International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 06 | June 2019 www.irjet.net p-ISSN: 2395-0072

Concepts, Methods and Applications of Neural Style Transfer: A review Article

Aayush Kapur

B.Tech CSE, Vellore Institute of Technology, Vellore

Abstract – *This article encompasses the study of the deep image representations learned by Convolutional Neural networks and comment on the applicability for high level image synthesis and manipulation. One such high level problem deals with rendering the content of the image in a style of some different image. We will explore the concept of neural style transfer which has been identified in research papers earlier published. This article serves to provide an intuitive understanding of the same and explore different applications of this technique in a consolidated manner. Further the neural style transfer technique will be applied with different parameters and a survey of the results obtained was conducted.*

***Key Words*: Neural style transfer and applications, CNN, Neural art, Style rendering, Image Content separation**

1. INTRODUCTION

Neural style transfer is an optimisation technique that allows to copy the style from the style image and apply it over to the content image producing varied and often interesting results. This is a technique outlined in Leon A. Gatys' paper, A Neural Algorithm of Artistic Style. This functions on the principle of minimising the style loss and content loss in the final output as compared to the style and content images that is the final output should have the least loss in style in relation to the style image and least loss in content in relation to content image. The final image is first initialised randomly and slowly built over using the loss functions in CNNs. [1]

As such the area of art with neural network has been little explored, various approaches for trying to understand what neural networks make of the images fed to them have been tried. However there is still a gap in as to how the computer perceives the abstract concepts such as art and thus it largely points to "how does computer understand creativity or even in future will we be able to generate sufficiently creative machines to give ingenious solutions like we have come to expect from humans?". Will there ever come a time when we can expect out of the box thinking from machines we have built.

This article will provide the implementation of approach where we can separate out the semantic meaning of image from its content and paint a different image. It can be thought of as initial steps on long road to the computational creativity.

2. LITERATURE SURVEY

The concept of neural style transfer was introduced by Gatys, L., Ecker, A. and Bethge in their paper titled "A Neural Algorithm of Artistic Style" [1]. They laid the foundation of the method to allow stylistic features of an image to be combined with the content features of another image to create the resultant image. Inspired by the Neural Style transfer paper Li, Wang, Liu and Hou gave a new method of interpretation of neural style transfer by treating it like a domain adaptation problem[2]. It was shown by them theoretically that matching of Gram matrices of feature maps is same as minimizing the Max mean discrepancy with the second order polynomial kernel. It was proposed by them that the main idea behind neural style transfer is to match the feature distributions between the style images and the generated images.

Alex J Champandard in his 2016 paper titled "Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artwork" introduced a novel concept to augment generative architectures with semantic annotations, either by manually authoring pixel labels or using existing solutions for semantic segmentation leading to a content aware generative algorithm that offers meaningful control over the outcome. [3]

A new technique for data augmentation was proposed by Jackson, Abarghouei, Bonner, Toby Breckon, Obara in their 2018 paper "Style Augmentation: Data Augmentation via Style Randomization". It allowed to improve the robustness of convolutional neural networks over classification and regression tasks. During training, their style augmentation randomizes texture, contrast and color, while preserving shape and semantic content. They also investigated the effect of style augmentation on domain transfer tasks and found that data augmentation significantly improves robustness to domain shift. [4]

2.1 gaps identified

Those methods were based on the idea of training a network a network by training it on a very amount of data and based on the model formed the input was classified according to the model.

However, there was not so to say a proper separation and extraction of the style of the data, it was more of finding out the discernible features and distinguishing based on those features.

And even the closest technique to this one namely texture transfer which can also be utilized for style transfer, that approach too works on pixel level only, it sees the differences between the images based on the differences in their pixels and concludes the difference in style while in neural style transfer, the difference in style is found out based on the difference in feature spaces which are made after the input image has been understood by the layers in CNN.

The other methods utilized along the similar lines of this project have been limited to only very simple complexity use cases. The previous work which tried to separate out style from the content was based only on recognizing numbers/ digits from different hand writings for which a neural network was trained. It involved either simple characters or poses or images or small figures to be recognized. But it was never tried out to understand and separate whole stylistic pattern of a painting.

