Arbitrary Style Transfer

Aditya Gupta 1231965425 **Rutwik Keskar** 1229665036 **Gandhar Bichkar** 1229572606

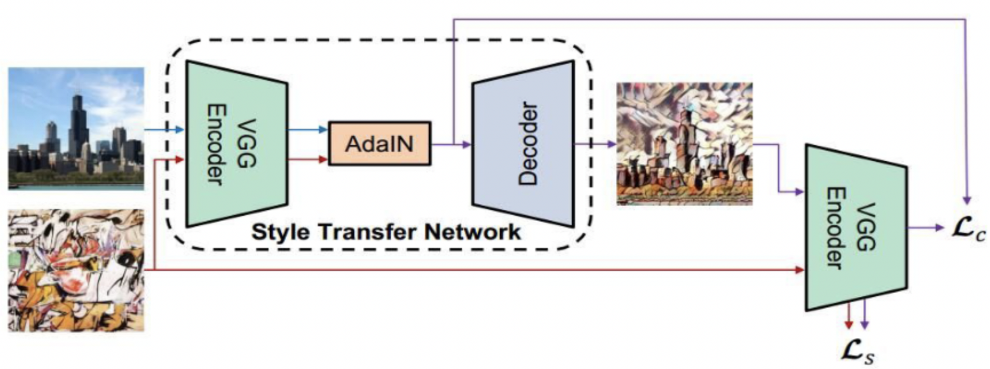
Abstract

Objective: Style transfer is an optimization technique used to take two images—a content image and a style reference image (such as an artwork by a famous painter)—and blend them together so the output image looks like the content image, but “painted” in the style of the style reference image.

Motivation: We got very interested after looking at the results that this paper produces. Through this project, we aim to push the boundaries of image synthesis and inspire new avenues of artistic exploration in the digital landscape. We are also determined to optimise this further or add more functionality.

Expected outcome: We should have the capability to apply stylistic elements from any given style image to a content image. Additionally, we should be able to adjust parameters to control the degree to which the stylization is applied. This could allow us to transform black and white vintage images into vibrant, colorful images according to our preferences.

Example of styling



$\alpha = 0$ $\alpha = 0.25$ $\alpha = 0.5$ $\alpha = 0.75$ $\alpha = 1$ Style

1 Execution Plan

1.1 Steps to follow

The paper talks about 3 components of the architecture which will be implemented or incorporated in our architecture. The first one is an encoder which is VGG19. Next one the middle layer or the AdaIN layer that the paper talks about. Here the characteristics are transfered from the style image to the content image. The final component is the decoder that we need to train on MS-COCO dataset. In later steps we wil try to add experiments of putting a parameter to tell upto what extend the content image needs to be stylised. Another experiment would be of combining multiple style images together. If time permits, we will to experiment with styling by text. In the final step we will try to make it more user friendly by providing commnadline functionalities.

1.2 Workload distribution

Our three-member team will collaboratively contribute to every phase of the project - coding, debugging, training, testing, updates. Being a collaborative task, the workload of each member will be 1/3rd of the total work done to achieve the outcome of the project.

1.3 Time table

We will try to finish implementing entire architecture by 10th April. We will try to train our model on MS-COCO dataset by 20th April and implement both the experiments. Later we would follow the final steps, or try to optimise and experiment by using newer models which would allow the user to stylise image based on text.

1.4 Expected challenges and how to handle them

We might face resource limitation such as google colab having limited usage and time limits. We might have a set-up a collaborative working environment that has coding sharing across multiple accounts.

2 Evaluation plan

2.1 Evaluating the outcome of your project

The easiest way to evaluate the outcome is though visual observation of the output image by seeing if it has styles (patterns) similar to the style image.

A quantifiable metric for evaluating the stylized image is by using the Peak Signal-to-Noise Ratio (PSNR) ratio. PSNR measures the quality of the reconstruction by comparing the pixel values of the generated image with the target image. Higher PSNR values suggest better quality.

2.2 Evaluating your performance + peer review

When evaluating individual performance and peer reviews, our approach will be multifaceted and focused on objective assessment. Individual performance will be evaluated based on several criteria, including the quality and timeliness of work, adherence to project guidelines, contribution to team collaboration and problem-solving, and the ability to meet assigned milestones. Peer reviews will play a crucial role in this evaluation process, where team members will provide constructive feedback on each other's work, assess code quality, identify areas for improvement, and recognize strengths. This peer feedback will be collected and analyzed to gain insights into each team member's contributions, foster continuous improvement, and ensure that project objectives are met efficiently.

References

The following references have been chosen for implementing the project. The first reference is the research paper that we will implement and the second one is the paper that we would study to optimise it further. The third one is the documentation of pytorch. The fourth one is the VGG19 pre-trained model source.

[1] Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization: Xun Huan and Serge Belongie
[2] Neural Style Transfer: A Review Yongcheng Jing, Yezhou Yang, Member, IEEE, Zunlei Feng, Jingwen Ye, Yizhou Yu, Senior Member, IEEE, and Mingli Song, Senior Member, IEEE
[3] https://pytorch.org/docs/stable/index.html
[4] https://iq.opengenus.org/vgg19-architecture/