IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 2)
Available online at: www.ijariit.com

Artistic style transfer using deep learning

P. Mounica
mounica2798@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

A. Nagaratnam
nagaratnam.cse@anits.edu.in
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

Mohammad Farzana
mdfarzana123@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

K. Muralidhar
muralikoripalli@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

ABSTRACT

In this paper, we are implementing the style transfer using convolutional neural networks. The style transfer means to extract the style and texture of a style image and applying it to the extracted content of another image. Our work is based on the work proposed by LA Gatys. We use a pre-trained model, VGG 16 for our work. This work includes the content reconstruction and style reconstruction from the content image and style image respectively. Now the style and content are merged in a manner that the features of content and style are retained.

Keywords— *Style transfer, Artistic style, Content loss, Style loss, Gram matrix, VGG16, CNN, Texture transfer, Image synthesis, Imagenet*

INTRODUCTION

Changing the style of the content image based on the given style image onto another image can be treated as a problem of texture transfer. Here the goal is to transfer a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image. A large number of non-parametric algorithms exists for texture synthesis. These algorithms resample the pixels of a source texture for synthesizing photorealistic natural textures. A per-pixel loss function that calculates the difference between output and ground truth images is used to train a feed-forward convolutional neural network in a supervised manner. This loss function does not capture the perceptual difference between images. A pre-trained convolutional neural network is used for extracting high-level image features, and the differences are used to calculate perceptual loss functions which generate high-quality images. However, an extremely difficult problem is to separate content from style in photo-realistic images. High-level semantic information is extracted for Deep convolutional neural network to produce powerful computer vision systems.

In our work, we show how Deep Convolutional Neural Networks independently manipulate the content and style of natural images by learning the generic feature representation. In this paper, we combine the approaches of Johnson et al [6], Gatys et al[1].

Efros and Freeman had introduced a correspondence map which comprises of features of the target image such as image intensity to constrain the texture synthesis procedure [2]. Hertzman et al. use image analogies to transfer the texture from an already stylized image onto a target image [3]. Ashikhmin focuses on transferring the high-frequency texture information while preserving the coarse scale of the target image [4]. Lee et al. improve this algorithm by additionally informing the texture transfer with edge orientation information [5]. Usage of only low-level image features of the target image to inform the texture transfer is a fundamental limitation, though these algorithms achieve remarkable results. However, a style transfer algorithm must be capable of extracting the semantic image content from the target image and then inform a texture transfer procedure to render the semantic content of the output image in correspondence with the style of the source image. Therefore, a fundamental prerequisite is to find image representations that independently model variations in the semantic image content and the style is required.

1.1 Initiation of gradient descent

All the images were initialized with white noise. Image synthesis can also be initialized with content and style image. These two alternatives do not have a strong effect on the outcome. If we initialize with white noise arbitrary values of new images would be generated. Whereas if we initialize with fixed images it leads to an arbitrary number of images in the outcome.

