ICIC Express Letters
Volume **16**, Number **7**, July **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **695–703**

MULTI TENSOR INJECTION (MUTI) FOR ADJUSTING TEXTURE INTENSITY AND IMAGE COLOR IN FAST NEURAL STYLE TRANSFER

Hidayaturrahman[1] and Dwi Hendratmo Widyantoro[2]

[1]School of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
hidayaturrahman@binus.ac.id

[2]School of Electrical Engineering and Informatics
Institut Teknologi Bandung
Jl. Ganesa No. 10, Lb. Siliwangi, Kecamatan Coblong, Kota Bandung 40132, Jawa Barat, Indonesia
dwi@stei.itb.ac.id

Received July 2021; accepted October 2021

Abstract. *Style transfer is an activity to move styles from one image to another by preserving the content of the target image. Initially, style transfer was performed slow. Then several studies were carried out to make the style transfer process faster, which is then known as fast neural style transfer. However, this technique still has a weakness where the style level, in this case, referred to texture and color, depends on the targets that have been set when training the model. Therefore, we propose an approach called Multi Tensor Injection (MuTI) to control texture and color intensity while doing style transfers. To measure the success rate of this method, we compare the level of image similarity with its content and style by using the loss function used to train the prior model. We found that this method can control the style transfer process well and preserve the existing content in the image.*

Keywords: Neural style transfer, Fast neural style transfer, Arbitrary style transfer, Configurable deep learning model

1. **Introduction.** Neural style transfer or more commonly referred as style transfer is an approach to transferring color and texture characteristics from an image to another. In their publication, Gatys et al. proposed an approach to transferring texture and color distribution of an image to other images [1]. This method applies gradient descent from features contained in the style image to being transferred into content image. One of the results in their research was combining photos from the "Tuebingen Neckarfront" building with "Starry Night" paintings made by Vincent Van Gogh.

In its development, transfer style is widely used to render artistic images. Images from the real world is combined with a variety of works art with various types of flow. Some techniques are developed such as texture transfers [2], color transfer [3], portrait stylization [4], and video stylization [5]. The images produced from this method often produce results that have shape distortion. However, this is not something that is needed to be concerned in artistic images.

The initial idea in doing a transfer style is slow because there must be enough iterations so the error value becomes small and the image with the desired style is successfully formed. Jonshon et al. introduced an approach to encoding styles from an image into a network that could be used to produce images with that style [5]. This method cuts the

DOI: 10.24507/icicel.16.07.695

time to style transfers very well. However, this method is limited in the type of style that can be transferred.

Arbitrary style transfers [6] then use Adaptive Instance Normalization (AdaIN) to solve this problem. AdaIN uses an encoder-decoder approach to do the style transfers. So this approach then makes it possible to do style transfers quickly while the image style can be changed. However, there are problems in this method, namely the weight value for content and style is always constant. To get a new model with different weights requires a retraining process for the model and that requires a long time.

Therefore in this research, Mutual Tensor Injection (MuTI) is introduced, an approach to developing style transfer process. This technique utilizes the intermediate output layer of the feature encoder, multiplies it with a certain weight, and then injects it into the intermediate layer of the feature decoder. So, it is possible to manipulate the content and style weights of the resulting image without having additional model training.

This research paper is divided into five chapters. Chapter 1 is introduction which is dealing with background of the study, and it presents the current technique of style transfer and its problem. Chapter 2 is showing related work about current style transfer method. Chapter 3 is presenting the architecture to solve stated problem and evaluation techniques to measure the proposed architecture. Chapter 4 is talking about the experiment and result. Finally, Chapter 5 comes up with final conclusion.

2. **Related Work.**

When viewed based on its speed to do style transfers, we can classify into two approaches in performing neural style transfers, namely "Slow" Neural Methods and "Fast" Neural Methods [7]. The neural style transfer method proposed for the first time by Gatys et al. [1] is classified as "Slow" Neural Method. All methods that use a similar approach to that of Gatys et al., performing online optimization on images can be categorized into the "Slow" category. Other methods that can be classified in this category are the methods by [3], Selim et al. [4], and Zhi et al. [8].

