Computationally Efficient Approaches for Image Style Transfer

Ram Krishna Pandey
Department of Electrical Engineering
Indian Institute of Science
Bangalore, India
ramp@iisc.ac.in

Samarjit Karmakar
Department of CSE
National Institute of Technology
Warangal, India
ksamarjit@student.nitw.ac.in

A G Ramakrishnan
Department of Electrical Engineering
Indian Institute of Science
Bangalore, India
agr@iisc.ac.in

***Abstract*—In this work, our focus is on developing fast image style transfer architectures for practical applications. We have proposed three modifications to the architecture of a recent, real-time, artistic style transfer technique to make it computationally more efficient. We have proposed the use of depth-wise separable convolution (DepSep) in place of convolution and nearest neighbor (NN) interpolation in place of transposed convolution. We have also explored the concatenation of nearest neighbour and bilinear (Bil) interpolations in place of transposed convolution. The stylized images from the modified architectures are perceptually similar in quality to those from the original architecture. The decrease in the computational complexity of our architectures is validated by the decrease in the testing time by 26.1%, 39.1%, and 57.1%, respectively, for DepSep, DepSep-NN-Bil and DepSep-NN modifications. Working with another architecture, we have examined how the quality of the stylized reconstruction changes with the change of the loss function to be minimized.**

***Keywords*–Image style transfer, depth-wise separable convolution, nearest neighbor interpolation, art, separable representation, computational complexity.**

I. Introduction

Art is possibly perceived by humans different from how animals do. From time immemorial, people have been fascinated by art. Art reflects, transmits and shapes our culture. History suggests that art is made for the production of "beauty". Human beings are emotionally attached to beauty, and perhaps this is the origin of art. Humans have the ability to distinguish between the texture, content, and style of images. With a complex interplay between these features, they can compose fine works of art. However, this task is difficult for a machine to perform. Recently, convolutional neural networks (CNNs) have shown promising results in computer vision tasks like object recognition. The ability of CNNs to capture the information in the hierarchy of representation, from low level (the raw information of pixels) to high level (information about the content of the image), led the explorers of artistic neural style transfer to propose algorithms for creating artistic images [1].

The task of artistic style transfer is interesting in the sense that the representations of the content and the style in the convolutional neural network are separable. Therefore, we can superimpose the texture or style of an artistic work (image) onto the content of a natural scene image. Artistic style transfer aims at superimposing the artistic style of an artist's artwork on to an image with the help of a learning algorithm that requires a good amount of computational power to train such an architecture.

II. Literature Review

Gatys et al [2] was the first to use the power of CNNs to reproduce famous works of art on natural images. They have shown that the content of an image are the filter responses at the deeper layers of the VGG network [3] trained for image classification task. Since the network has been trained on ImageNet dataset [4] for image classification task, this network will also capture the content of an image at the deeper layers of the network. The style of an image is the linear combination of the Gram matrices of the feature maps taken at different layers of the same network. Gram matrix captures the correlation of the feature maps at a layer.

Let the feature maps of the l^{th} layer of the network be $F_l(S)$, where S is the style image. Then the Gram-based representation of this layer is given as:

$$G(F_l(S)) = [F_l(S)][F_l(S]^T \tag{1}$$

The weighted combination of the Gram-based representations of multiple layers of the network is considered for a representation of the style of an image. Assuming we take Gram-based representations for the layers $l = 1, 2, ...n$, the style representation (Sty) for the style image S can be given as,

$$Sty = \sum_{l=1}^{n} w_l \times G(F_l(S)) \tag{2}$$

The style loss is given by

$$L_s = \sum_{l=1}^{n} w_l \times ||G(F_l(Y)) - G(F_l(S))||_F \tag{3}$$

Here, w_l is the weight assigned to the l^{th} layer, Y is the desired image and S is the style image.

The content representation is simply the feature map of the k^{th} layer of the network. The deeper layers of the network are considered, since they capture more of the higher-level features of the image (which represent the content), than the shallow layers.

