

Author Guidelines for ECCV Submission

Anonymous ECCV submission

Paper ID ***

Abstract. The abstract should summarize the contents of the paper. LNCS guidelines indicate it should be at least 70 and at most 150 words. It should be set in 9-point font size and should be inset 1.0 cm from the right and left margins. ...

Keywords: We would like to encourage you to list your keywords within the abstract section

1 Introduction

Transferring the style or texture of one image to another is a form of texture transfer technique. Texture transfer techniques are used in arts industry to produce high quality synthetic images of real world object. An example would be to transfer the texture of bark to an art asset such as a 3D model of a tree.

Texture transferring is a difficult task that requires specialized tools, time, and the skills of an experienced artist to achieve good results. Conventional techniques for texture transfer usually include the use of tools such as texture cloner, brushes, and gradients, most often supplied by Adobe Photoshop.

Early attempts to solve the texture transferring task used low level image features and therefore had shortcomings in extracting any semantic meaning from the input image. With the improvements in object localization and detection using Convolutional Neural Networks (CNNs), extracting semantic information from images became more plausible. Neural style transfer is a technique that was first used to transfer art styles from one painting to a photography [1]. Neural style transfer generates new images with the help of CNNs.

In this project, we will try to provide a tool for architects and artists alike to use in their work. The tool will recreate a building image using a certain architectural style. This architectural style will be extracted from another building image. With this, an architect can see how a building would look like if it were to be constructed/renovated with respect to the architectural style of other buildings.

The problem we will try to tackle in this project is to use neural style transfer to transfer the architecture of a building to another building. Given images of building A and B, we want to generate an image G such that it contains the building of B with the architecture of the building in A. We used Darknets to generate the new buildings. A Darknet is a relatively small and compact neural network used for YOLO algorithms. YOLO algorithms are one pass object detection algorithms. We used Darknets because of their relatively small size

and because they recognize object in areas. This enables Darknets to conserve knowledge of object positions.

2 Background

The first neural style transfer technique compares the activations of a VGG network [1] to generate a new image. Gatys et al. [1] generated an image G using a content image C and style image S . An image G was generated by minimizing the following cost function:

$$C(C, S, G) = \alpha C_{content}(C, G) + \beta C_{style}(S, G) \quad (1)$$

where $C_{content}$ is the similarity of the activations of some layer, C_{style} is the similarity of the Gram-matrices of some layers, and α and β are weights that must be carefully tuned to get good results.

There have been many different neural style transfer techniques developed since, these are summarized in [3]. Some of which are Slow Neural Method Based on Online Image Optimization, Parametric Slow Neural Method with Summary Statistics, Nonparametric Slow Neural methods with MRFs, Fast Neural Method Based on Offline Model Optimization, Per-Style-Per-Model Fast Neural Method, Multiple-Style-Per-Model Fast Neural Method, and Arbitrary-Style-Per-Model Fast Neural Method. Because of the vast amount of neural style transfer algorithms created, we altered the original algorithm mentioned in [1].

Although the original NST algorithm proposed by Gatys et al. had good performance, the algorithm itself was slow. It takes a lot of time just to compute an image of size 512x512. As a result, people have tried to speed up the algorithm. Ulyanov et al used a method of feed-forward stylization by having a generator network $g(x, z)$, which can be applied to a given input image x and style image x_0 . This function g is a convolutional neural network learned from examples. It solves the problem:

$$\min_g \frac{1}{n} L(x_0, x_t, g(x_t, z_t)) \quad (2)$$

Although this improved the speed of the stylization, the results were not as good as Gatys. As a result, people have been tackling the problem of improving the speed. In [5], they proposed to alter the Ulyanov generator by using instance normalization at both training and testing times. Their results speed up the original neural style transfer algorithm while also improving the performance.

3 Approach

We decided to use the same approach as [1] but with a custom C_{style} . We trained a CNN f to minimize the following cost function:

$$C(A, P, N) = \max(\|f(A) - f(P)\|_2^2 - \|f(A) - f(N)\|_2^2 + \alpha, 0) \quad (3)$$

where α is called the margin, A and P belongs to the same class and N belongs to a different class. The style cost function was then defined to be the following

$$C_{style}(S, G) = ||f(S) - f(G)||_2^2 \quad (4)$$

The CNN f computes a style embedding; that is, a vector that encodes the style of an image.

References

1. Authors: The frobnicatable foo filter (2014) BMVC14 submission ID 324. Supplied as additional material **bmvc14.pdf**.

2. Authors: Frobnication tutorial (2014) Supplied as additional material **tr.pdf**.

3. Alpher, A.: Frobnication. Journal of Foo **12**(1) (2002) 234–778

4. Alpher, A., , Fotheringham-Smythe, J.P.N.: Frobnication revisited. Journal of Foo **13**(1) (2003) 234–778

5. Alpher, A., , Fotheringham-Smythe, J.P.N., Gamow, G.: Can a machine frobnicate? Journal of Foo **14**(1) (2004) 234–778