3. APPLICATIONS OF NEURAL STYLE TRANSFER

1. It uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images.

2. Offers a path forward to an algorithmic understanding of how humans create and perceive artistic imagery.

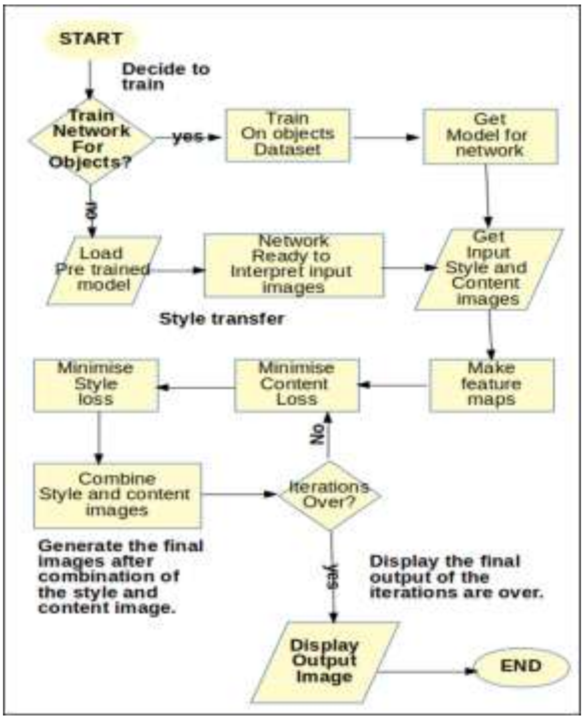
3. This technique can be extended to data augmentation. Style transfer augmentation is a very exciting data augmentation technique for learning a more diverse set of invariances and for domain transfer.

4. IMPLEMENTATION

4.1 steps followed

1. Build several loss functions to minimize the losses
2. Build the custom model with the dataset API
3. Use a model to learn feature maps to describe the content and style representation of our images.
4. Generate an iteratively updation based output image.
5. This iteration based process will be slow, may be try to speed up this process by making an untrained image transformation network.
6. Explore other applications depending feasibility.
7. Do a survey test on the generated results.

Figure-1: Flowchart for algorithm



source: self drawn

4.2 Algorithm:

1. The network is trained on dataset for different objects (Object detection dataset) or load a pre trained object detection based model for the neural network.

2. The content similarity formulae and style similarity were defined as written below.

3. for content similarity:
3.1 an initial random image was assumed
3.2 using standard error backpropagation, the initial random image was changed over to match the content image.
3.3 this took place by minimizing the content loss.

4. for style similarity:
4.1 the feature maps generated for layer are considered.
4.2 correlation between feature map at different layers is used to make gram matrices.
4.3 the mean squared distance is minimized between the original image gram matrix and the generated image gram matrix till it is within acceptable limits.

5. This leads to two separate components one depicting the content and the other depicting the style of the image, these components are combined using the formulae, by taking a new loss function

$$\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x})$$

with,
alpha being the weight factor for content part of the image
beta being the weight factor for style part of the image

Figure-2: content similarity

$$\mathcal{L}_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2 .$$

The derivative of this loss with respect to the activations in layer l equals

$$\frac{\partial \mathcal{L}_{content}}{\partial F_{ij}^{l}} = \begin{cases} \left(F^{l} - P^{l}\right)_{ij} & \text{if } F_{ij}^{l} > 0 \\ 0 & \text{if } F_{ij}^{l} < 0 . \end{cases}$$

where p, x are original and generated images respectively.

Figure-3: style similarity

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}\left(G_{ij}^{l} - A_{ij}^{l}\right)^2$$

$$\mathcal{L}_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l$$

Where a is the original image and x is the generated image.
A^l and G^l are style representations in the layer l.

E_l is the layer wise contribution and L_{style} is the total style loss.

For style similarity:
For a gram matrix G belonging to $R(N_l * M_l)$ we have G_{ij} as summation of ($F^l_{ik} * F^l_{jk}$),
Utilising the gradient descent for arriving at the resultant image which matches the stylistic representation of the input style image, starting from the white noise based image.

Do this by:
minimising the mean squared between the entries of the Gram matrix of the original image and the entries of the generated image.

4.3 Explanation of algorithm

The neural network has to be trained or a pre-trained model loaded so that the network gets the ability to make sense of what object is there in the image, essentially it will provide the network with the ability of understanding the image meaning to make a mathematical internal representation of the Input image for further processing.

The random image was slowly changed over to match the content image based on the responses it generates to a certain layer in the network, till the point the response is not within acceptable limits, using standard error backpropagation the image is modified.

Gradient descent is used to arrive at new image from the original white noise image, this new image will match the stylistic representation of the original image, the stylistic loss is minised by using the gram matrices.