The weighted combination of the content representations of multiple layers of the network is taken as the representation of the content of an image. Assuming we include content representations for the layers $k = 1, 2, ...n$, the content representation (Con) for the content image N is given by,

$$Con = \sum_{k=1}^{n} w_k \times F_k(N) \tag{4}$$

Here, w_k is the weight assigned to the k^{th} layer, Y is the desired image and N is the content image.

Generally, we consider only the higher layers in the network for content. In [2], the authors have taken the fourth layer (k = 4) for content representation and the first to the fifth (l = 1 to 5) for style representation, with $w_k = 1$ and $w_l = 1$. The weights in all other layers are 0.

The objective is to find $\hat{Y}$, which satisfies the following:

$$\hat{Y} = \underset{Y}{\operatorname{argmin}}(\alpha L_c + \beta L_s) \tag{5}$$

where L_c and L_s are the content and style losses, respectively. This is a slow parametric method to obtain the desired image Y, since we need to iteratively optimize the objective function for each $\{N, S\}$ pair. The content loss can be given as,

$$L_c = \sum_{k=1}^{n} w_k \times ||F_k(Y) - F_k(N)||_F \tag{6}$$

Johnson et al [5] proposed a real-time method for fast neural style transfer, where they train an image transformation network to find a non-linear function that maps the content image to the desired output image for a given style image, on which the network has already been trained. They use a loss network pre-trained for image classification task to define perceptual loss functions that measure perceptual differences in content and style between the images. The loss network remains fixed during the training process.

The image transformation network is a deep residual convolutional neural network with parameters λ. It consists of three convolution layers, followed by five residual layers (RL) and two transposed convolution layers. The residual networks make it easy to find the identity function and improve the gradient flow.

The loss network is a pre-trained VGG network on image classification task. They obtain the loss in a way similar to that of Gatys et al [2], where they take the summation of the feature responses of k layers of the network along with the Gram-based representations of l layers. Along with the content loss and style loss, they also take into account the total variation loss (L_{tv}), making use of the total variation regularizer, which encourages spatial smoothness in the output image. The total loss, expressed as,

$$L_{total} = \alpha L_c + \beta L_s + \gamma L_{tv} \tag{7}$$

is minimised by backpropagating using stochastic gradient descent optimizer.

This is a computationally faster method for neural style transfer, where they train a feedforward network on multiple content images and the image of a single stylized artwork. While testing, the output image can be obtained in a single forward pass, since the network has been trained to stylize any natural image with a single stylized artwork.

III. Problem Definition and formulation

Charbonnier loss function [6] has been reported to preserve more content, when used in place of MSE, for the computation of the content loss. As an explorative study, we have used Charbonnier loss function instead of MSE for one or both of content and style losses in the method proposed by Gatys et al [2], and studied the quality of the resulting stylized images.

As the main work of this paper, we aim to further improve the architecture of Johnson et al [5] to make it computationally more efficient, without decreasing the perceptual quality of the stylized output image. The problem of neural style transfer can be mathematically formulated as follows. Suppose N is the given natural scene image, whose content we want to preserve and S is an artistic work (style image), whose texture or style we want to superimpose on the content image, N. Let the desired image be Y. We require $\hat{Y}$, which minimizes the combined content-style loss function:

$$\hat{Y} = \underset{Y}{\operatorname{argmin}}(\alpha \times ||C(Y) - C(N)||_F + \beta \times ||X(Y) - X(S)||_F) \tag{8}$$

Here, the operators C and X extract the content of the image and the style of the image, respectively.

IV. Contributions of this work

Our main contributions are as follows:

- We have shown that using Charbonnier as the content loss function results in the preservation of more content in the output image of [2].
- We have improved the computational efficiency of the already fast method proposed by Johnson et al. [5] by reducing the number of parameters using depth-wise separable convolution layers instead of regular convolution layers in the image transformation network. Further, in two separate experiments, we have replaced the transposed convolution layer with (i) nearest neighbour upsampling with gradient flow, as shown in Fig. 3 or (ii) the concatenation of nearest neighbour and bilinear upsampling, as shown in Fig. 4.