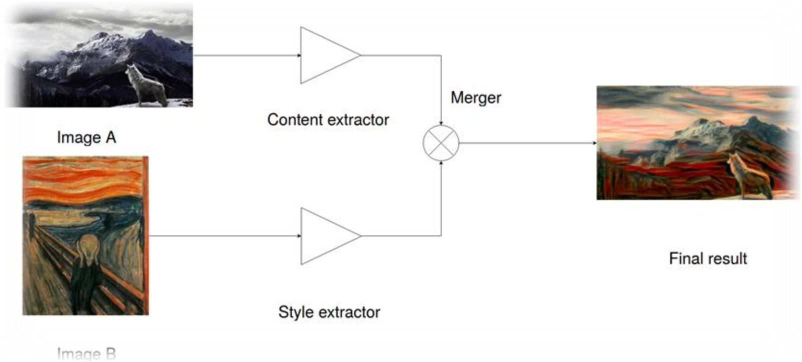

Fig. 1: Extracting content and style to get the final result

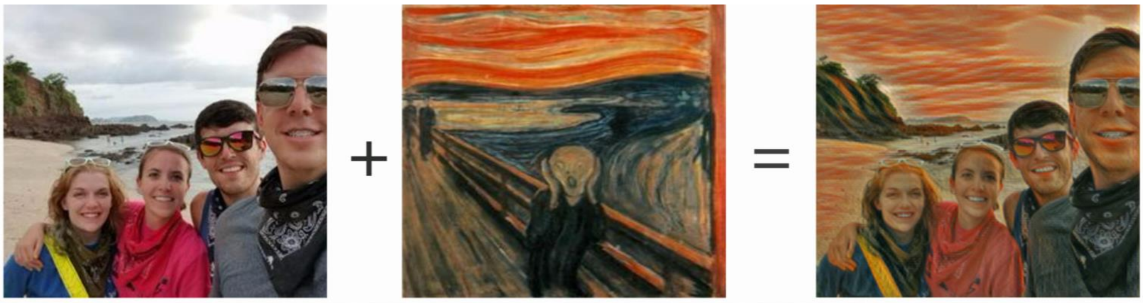

Fig. 2: Example which illustrates the content and style representation

1.2 Dealing with content and style matching

While synthesizing the content and style of the source image, we cannot obtain an image that matches both constraints simultaneously. However the loss function that minimized during image synthesis is a linear combination between content and style losses respectively, the emphasis is on forming content or style. However, if we focus more on content the output will match the photograph and style will not be well matched. Whereas if we focus more on style, the output will match the artwork. However one must adjust the trade-off between content and style image to create best outputs.

1.3 Effect of CNN layers

The choice of layers to match the content and style representation is also a factor for image synthesis. Multiple layers of neural networks are included in the style representation. Local-scale on which image is matched is determined by the number and positions of neural layers which leads to various visual images. When the style representations are matched to a higher scale which pressures the structure of local images more smooth and continuous visual images are obtained.

2. CNN ARCHITECTURE

In convolution neural network we take an input image and define a weight matrix and the input is convolved to extract specific features from the image without losing the information about its spatial arrangement.

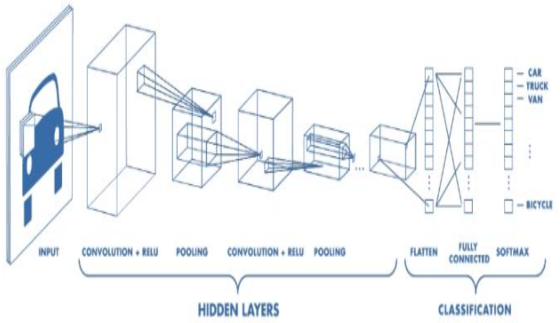

Fig. 3: Architecture of CNN

3. VGG 16 ARCHITECTURE

There are 16 convolutional layers in VGGNet. Because of its very uniform architecture, it is very appealing. It is currently the most preferred choice for extracting features from images. VGGNet is a bit challenging to handle as it consists of 140 million parameters. The content image, style reference image and input image are processed together such that input image is changed to look like a content image painted with style image.

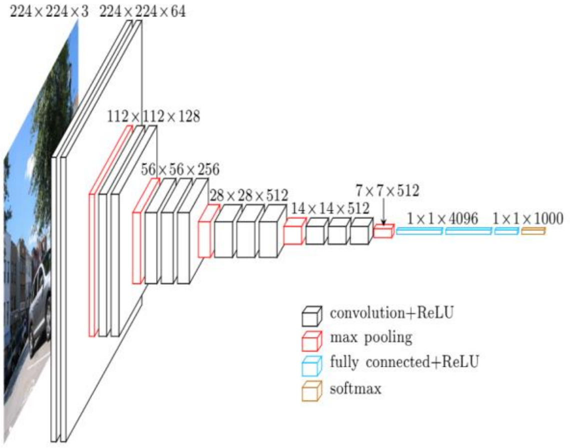

Fig. 4: VGG16 Architecture

4. CONTENT REPRESENTATION AND LOSS

A nonlinear filter bank functionality is provided by each layer in the neural network whose complexity depends on the position of the layer in the neural network. Based on the filter responses to that image, the given input image is encoded in each layer of the convolutional neural network. The number of distinct filters in a layer is equal to the number of feature maps, i.e. if there are 'n' number of filters there will be 'n' feature maps of size 'N' where N is the number of cells in the map, i.e. equal to product of height and width of the feature map.

We construct images whose feature maps at a chosen convolution layer match the corresponding feature maps of a given content image. We predict the two images to contain the same content but not inevitably the same texture and style.

Given a chosen layer l, the content loss is defined as the Mean Squared Error between the feature map F of our content image C and the feature map P of our generated image Y.

$$L_{content} = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^{2}$$

Minimizing the content loss means that the mixed image is likely to have a feature activation in the layers which is similar to the activation of content-image. To capture the finer textures we have to apply this to the starting layers and the deeper layers are used to capture higher level elements of image style.