While the method that is categorized as "Fast" Neural Method is a method that performs optimization on certain models so that the model can then be used to produce images with the desired style. The several methods categorized as "Fast" Neural Methods are those by Huang and Belongie [6], Li and Wand [9], and Shen et al. [10].

Huang and Belongie [6] proposed a framework involving a layer called the Adaptive Instance Normalization (AdaIN) layer. This layer functions to match the average value and co-variance of feature maps making it possible to style transfers with a single feed forward process. This method is able to provide the results of style transfers quickly and is also not limited to certain styles when running them.

3. **Multi Tensor Injection (MuTI).**

Multi Tensor Injection (MuTI) is a method that is able to inject several tensor values at once into the intermediate layer in a model. In this study, MuTI is used to manipulate the values in the fast neural style transfer model. This is intended to provide flexibility for users who do style transfers to weigh the value of the content and style generated by the model.

3.1. **Architecture.**

The model used in this study is the development of the existing model in the Arbitrary Style Transfer research [6]. The modification is to add a new object called Tensor Injector. This instance is added to do Multi Tensor Injection (MuTI) by receiving encoding features from the decoder layers and then inject it into the decoder layers. This injection process can be done with several techniques, namely direct injection and mean injection. In general, Figure 1 shows the architecture of the model for conducting MuTI. Furthermore, according to given configuration, the Tensor Injector will inject the value into the intermediate layer that is on the decoder with a certain weight. This will be a control for intensity of texture and color for the resulting image.

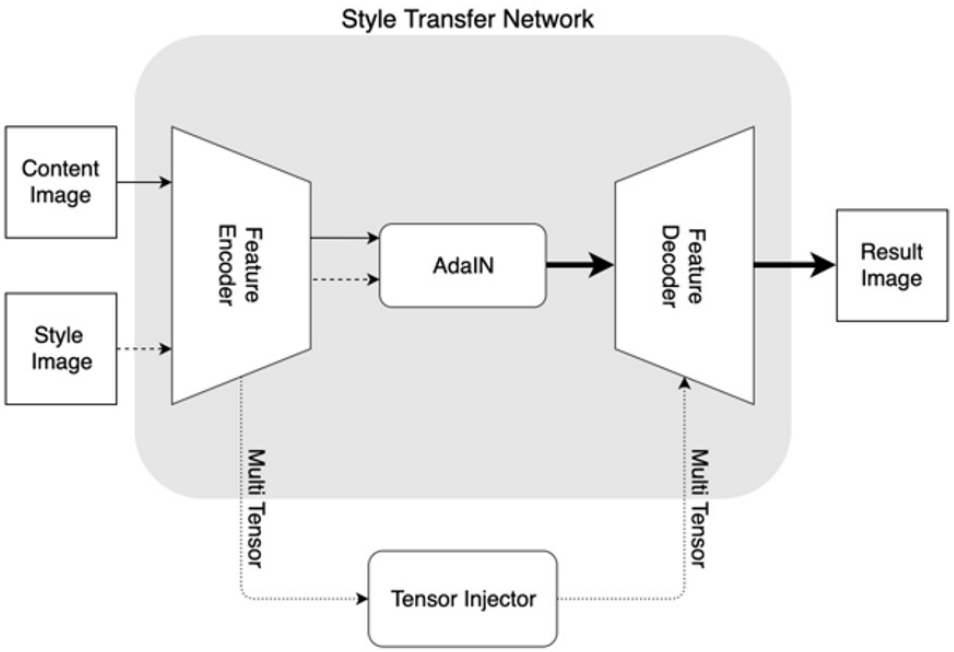

FIGURE 1. Model architecture that uses MuTI in performing fast neural style transfer

3.1.1. *Feature encoder.* The feature encoder is the part that will encode the content image and style image. The results of the encoding will then be input for the AdaIN process. In addition, the encoder feature will also produce intermediate encoding values which will be used when using MuTI. Feature encoder model was adopted from the VGG-19 model. The layers used in the encoder are conv1_1 to conv4_1. So in the encoder there are nine convolution layers followed by the relay activation function and three pooling layers. Following are the details of the feature encoder model.

