken from (Zhou et al., 2018). For each original texture sample, a random crop of size 2k x 2k is created during training, then from this random crops we create another smaller one of size k x k. These two obtained images will be used as the input of the generator (k x k) respectively, and the larger image (2k x 2k) will be used as the ground truth "label".

The main loss function was obtained by combining the general adversarial loss function of (Zhou et al., 2018) alongside the style loss from (Gatys et al., 2015)

$$L_{\text{total}} = L_{\text{adv}} + \lambda_1 L_{L1} + \lambda_2 L_{\text{style}} \tag{1}$$

The value of the lambda1 and lambda2 are taken from (Zhou et al., 2018) while the style loss is obtained from a pre-trained VGG-19 model, and computes Gram matrices for the ReLU-activated feature maps output (Gatys et al., 2015)

4. Results

Non Parametric Approach The results of the algorithm used are particularly effective for semi-structured textures as well as stochastic textures. The two most typical problems are excessive repetitions, and mismatched or distorted boundaries. Both are mostly due to the input texture not containing enough variability

General Deep Model Although the reference paper (Zhou et al., 2018) starts experimenting early on with a specialized model based on different texture types, I decided to see how the same model would perform with a much more general dataset. The resulting model created through training of more than 1500 iterations failed to converge, and since the GPU availability of colab is limited I switched to more specific models, based solely on one type of texture (grass, dirt, water, ecc..).

Specialized Deep Models The second approach, on the other hand, was achieved by following the methodology described. As a result, the output textures 3 (More example are present in the github repository), while always limited in some way, yielded a satisfactory result. The whole training procedure was repeated multiple times with always different texture samples (Bricks, Wood, Grass, etc...), this means that each model is able to generate an appropriate texture as output only for that same type. All models were trained for over 1000 iterations. While the results were not an exact match to the input, achieving optimal results was constrained by Colab limitations. However, as demonstrated by (Zhou et al., 2018), a longer training schedule would likely yield results much closer to the original.

5. Conclusions and Future Works

Initial non-parametric approaches have achieved qualitative results, but image quilting, despite its simplicity and efficiency for repetitive textures, has drawbacks. It struggles with complex textures, leading to repetitive patterns, and has significant processing delays due to exponential complexity with larger input images, making it unsuitable for real-time applications. In contrast, deep learning approaches require substantial computational power for model training but generate textures much faster afterwards. The training phase captures the statistical essence of textures, enabling rapid creation of large textures during the forward pass through the network.

Bibliography.

References

Efros, A. A. and Freeman, W. T. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pp. 341–346, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113374X. doi: 10.1145/383259.383296. URL https://doi.org/10.1145/383259.383296.

Gatys, L. A., Ecker, A. S., and Bethge, M. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *CoRR*, abs/1505.07376, 2015. URL http://arxiv.org/abs/1505.07376.

Isola, P., Zhu, J., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. URL http://arxiv.org/abs/1611.07004.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. URL http://arxiv.org/abs/1603.08155.

Wei, L.-Y. and Levoy, M. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pp. 355–360, New York, NY, USA, 2001. Association for Computing Machinery. ISBN 158113374X. doi: 10.1145/383259.383298. URL https://doi.org/10.1145/383259.383298.

Xian, W., Sangkloy, P., Lu, J., Fang, C., Yu, F., and Hays, J. Texturegan: Controlling deep image synthesis with texture patches. *CoRR*, abs/1706.02823, 2017. URL http://arxiv.org/abs/1706.02823.

Zhou, Y., Zhu, Z., Bai, X., Lischinski, D., Cohen-Or, D., and Huang, H. Non-stationary texture synthesis by adversarial expansion. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 37(4):49:1–49:13, 2018.