In Xception model, Francois Chollet [7] shows the working of depthwise separable convolution in deep learning architectures. It results in a huge reduction in the number of parameters, while the model performance is similar. Nearest neighbour interpolation is a classical, computationally efficient technique for upsampling images. It repeats a pixel four times to obtain the output image. The output images shown in Fig. 7 and the testing times reported in Table I validate the above statements on the output quality and the computational efficiency of our proposed modifications.

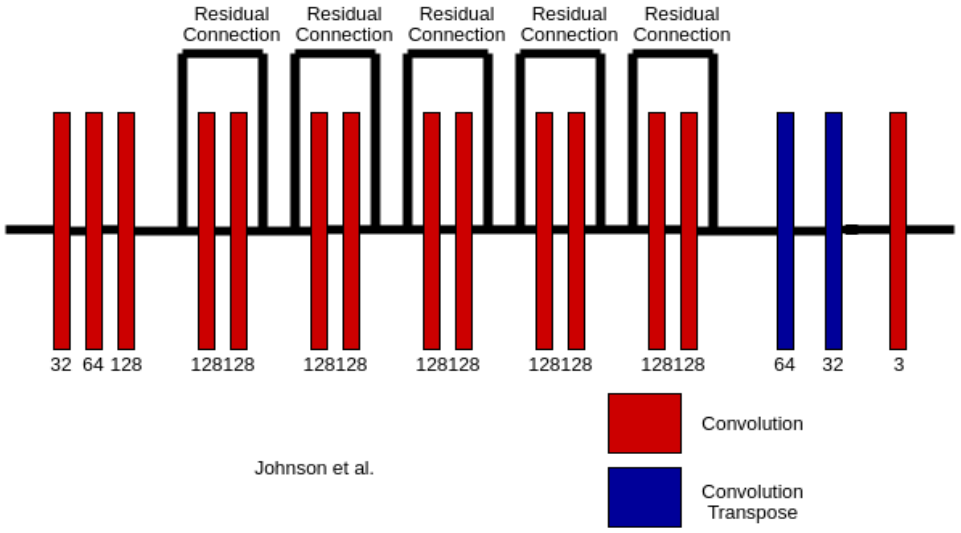


Fig. 1. The original image transformation architecture proposed by Johnson et al [5] for real-time neural style transfer. It consists of convolutional layers (CL), residual blocks (RB) and transposed convolutional layers (TCL). The first CL has 32 filters of kernel size 9x9 and stride 1; the second and third, 64 and 128 filters of kernel size 3x3 and stride 2. Each RB contains CLs having 128 filters of kernel size 3x3 and stride 1. The two TCLs have 64 and 32 filters of kernel size 3x3 and stride 1/2. The last CL has 3 filters of kernel size 9x9 and stride 1. 'ReLU' non-linearity has been applied to all the layers except the last CL.

V. Our approaches for faster style transfer

We have suggested modifications for faster style transfer on the image transformation network proposed by Johnson et al [5]. In the first experiment, all the convolutional layers have been replaced with depth-wise separable convolutional layers. This DepSep architecture is illustrated in Fig. 2. In the second experiment, in addition, we have replaced transposed convolution layers (TCL) with nearest neighbour upsampling. This DepSep-NN architecture is illustrated in Fig. 3. In the final experiment, the TCLs are replaced by the concatenation of nearest neighbour and bilinear upsampling [8]. Figure 4 illustrates this DepSep-NN-Bil architecture.

We have used $\alpha = 7.5$, $\beta = 10^2$ and $\gamma = 2 \times 10^2$ for the loss function in eqn. 7. We have implemented the proposed architectures as above and trained the networks on the Microsoft COCO dataset [10]. We have used Adam optimizer [11] to minimise the loss, with learning rate as 0.001, β_1 as 0.999 and β_2 as 0.99. Instance normalization has been applied after every depthwise separable convolutional layer, as suggested by Ulyanov et al [12].

The models have been implemented in the Tensorflow [13] deep learning framework and trained on Nvidia Titan X GPU, which takes roughly 18-20 hours for 2 epochs (each epoch having 20,650 iterations).