5. STYLE REPRESENTATION AND LOSS

Firstly, we need to know which features of the style-layers are active simultaneously in the style image so that we can copy the activation-pattern to the mixed image. For this, we need to calculate the gram matrix for tensors output by the style-layers. A gram matrix is one which comprises of correlated features. It is calculated by taking the dot product for the vectors of feature activations of a style-layer. When an entry in a gram matrix is a value close to zero it means that the two features in the style layer do not activate in a style image. We can say that the two features activate simultaneously in style-layers when the value of an entry in the Gram Matrix is huge. We then replicate this activation pattern to create a mixed-image. [1,7]

Each entry in the Gram matrix G can be given below where F is a feature map:

$$G_{ij} = \sum_{k} F_{ik}F_{jk}$$

The loss function for style is quite similar to the content loss, except that we calculate the Mean Squared Error for the Gram-matrices instead of the raw tensor-outputs from the layers.

$$L_{content} = \frac{1}{2}\sum_{i,j}^{L}\left(G_{ij}^{l} - A_{ij}^{l}\right)$$

6. APPROACH

The results were based on the VGG network. Here we train the VGG network with imagenet dataset to pre-train the model. The VGGNet model with 16 layers that is VGG16 is used. The imagenet dataset consists of over 10 million images. VGGNet is good in localization and classification tracks. These models perform well in computer vision tasks.

6.1 Reading content and style images

Initially, the content and style images are loaded. Height and width of both the images should match so that we can apply to style to the content image more appropriately. The height and width of the image to be 512×512. Any other dimensions can also be used. And the image contains RGB value. So the dimension of the image would be 512×512×3.

Once the images are loaded properly, an extra dimension to the content and style images are added. This is done as to concatenate the images via X-axis to obtain a tensor.

Before proceeding further to obtain the output image some transformations have to be done to the content and style images. First, subtract the mean values of RGB from both the images. The mean values are calculated from the imagenet dataset. Next, the order of RGB is flipped to BGR because VGG network works on BGR. At the end again flipping is done to the ordering of the combined image i.e, styled image to get the proper RGB image.

The style transfer problem can be thought of as an optimization problem. Here the loss function we want to minimize is divided into a content loss, style loss and total variation loss. The importance that we have to give to these losses are determined by predefined scalar weights.

6.2 Obtaining a combination image

A placeholder variable is used to store the combination image. It would retain the content of the content image while incorporating the style of the style image.

Then concatenate all the three content, style and combination image into a tensor which is used by the VGG network.

6.3 Calculating content loss

We here follow the Johnson et. al (2016) and the content feature is drawn from the block2_conv2 as it gives more appealing results.

The content loss is the Euclidean distance i.e, the scaled squared distance between feature representations of content and combination images.

6.4 Calculating Style Loss

For style loss, first, compute the gram matrix. The Gram matrix is proportional to the covariances of corresponding sets of features. This tells which features are active together. This allows capturing information about style independent of content.

The style loss is the Frobenius norm of the difference between Gram matrices of style and combination images.