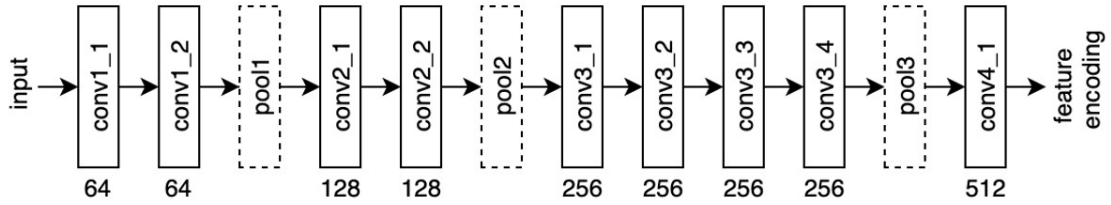

FIGURE 2. Architectural feature encoder model

3.1.2. *Adaptive Instances Normalization (AdaIN).* AdaIN has duty to receive feature encoding from image content and style images generated by the feature encoder and then generate input for the feature decoder. AdaIN is a development of Instance Normalization (IN) [11]. In general, each time a convolution is carried out, afterwards a Batch Normalization (BN) will be carried out so that the learning process can run more easily [12].

$$AdaIN(c,s) = \sigma(s)\left(\frac{c-\mu(c)}{\sigma(c)}\right) + \mu(s) \tag{1}$$

AdaIN does a style transfer of feature space by transferring statistics from its features, specifically through mean μ and variance σ values of content c and style s images. AdaIN has the same function as the swap layer style [13] but with much lower computing costs, it can even be said to be almost non-existent.

3.1.3. *Feature decoder.* After normalizing the feature encoding value through AdaIN, the encoding feature will be decoded using the feature decoder. This model is generally the opposite of the encoder model in which all pooling layers are replaced with nearest up sampling to reduce the effects of the chessboard. Because normalization has been carried out at the AdaIN stage, this model does not need to be normalized anymore.