VI. Results and Discussion

Figure 6 shows the different stylized outputs reconstructed by [2] when the losses L_c and L_s are varied: (a) when both the losses are MSE; (b) when both the losses are Charbonnier; (c) and (d) when content loss is MSE and style loss is Charbonnier and vice-versa.

Figure 7 shows the qualitative results of our experiments to reduce the computation complexity of [5]. The testing times for the models are listed in Table I. We see that replacing convolution layers with depth-wise separable convolution layers has led to 26.1% decrease in testing time. Further, replacing transposed convolution with nearest neighbour upsampling has led to 57.1% decrease in the testing time, while replacing it with the concatenation of nearest neighbour and bilinear upsampling has led to 39.1% decrease. Figure 7 shows that the images produced by all the models are similar, with negligible change to the perceptual quality.

TABLE I: Comparison of the testing time taken by the original model by Johnson et al. [5] with those of the modified architectures proposed by us and shown in Figs. 2, 3 and 4.

Details	Architecture	Testing time in sec.	% decrease in time
Fig. 1	Johnson et al. [5]	1.33	-
Fig. 2	Modified 1: DepSep	0.97	26.1
Fig. 4	Modified 2: DepSep-NN-Bil	0.81	39.1
Fig. 3	Modified 3: DepSep-NN	0.57	57.1

Note: We asked several people to rate the quality of the reconstructed, stylized images and most of them rated the outputs of DepSep-NN-Bil to be the best.

VII. Conclusion

In the first part of this paper, we have explored various models by changing the loss function, such as the mean square, MSCE [14] and Charbonnier in the original implementation proposed by Gatys et al. [2] and have shown their results. Since there is no metric to decide which of the losses or their combination is better, we have left it for the viewer to decide. This will help researchers working in similar areas to use the proper loss function (or a combination) based on their requirement. Secondly, we have further improved the already fast architecture proposed by Johnson et al. [5] using depth-wise separable convolution, nearest neighbour interpolation

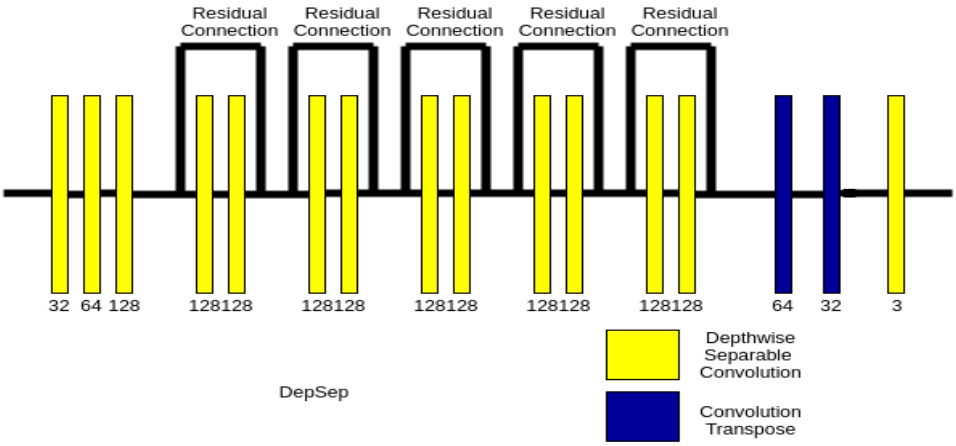


Fig. 2. 'DepSep' architecture: an improvement over the image transformation network of Johnson et al. [5]. The convolutional layers have been replaced with depth-wise separable convolutional layers, with channel multiplier = 4.

