Towards Accessible Photorealistic Style Transfer

Wesley Williams, Supervised by Dr. Sion Hannuna

University of Bristol

Introduction

Photorealistic style transfer transforms a content photo using the style of another whilst ensuring that the result is photorealistic.

The area is very closely related to the area of artistic style transfer. Apps such as Prisma have been hugely popular in this area by allowing users to apply an artistic style from a selection to their photos. A similar tool for applying photorealistic style transfer would be popular as an alternative to manually editing a photo to achieve an effect.

Although there has been advances in the area, there is yet to be a solution which users can easily style their photos. Current solutions require users to configure their environment with libraries such as PyTorch and use Python scripts to style their photos. Most of the existing techniques requires high-end GPUs to achieve attractive results.

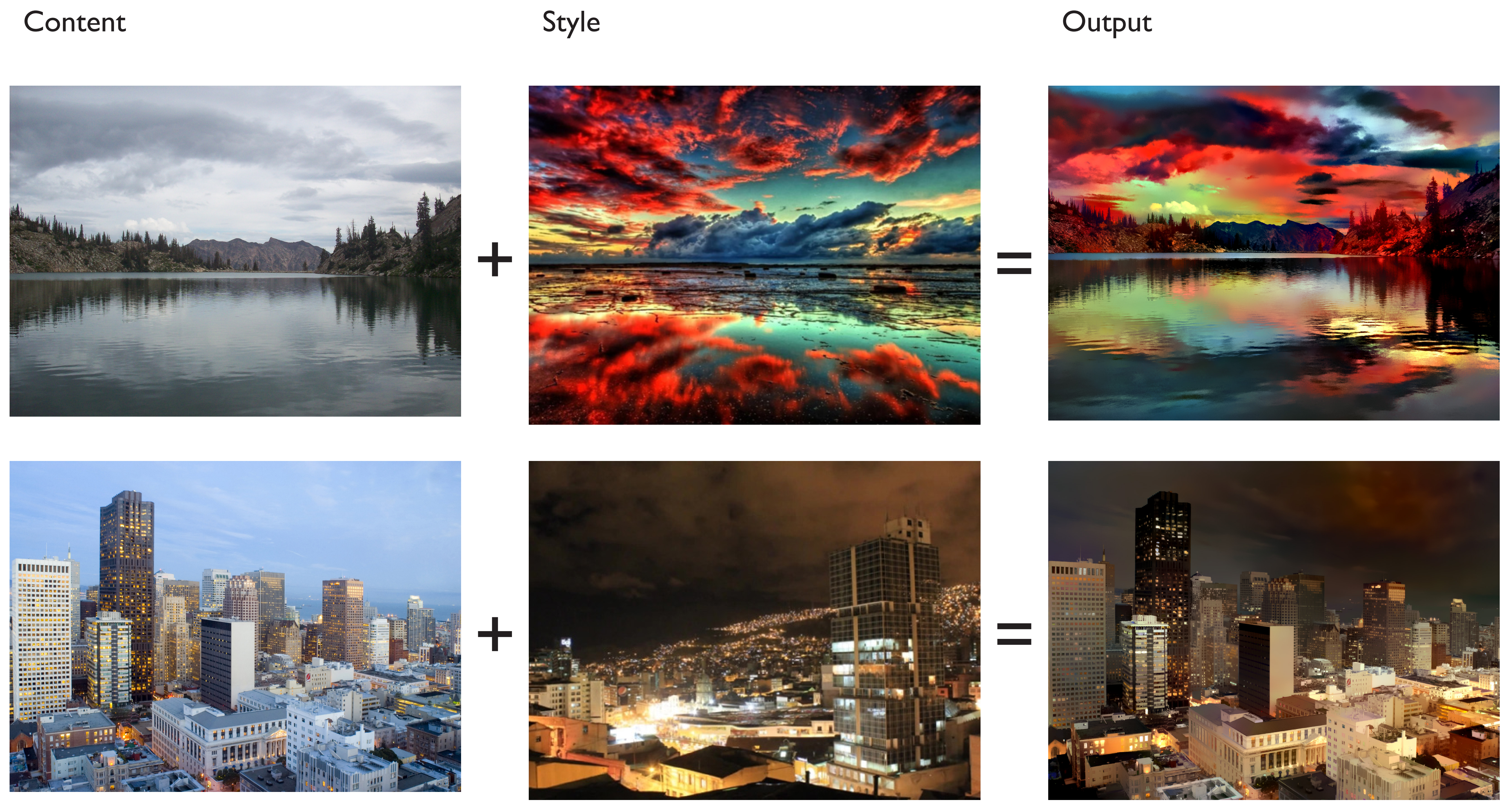
Aims

- Evaluate current solutions and select an approach that balances inference time, quality and flexibility.
- Produce a packaged tool to allow creatives to easily experiment, without the burden of configuring an environment and running source code.
- Explore methods of further improving the performance, with regard to lessening the need for high-end hardware.

Progress

- Evaluated and experimented with several techniques of performing photorealistic style transfer.
- Involved the setup and understanding of numerous different network architectures and frameworks, each with their own hurdles.
- Explored methods for inference on end-user hardware.
- Contributed towards open-source projects in the area, allowing others to experiment with techniques easier.