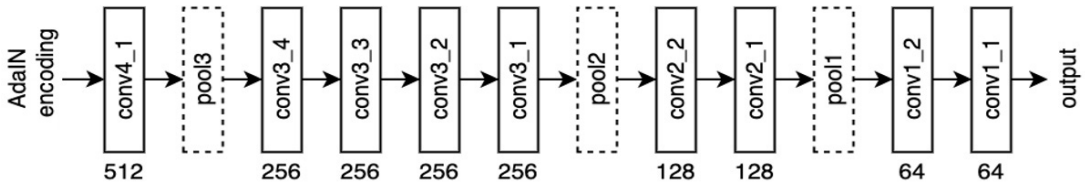

FIGURE 3. Architectural feature decoder model

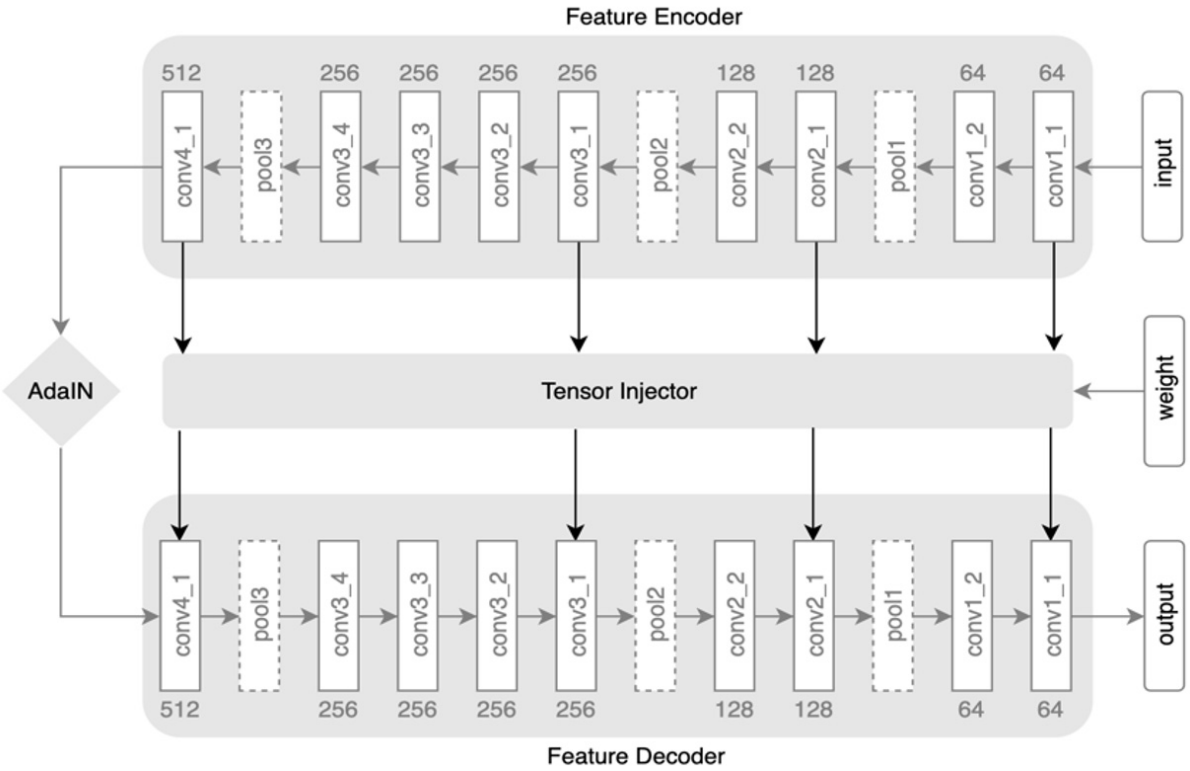

FIGURE 4. MuTI mechanisms with tensor injectors

3.1.4. *Tensor Injector.* Tensor Injector is an object that is used to do Multi Tensor Injection (MuTI) with input taken from intermediate output on the E_l encoder feature and injects it into the intermediate layer of the D_l feature decoder where l is the corresponding size layer in the encoder and decoder model. In general, the MuTI mechanism can be seen in Figure 4. MuTI is implemented in the feature decoder before up sampling is performed. So if the decoder model is seen as a combination of several blocks b where at the end of each block an up sampling is done with the input from the previous block D_{b-1} and the input in the first block is the output of the value AdaIN a, then the form of the decoding function to produce images can be written as follows

$$D_b = decode(D_{b-1}) \tag{2}$$

$$D_0 = a \tag{3}$$

So for MuTI, where the tensor of the intermediate output layer of the encoder E model is injected with w weight before up sampling, its function can be written in the following form

$$D_b = decode(MuTI(D_{b-1}, E_{b-1}, w)) \tag{4}$$

MuTI can be done with two approaches namely direct injection and mean injection. Direct injection is done by directly adding a value to each encoding feature m in layer l decoder with the result that the encoder value corresponds to the weight given. In the formal form, MuTI with the direct injection approach can be written as follows

$$MuTI\left(D_l^m, E_l^m, w\right) = \sum_{i=1}^{m} D_l^i + wE_l^i \tag{5}$$

As for MuTI with the mean injection approach, the value of each feature map is shifted based on the average value of each feature encoding m that is in layer l and then multiplied by the weight input. So in formal form MuTI with the mean injection approach can be written as follows

$$MuTI\left(D_l^m, E_l^m, w\right) = \sum_{i=1}^{m} D_l^i + w\mu\left(E_l^i\right) \tag{6}$$