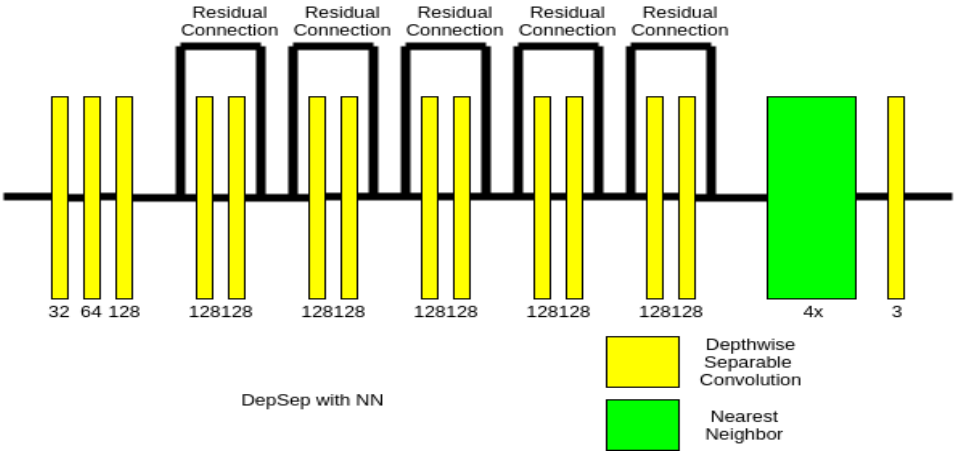


Fig. 3. 'DepSep-NN' architecture, an improvement over DepSep, by the replacement of transposed convolutional layers with nearest neighbor up-sampling.

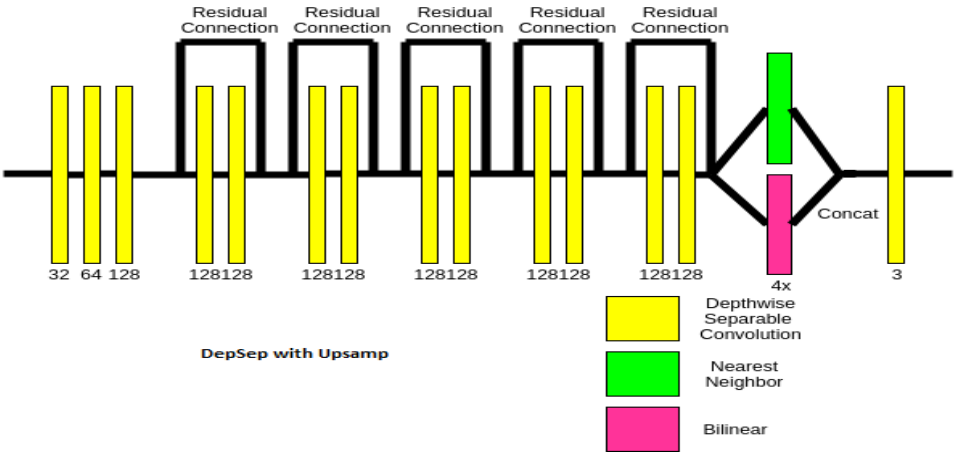
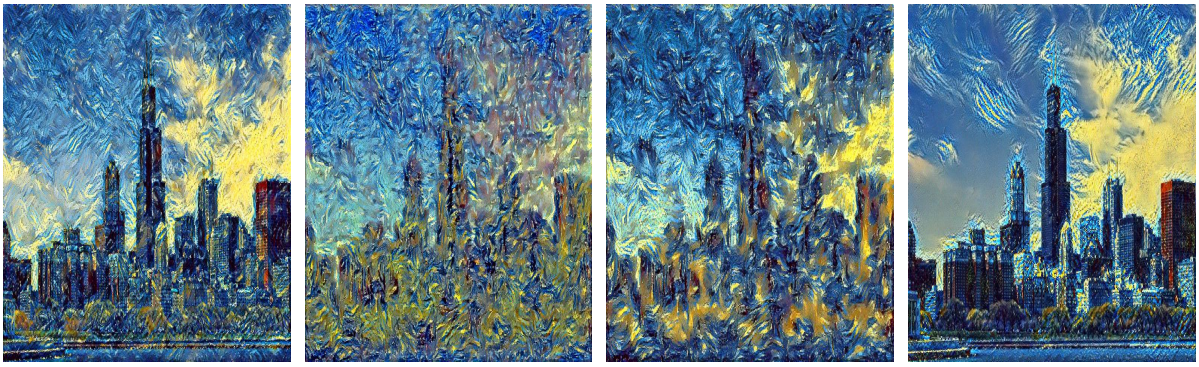


Fig. 4. 'DepSep-NN-Bil' architecture, an improvement over DepSep, by replacing transposed convolutional layers with the concatenation of nearest neighbor and bilinear up-sampling.