4.4 Complexity analysis (time complexity)

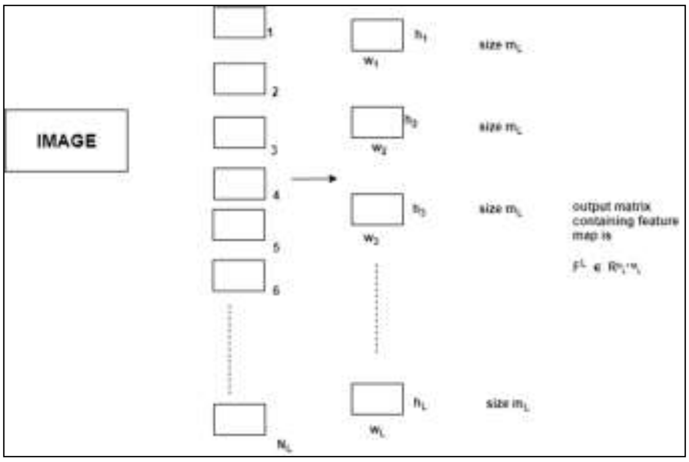
dependencies were imported
functions were declared

The style transfer function is run based on the number of iterations or "num_iter" given by the user, inside the style transfer function for each iteration the code goes over every layer in the model and all the style features for the gram matrices.

O(num_iter*model.layers+num_iter*style_features)

(space complexity)
O(num_iter*width*height)
O(num_iter*features*feature_width*feature_height)

Figure-4: breaking the image into smaller parts which in turn have their corresponding feature maps



source: self drawn

5. RESULTS

On running the style transfer with block5_conv2 features for content image and all the 5 layers for style image we get the following output, this has been done with 50 iterations over 5 different sets of input images.

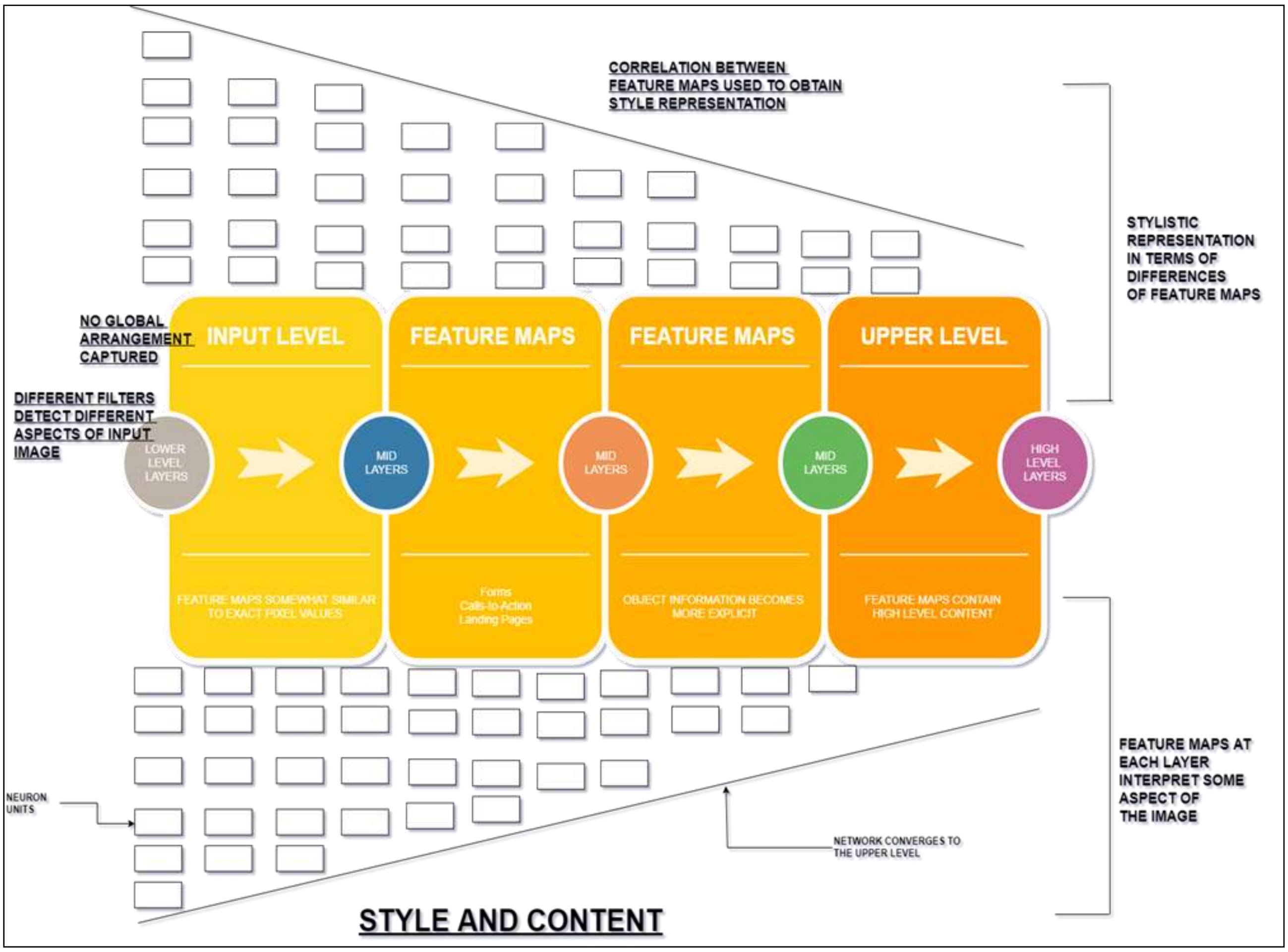
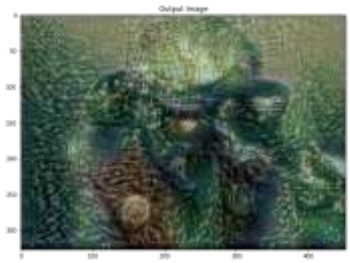