3.2. **Evaluation techniques.** To assess quality, two approaches were used to assess the quality of the transfer style images using MuTI. The first approach is assessed against the visual form of the resulting image. In the first approach, the integrity of the image, the color distribution and texture of the resulting image will be observed. While the second approach is seen from the value of the style loss and content loss of the image. $\mathcal{L}_c$ content loss is the Euclidean distance of the E_c content image encoding feature and the E_o output image encoding feature. While the $\mathcal{L}_s$ style loss is a similar statistical value as used in Instance Normalization (IN) [9] and is used as a loss function in arbitrary style transfer models. In formal form content loss and style can be written as follows

$$\mathcal{L}_c = \|E_c - E_o\|_2 \tag{7}$$

$$\mathcal{L}_s = \sum_{i=1}^{L} \left\|\mu\left(E_s^i\right) - \mu\left(E_0^i\right)\right\|_2 + \sum_{i=1}^{L} \left\|\sigma\left(E_s^i\right) - \sigma\left(E_0^i\right)\right\|_2 \tag{8}$$

4. **Experiments and Analysis.** To find out the effectiveness of the development of this model, several scenarios are carried out in doing the transfer style. For the first scenario, style transfer is performed by direct injection. The next scenario is to do a mean injection using the image style encoding feature.

Content Injection. In doing content injection, direct injection is performed on intermediate values in the decoder model, where the injected value is a feature encoding of the content image. In this experiment, the weight values used ranged from 0.5 to -0.5. Figure 5 shows some of the images produced along with their weights. It can be seen, by directly doing content injection, it affects the color of the result images.

Style Injection. Style injection is carried out with several approaches. These approaches include direct injection and mean injection. Each of these approaches has different characteristics.

Style Injection – Direct Injection. In the direct injection approach for the case of style injection, there are several obstacles. Unlike the case of content injection, the size of the image style, spatially, is not always the same as the size of the image content. This makes direct injection of intermediate values in the encoder model impossible because images of different sizes will produce different amounts of encoding features. Therefore, resize the style image to the size of the content image. As seen in Figure 6, the resulting image is still produced by reflecting the style obtained from the style image, but the style image

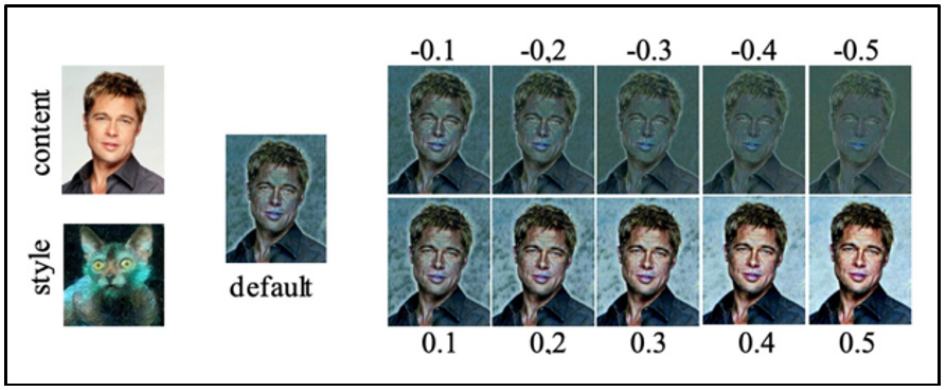

FIGURE 5. MuTI uses direct "content" injection with weights from −0.5 to 0.5.