(a) Chicago skyline (content image) **(b)** Udnie (style image) **(c)** The Starry Night (style image)

Fig. 5. Each model has been tuned to a specific style image. For each style image, we train a model on a dataset of images. The content image is fed into the model while testing. (a) A natural scene image - Chicago skyline. (b) "Udnie" image by Francis Picabia, 1913. (c) "The Starry Night" image by Vincent van Gogh, 1889.



(a) Content loss - MSE Style loss - MSE **(b)** Content loss - Char Style loss - Char **(c)** Content loss - MSE Style loss - Char **(d)** Content loss - Char Style loss - MSE

Fig. 6. A qualitative analysis of the effect of different loss functions on the stylized output of the method proposed by Gatys et al. [2]. Chicago skyline is the content image used, and Starry night, the style image. We have varied the content and style losses with MSE and Charbonnier (Char) loss functions. The details are given in the subcaptions above.

(a) Johnson et al **(b)** DepSep **(c)** DepSep-NN-Bil **(d)** DepSep-NN

Fig. 7. The results obtained from the various architectures proposed, for Chicago skyline as the content image and Udnie as the style image. (a) The stylized output of [5] proposed by Johnson et al. (b) Output of our DepSep model shown in Fig. 2. (c) Output of our DepSep-NN-Bil model shown in Fig. 4. (d) Output of our DepSep-NN model shown in Fig. 3.

and the combination of nearest neighbour and bilinear interpolation [8] [9] and provided three computationally efficient architectures. These proposed architectures can reconstruct the stylized images almost similar (if not better) in perceptual quality to that reconstructed by the original model in [5]. Our proposed architectures show significant improvements in testing times of 26.1%, 39.1%, and 57.1%, respectively. Thus, our modified architectures can obtain stylized images faster and this fun consumer photo technology has good industrial applications such as in Prisma, Adobe and deepart.io photo editors.

REFERENCES

[1] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song, "Neural Style Transfer: A Review," arXiv:1705.04058, 2017.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in Proc. IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR 2016), 2016.

[3] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Proc. Int. Conf. Learning Representations (ICLR 2015), 2015.

[4] J. Deng, W. Dong, R. Socher, Li-Jia Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," Proc. IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR), 2009.

[5] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution, Proc. European Conf. Comput. Vis. (ECCV), 2016.

[6] Barron, and Jonathan T, "A more general robust loss function," arXiv:1701.03077, 2017.

[7] Franois Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Proc. IEEE Conf. Comput. Vis. and Pattern Recog. (CVPR), 2017.

[8] Ram Krishna Pandey, and A G Ramakrishnan, "A hybrid approach of interpolations and CNN to obtain super-resolution," arXiv:1805.09400, 2018.

[9] R. K. Pandey, S. R. Maiya and A. G. Ramakrishnan, "A new approach for upscaling document images for improving their quality," 14th IEEE India Council Int. Conf. (INDICON), Roorkee, pp. 1-6, 2017. doi: 10.1109/INDICON.2017.8487796

[10] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C.L., "Microsoft coco: Common objects in context," Proc. European Conf. Comput. Vis. ECCV 2014.

[11] Kinga, D., and J. Ba, "Adam: A method for stochastic optimization," Proc. Int. Conf. Learning Representations (ICLR), 2015.

[12] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky, "Instance normalization: The missing ingredient for fast stylization," arXiv:1607.08022, 2016.

[13] Abadi, Martn, et al., "Tensorflow: a system for large-scale machine learning," In OSDI, vol. 16, pp. 265-283, 2016.

[14] R. K. Pandey, N. Saha, S. Karmakar, and A. G. Ramakrishnan, "MSCE: An Edge-preserving Robust Loss Function for Improving Super-resolution Algorithms," 25th Int. Conf. on Neural Information Processing, Siem Reap, Cambodia, 2018.