Examples



Source: [2]

Selected Approach

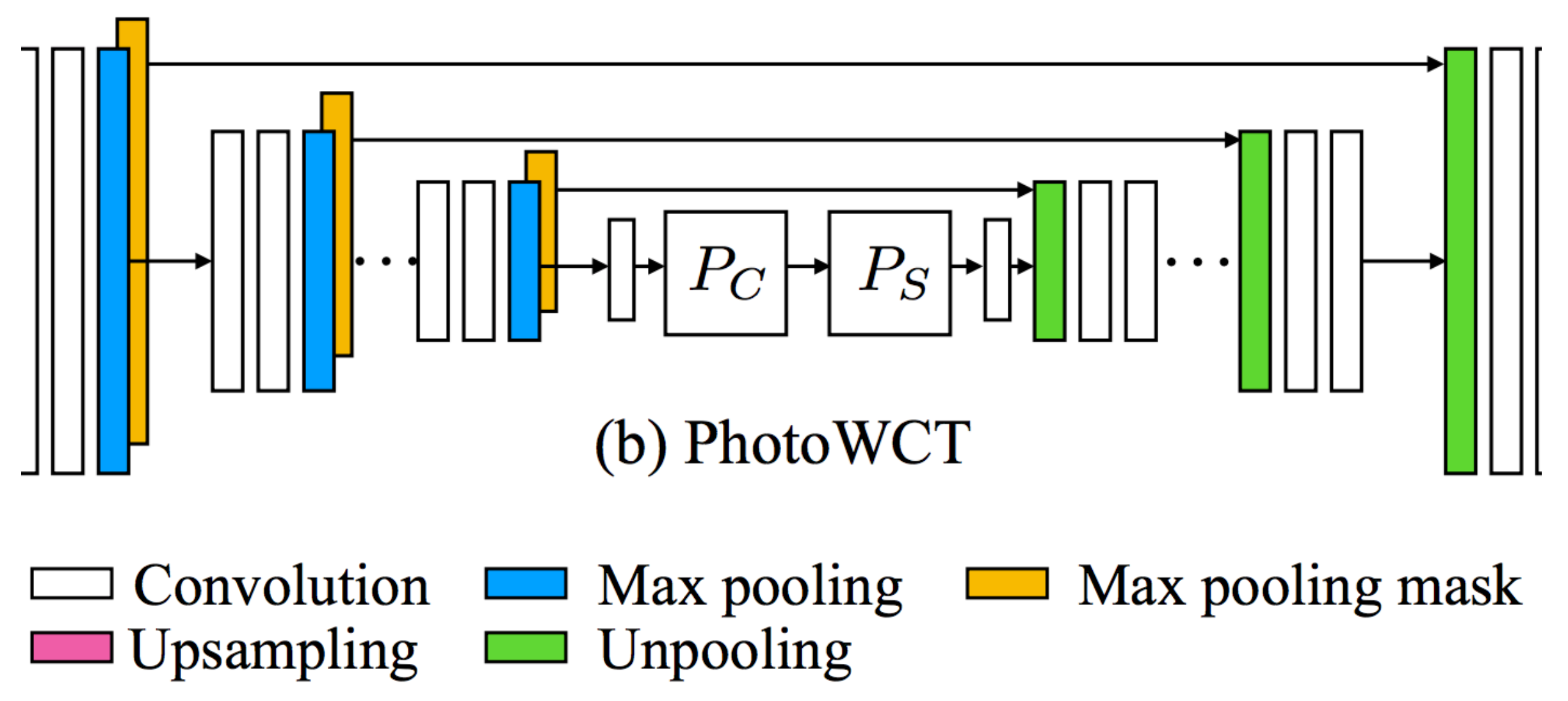
Originally I was working on extending the work of [3] by adding a photorealistic smoothing step such as the SPE suggested in [4]. However, during the project a paper [1] was released which took a similar approach but also improved on the steps from [3] and proposed a better smoothing technique.

A solution using similar techniques to [1] seems the most applicable to this project. Because:

- Allows arbitary styles to be applied (no prior training of a specific style).
- 60 times faster than previous state of the art.
- Results are more photorealistic than previous state of the art.

A summary of the technique is as follows:

- Use a pretrained VGG-19 model as an encoder, and train a decoder to reconstruct the image.
- Whitening and coloring transforms used on the encoded features in order to match the statistics of content features with those of style features.
- Applying a smoothing step to remove artifacts.



Encoder architecture and projection steps. P_C,P_S represent the whitening and coloring transforms respectively. Source: [1]

Challenges

- Solutions currently require significant processing time on high-end hardware to produce results.
- Evaluating different projects has required numerous frameworks to be used and extra packages to be installed on BlueCrystal.
- Methods use uncommon layers (e.g. unpooling layer) making their use difficult, some libraries are yet to support this layer.
- Rapidly developing area (new state of the art published on 22/02).
- Limited literature on inference within desktop applications. Most literature discusses inference at data centres or on mobile devices/embedded systems.
- Some operations may be too costly to perform on the end-user device, a compromise may have to be made on techniques used.
- Most deep learning libraries are designed with inference in the data centre in mind, they are not easy to package into an application to run on end-user devices.
- Some published implementations are incomplete (no training code etc.).

Future Work

- Implement [1] with Caffe2.
- Create a tool to interface with the network for desktop users.
- Try different networks. Can a lightweight alternative to VGG-19 be used (e.g. MobileNet, SqueezeNet).
- Explore options for mobile (CoreML, Caffe2 etc.)
- Compare the performance trade off of different smoothing techniques.

References

- [1] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz. A Closed-form Solution to Photorealistic Image Stylization.
- [2] F. Luan, S. Paris, E. Shechtman, and K. Bala. Deep photo style transfer. In CVPR, 2017.
- [3] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang. Universal style transfer via feature transforms.
- [4] R. Mechrez, E. Shechtman, and L. Zelnik-Manor. Photorealistic style transfer with screened poisson equation