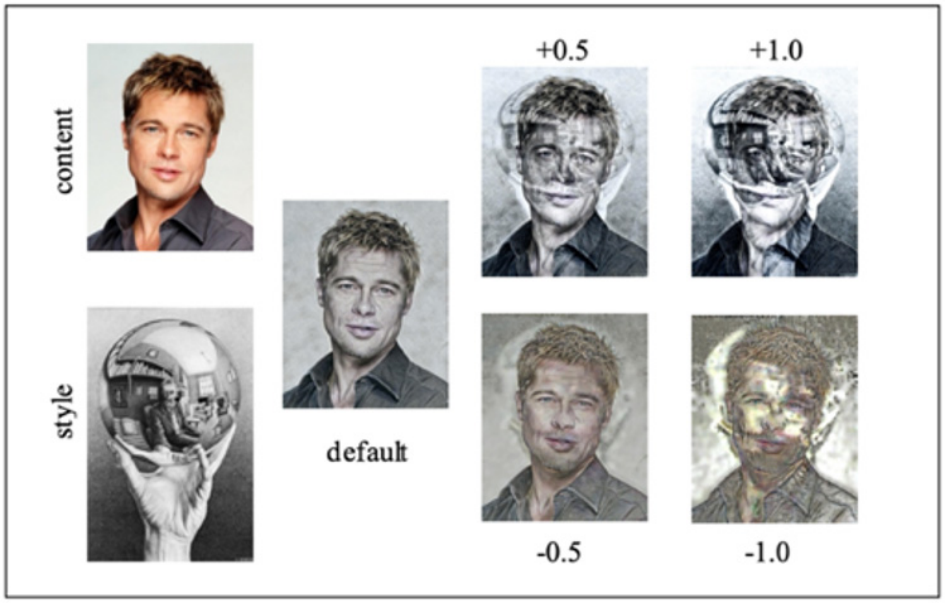

FIGURE 6. MuTI uses direct "content" injection with weights from −1 to +1.

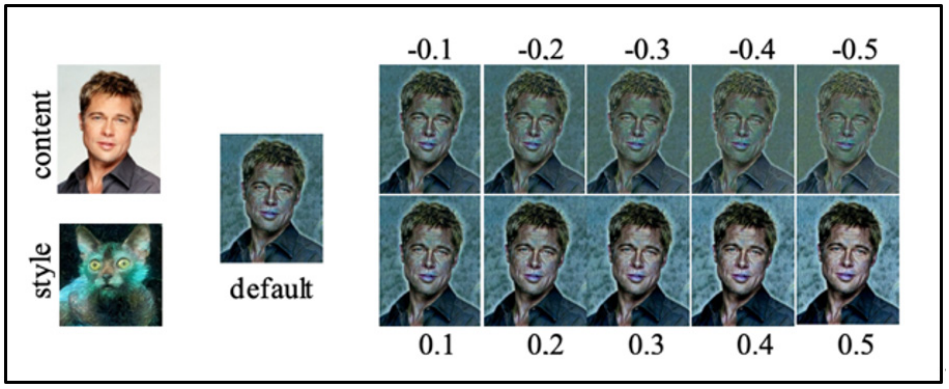

FIGURE 7. MuTI uses mean "style" injection weights from −0.5 to 0.5.

also appears in the resulting image so that it creates a super impose effect for positive weights.

Style Injection – Mean Injection. The next approach used to do style injection is to do mean injection. This approach is based on the same spirit as the AdaIN technique that is used to perform feature mapping when doing training. The resulting image can be seen in Figure 7. In the picture of this result, it can be seen that this technique does not

damage the image when tensor injection is done as happens when doing direct injection. The resulting image is quite good where the image style and content can still be well represented.

Analysis. To assess quality, two approaches were used to assess the quality of the transfer style images using MuTI. The first approach is assessed from the visual form of the resulting image. The second approach is seen from the value of the style and content loss of the image.

When viewed from the results of the image produced in plain view, the method that maintains image content well is MuTI that uses direct injection with input from content image encoding features and MuTI that uses mean injection with input from image encoding style features, as seen in Figure 6 and Figure 7. While the MuTI approach that uses direct injection with input from image style encoding features cannot properly maintain the semantic value of the image because there is the super-impose phenomenon.