Figure-5: Relationship between the style and content
Source: Self drawn

Table-1: Output of style transfer with block5_conv2 features

No.	Input content image	Input style image	Output image
1			
2			

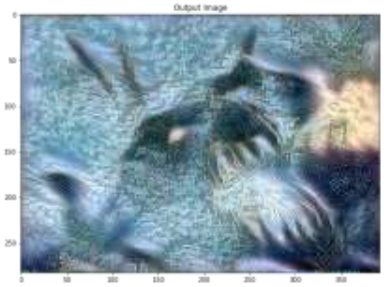
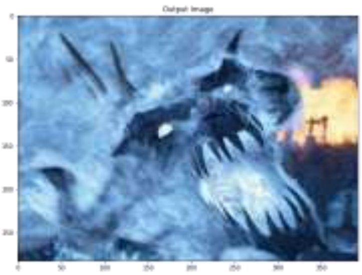
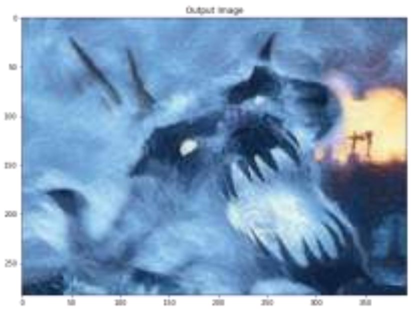
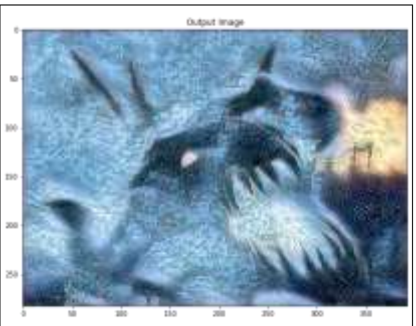
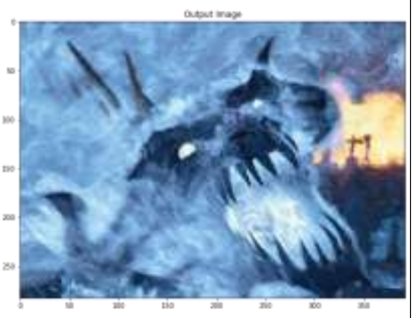
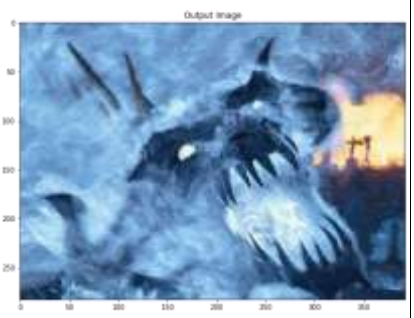
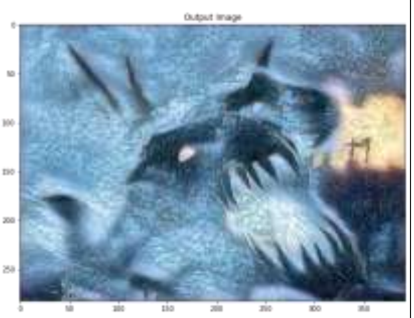
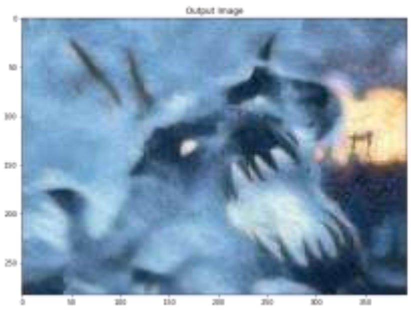
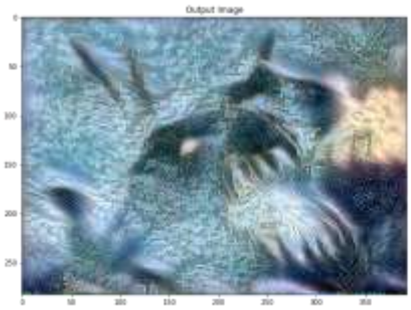
3			
4			
5			