Examples

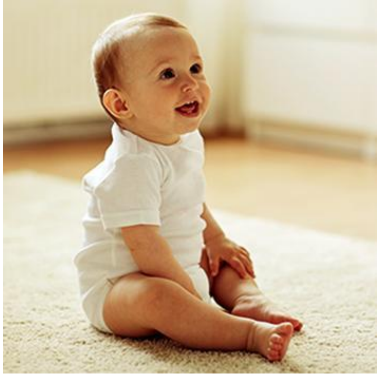

Fig. 5: Content Image

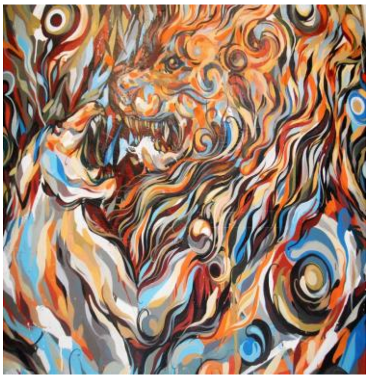

Fig. 6: Style Image

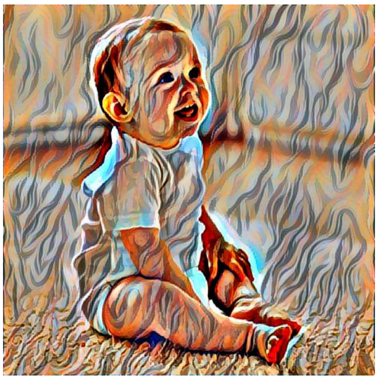

Fig. 7: Mixed Image of figure 5 and 6

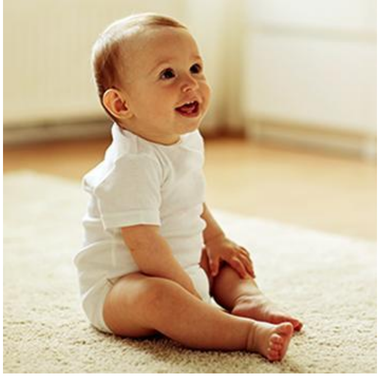

Fig. 8: Content Image

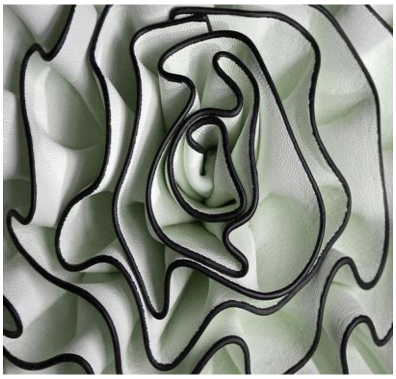

Fig. 9: Style Image

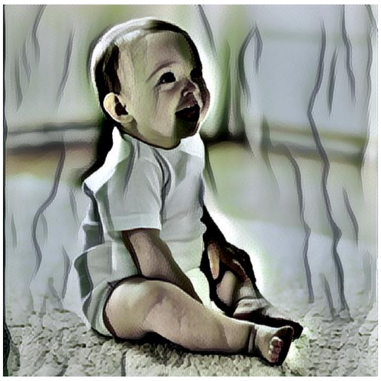

Fig. 10: Mixed image of figure 5 and 6

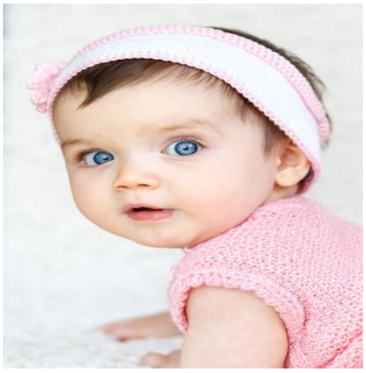

Fig. 11: Content Image

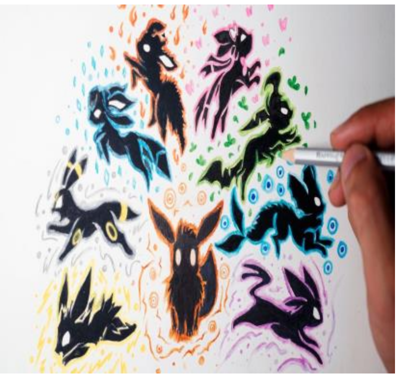

Fig. 12: Style Image

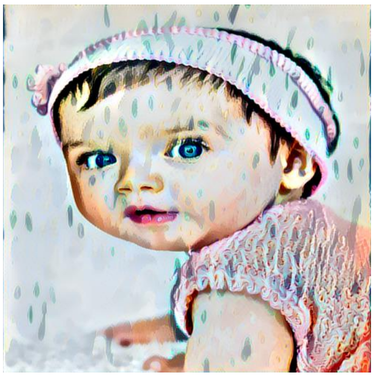

Fig. 13: Mixed image of figure 11 and 12

6.5 Obtaining the output image

Once the losses are calculated the output image is obtained by considering the layers which obtain the minimum loss. We then use L_BGFS algorithm to obtain the style image. A constant number of iterations (say 10 or 15) are performed on the combination image to get a more appealing look of the output image. During these iterations, the losses between the content image, style image with the combination image are minimized.

6.6 Effect of different layers in the Convolutional Neural Network

One more important factor in this process of image synthesis is to choose the layer on which the content and style representations. The style representation as mentioned above is a multi-scale representation that requires multiple neural network layers. The scale on which style is matched depends on the number and position of the neural network layer, which leads to different visual experiences.

7. RESULT

The overall output of this paper is that CNN was used for the representation of the content image and style image and both being completely separable. Then we have combined both the images to produce a perpetually meaningful image. We combined both content and style representations of source images and synthesised both images with equal texture to provide an output image.

8. REFERENCES

[1] Leon A Gatys, Alexander S. Ecker, Matthias Bethge "Image Style Transfer Using Convolutional Neural Networks "

[2] A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer at ACM, 2001.

[3] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin "Image analogies". In Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pages 327–340. ACM, 2001.

[4] N. Ashikhmin. "Fast texture transfer". IEEE Computer Graphics and Applications, July 2003.

[5] H. Lee, S. Seo, S. Ryoo, and K. Yoon., "Directional Texture Transfer". In Proceedings at 8th International Symposium on Non-Photorealistic Animation and Rendering, NPAR USA, 2010. ACM

[6] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution", Stanford University

[7] J. Portilla and E. P. Simoncelli. "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. International Journal of Computer Vision", Oct. 2000.