Furthermore, when viewed from the value of style and content loss of the two methods that give good results, MuTI which uses the mean injection with input from image encoding feature style is more able to suppress the value of style loss, as seen in Figure 8.

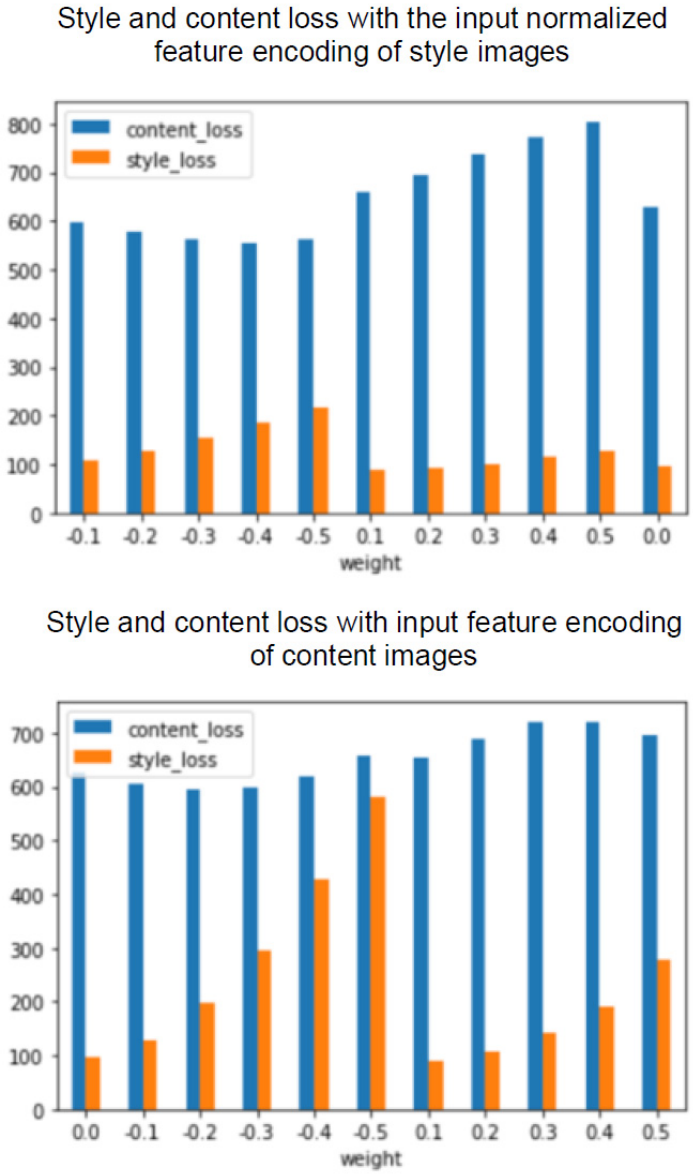

FIGURE 8. Comparison of content and style loss values for MuTI with input from content encoding and normalized styles

This is possible because this technique uses the same spirit as the AdaIN method, to map the image encoding value of the content into the distribution of the image style encoding value. So it makes more sense to shift the distribution of the value of the feature encoding to the value in accordance with the distribution.

The value of content and style loss for images generated using MuTI direct injection that uses input from encoding style images can be seen in Figure 9. In the figure, it can be seen that direct injection of the encoding feature value from the style dramatically increases the loss value of the image, both content and style loss.