Table-2: content and style feature comparisons

Content feature maps	Style layers	output
block1_conv 2	Layer 1	
block1_conv 2	Layers 1,2,3	

block1_conv 2	Layers 1,2,3,4, 5	
block3_conv 2	Layer 1	
block3_conv 2	Layers 1,2,3	
block3_conv 2	Layers 1,2,3,4, 5	
block5_conv 2	Layer 1	
block5_conv 2	Layers 1,2,3	
block5_conv 2	Layers 1,2,3,4, 5	

(The outputs have been generated over 20 iterations.)

5. INTERPRETATION

On changing the feature maps for the content images and for the style images to find out the behaviour of the output image. It was found that the higher layer feature maps for content images help to include the broad level details about the content while the lower layer feature maps include pixel by pixel almost identical information for image reconstruction. In the case of style images including all the five layers leads to better style output images.

In order to identify whether human subjects will be able to identify whether the images were produced by machine or think that the images have been made by man, a survey through Google forms was conducted. It led to mixed results with all the images produced by the machine being successful in making humans think otherwise more than 60% of the time.

6. CONCLUSIONS

This approach to neural networks has allowed us to understand how we might make the machines perceive the images in a way very similar to our natural visual system. It has shed on the fact that the neural networks can be used intelligently to use the style transfer mechanism over artistic images.

This can be extended over to be utilized for other forms of data as well. Another interesting application can be in the field of data augmentation where we might use style transfer to extend our data set. A very promising work for future in the field of style transfer can be machine creativity, though such systems are still in conceptual stages.

7. REFERENCES

[1] Gatys, L., Ecker, A. and Bethge, M. (2016). A Neural Algorithm of Artistic Style. Journal of Vision, 16(12), p.326.

[2] Li, Wang, Liu and Hou, "Demystifying Neural Style Transfer." Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, doi:10.24963/ijcai.2017/310

[3] Alex J Champandard, (2016), Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artwork

[4] Jackson, Abarghouei, Bonner, TobyBreckon, Obara, (2018), Style Augmentation: Data Augmentation via Style Randomization

[5] Towards Data Science. (2019). Neural Style Transfer Applications: Data Augmentation. [online] Available at: https://towardsdatascience.com/neural-style-transfer-applications-data-augmentation-43d1dc1aeecc [Accessed 8 Apr. 2019].

[6] sunshineatnoon.github.io. https://sunshineatnoon.github.io/posts/2017/05/blog-post-1/ (accessed June 7, 2019).

[7] /@purnasaigudikandula. "Artistic Neural Style Transfer with Pytorch." Medium, Medium, 22 Mar. 2019, medium.com/@purnasaigudikandula/artistic-neural-style-transfer-with-pytorch-1543e08cc38f.

[8] /@tensorflow. "Neural Style Transfer: Creating Art with Deep Learning Using Tf.keras and Eager Execution." Medium, TensorFlow, 27 Sept. 2018, medium.com/tensorflow/neural-style-transfer-creating-art-with-deep-learning-using-tf-keras-and-eager-execution-7d541ac31398.

[9] Luan, Fujun, et al. "Deep Photo Style Transfer." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, doi:10.1109/cvpr.2017.740

[10] Gatys, Leon A., et al. "Image Style Transfer Using Convolutional Neural Networks." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, doi:10.1109/cvpr.2016.265