The Journal of Open Source Software

pystiche: A Framework for Neural Style Transfer

Philip Meier[1] **and Volker Lohweg**[1]

1 inIT — Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe (TH-OWL)

DOI: 10.21105/joss.02761

Software

- Review
- Repository
- Archive

Editor: Kristen Thyng

Reviewers:

- @kthyng

Submitted: 08 October 2020
Published: 19 October 2020

Summary

The seminal work of Gatys, Ecker, and Bethge gave birth to the field of *Neural Style Transfer* (NST) in 2016 (Gatys, Ecker, & Bethge, 2016). The general idea behind an NST can be conveyed with only three images and two symbols:

In words: an NST describes the merger between the content and artistic style of two arbitrary images. This idea is nothing new in the field of *Non-photorealistic rendering* (NPR) (Gooch & Gooch, 2001). What distinguishes NST from traditional NPR approaches is its generality: an NST only needs a single arbitrary content and style image as input and thus "makes – for the first time – a generalized style transfer practicable" (Semmo, Isenberg, & Döllner, 2017).

Besides peripheral tasks, an NST at its core is the definition of an optimization criterion called *perceptual loss*, which estimates the perceptual quality of the stylized image. Usually the perceptual loss comprises a deep neural network that needs to supply encodings of images from various depths.

`pystiche` is a framework for NST written in Python and built upon the *Deep Learning* (DL) framework PyTorch (Paszke et al., 2019). It provides modular and efficient implementations for commonly used perceptual losses (Gatys et al., 2016; Li & Wand, 2016; Mahendran & Vedaldi, 2015) as well as neural net architectures (Krizhevsky, Sutskever, & Hinton, 2012; Simonyan & Zisserman, 2014). This enables users to mix current state-of-the-art techniques with new ideas with ease.

Due to its vivid nature, the field of NST gained a lot of traction in the short time after its emergence (Jing et al., 2019). While many new techniques or augmentations have been developed, the field lacks standardization, which is especially evident in the reference implementations of the authors. `pystiche` aims to fill this gap.

Statement of Need

Currently, unlike DL, there exist no library or framework for implementing NST. Thus, authors of new NST techniques either implement everything from scratch or base their implementation upon existing ones of other authors. Both ways have their downsides: while the former dampens innovation due to the lengthy implementation of reusable parts, with the latter the author inherits the technical debt due to the rapid development pace of DL hard- and software. In order to overcome this, `pystiche` pursues similar goals as DL frameworks:

1. **Accessibility** Starting off with NST can be quite overwhelming due to the sheer amount of techniques one has to know and be able to deploy. `pystiche` aims to provide an easy-to-use interface that reduces the necessary prior knowledge about NST and DL to a minimum.
2. **Reproducibility** Implementing NST from scratch is not only inconvenient but also error-prone. `pystiche` aims to provide reusable tools that let developers focus on their ideas rather than worrying about everything around it.

`pystiche`s core audience are researchers, but its easy-to-use user interface opens up the field of NST for recreational use by laypersons.

Acknowledgements

This contribution is part of the project *Fused Security Features*, which is funded by the *Ministry for Culture and Science of North Rhine-Westphalia* (MKW NRW) under the Grant ID `005-1703-0013`. The authors thank Julian Bültemeier for extensive internal testing.

References

Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*. doi:10.1109/CVPR.2016.265

Gooch, B., & Gooch, A. (2001). *Non-photorealistic rendering*. A. K. Peters, Ltd. ISBN: 1568811330

Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2019). Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*. doi:10.1109/TVCG.2019.2921336

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems 25 (NIPS)*. Retrieved from https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-pdf

Li, C., & Wand, M. (2016). Combining markov random fields and convolutional neural networks for image synthesis. In *IEEE conference on computer vision and pattern recognition (CVPR)*. doi:10.1109/CVPR.2016.272

Mahendran, A., & Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *IEEE conference on computer vision and pattern recognition (CVPR)*. doi:10.1109/CVPR.2015.7299155

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32 (NIPS)*. Retrieved from http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Semmo, A., Isenberg, T., & Döllner, J. (2017). Neural style transfer: A paradigm shift for image-based artistic rendering? In *Proceedings of the symposium on non-photorealistic animation and rendering (NPAR)*. doi:10.1145/3092919.3092920

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Computing Research Repository (CoRR), abs/1409*. Retrieved from http://arxiv.org/abs/1409.1556