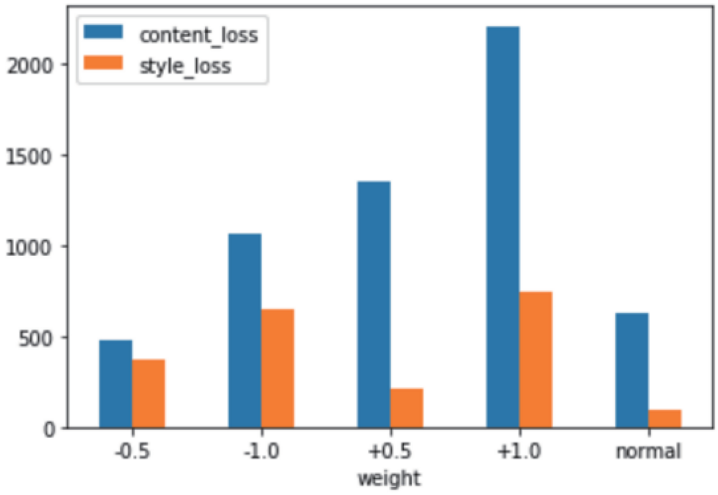

FIGURE 9. Content loss and style loss for images generated using MuTI direct injection with input from image style encoding features

5. **Conclusions.** Based on observations made on the results of experiments, the approach that allows for controlling the Fast Neural Style Transfer is direct injection with the source tensor of the content image encoding feature and the mean injection with the source tensor of the image style encoding feature.

Doing direct injection using the encoding feature of the image style cannot be used to control the weighting of the content or style of the image that the style transfer is performed. This is because at the time of the style transfer, the resulting image shows the phenomenon of super impose.

If the results of the mean injection with input feature encoding style image and direct injection with input feature encoding the image content is compared to the loss value, the mean injection approach with the input feature encoding style image gives better results because it can better suppress the loss value of the image style.

REFERENCES

[1] L. Gatys, A. Ecker and M. Bethge, A neural algorithm of artistic style, *Journal of Vision*, vol.16, no.12, DOI: 10.1167/16.12.326, 2016.

[2] E. Risser, P. Wilmot and C. Barnes, Stable and controllable neural texture synthesis and style transfer using histogram losses, *PLoS ONE*, vol.15, no.6, 2017.

[3] L. A. Gatys, A. S. Ecker and M. Bethge, Image style transfer using convolutional neural networks, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2414-2423, DOI: 10.1109/CVPR.2016.265, 2016.

[4] A. Selim, M. Elgharib and L. Doyle, Painting style transfer for head portraits using convolutional neural networks, *ACM Trans. Graphics (ToG)*, vol.35, no.4, DOI: 10.1145/2897824.2925968, 2016.

[5] J. Johnson, A. Alahi and L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in *Computer Vision –C ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*, B. Leibe, J. Matas, N. Sebe and M. Welling (eds.), Cham, Springer International, 2016.

[6] X. Huang and S. Belongie, Arbitrary style transfer in real-time with adaptive instance normalization, *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[7] H. H. Zhao, P. L. Rosin, Y. K. Lai et al., Automatic semantic style transfer using deep convolutional neural networks and soft masks, *The Visual Computer*, vol.36, pp.1307-1324, 2020.

[8] Y. Zhi, H. Wei and B. Ni, Structure guided photorealistic style transfer, *Proc. of the 26th ACM International Conference on Multimedia (MM'18)*, 2018.

[9] C. Li and M. Wand, Combining Markov random fields and convolutional neural networks for image synthesis, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.2479-2486, DOI: 10.1109/CVPR.2016.272, 2016.

[10] F. Shen, S. Yan and G. Zeng, Neural style transfer via meta networks, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.8061-8069, DOI: 10.1109/CVPR.2018.00841, 2018.

[11] D. Ulyanov, A. Vedaldi and V. Lempitsky, Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4105-4113, DOI: 10.1109/CVPR.2017.437, 2017.

[12] S. Loffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, *Proc. of the 32nd International Conference on International Conference on Machine Learning (ICML'15)*, vol.37, pp.448-456, 2015.

[13] T. Q. Chen and M. Schmid, Fast patch-based style transfer of arbitrary style, *NIPS*, Barcelona, Spain, 2016.