Artistic Image Style Transfer with Improved Style Attention Networks

Anjali Bajaj
dept. of Computing Science
University of Alberta
Edmonton, Alberta-Canada
abajaj2@ualberta.ca

Baljinder Kaur
dept. of Computing Science
University of Alberta
Edmonton, Alberta-Canada
bkaur2@ualberta.ca

Sabrina Nasrin
dept. of Computing Science
University of Alberta
Edmonton, Alberta-Canada
snasrin@ualberta.ca

***Abstract*—Art is a combination of imagination, dedication, and efforts put into the process. Although, humans have mastered the skill by presenting unique visual experiences through designing a complex interplay between the content and style of an image such as paintings. However, in today's technological era this process still needs more exploration. Currently, there are no such expert existing artificial systems that can present skills the same as humans are doing. Even though these distinct visual styles are very recognizable to human observers, the visual style is a difficult concept to rigorously define in terms of computer vision. However, a convolutional neural network which is a class of deep neural networks is used to extract features from the artistic images makes it possible to separate the style information and the image content. The main objective of arbitrary image style transfer with style attention network is to learn the characteristics of the style image and apply the learned style to the content image keeping the structure of the content image unchanged. In this paper, we introduce a new restoration loss with the loss function of SANet. This restoration loss is improving the preservation of the content structure with enriched style patterns. Our experimental results suggest that SANet with new restoration loss produces better output than the original SANet and other state-of-the-art algorithms. Our project report is divided into various sub heading such as**

Index terms*— *Image Style Transfer, Convolutional Neural Network (CNN), Generative Adversarial Network (GAN), AdaIN, AvatarNet, WCT, SANet, Loss, LPIPS Distance

I. Introduction

Image style transfer refers to a technique of transferring the style of one image into another. For thousands of years, the art theories behind different artworks have been drawing the attention of the artists [1]. Different artists have their styles of painting. Painting a picture in a particular art style not only requires an expert artist but also needs a lot of time. Since mid-1990, computer science researchers also started to research the art theories behind these appealing artworks. Before 2000, researchers focused on manually modeled complex mathematical formulas to summarize and generate textures [2]. However, this process was time-consuming as well as labor-intensive. Recently, image style transfer has become a trending research topic in computer vision. This technique requires two images to specify to get the final output. The first image is the base image also known as content image and the second one is the style image whose style is to transfer to the content image. There is a lot of different styles, such as abstract, impressionism, realism, modernism, futurism, etc. which can be regarded as style images. The style transfer algorithm changes the image style while maintaining the content image's structure, resulting in a final composite image that is a perfect blend of the input image content and desired style [3]. Deep learning-based image style transfer takes the advantage of feature extraction of convolutional neural networks (CNN). These convolution neural networks are exciting because they can in some cases create results with state-of-the-art quality with their more powerful multilevel image fea-

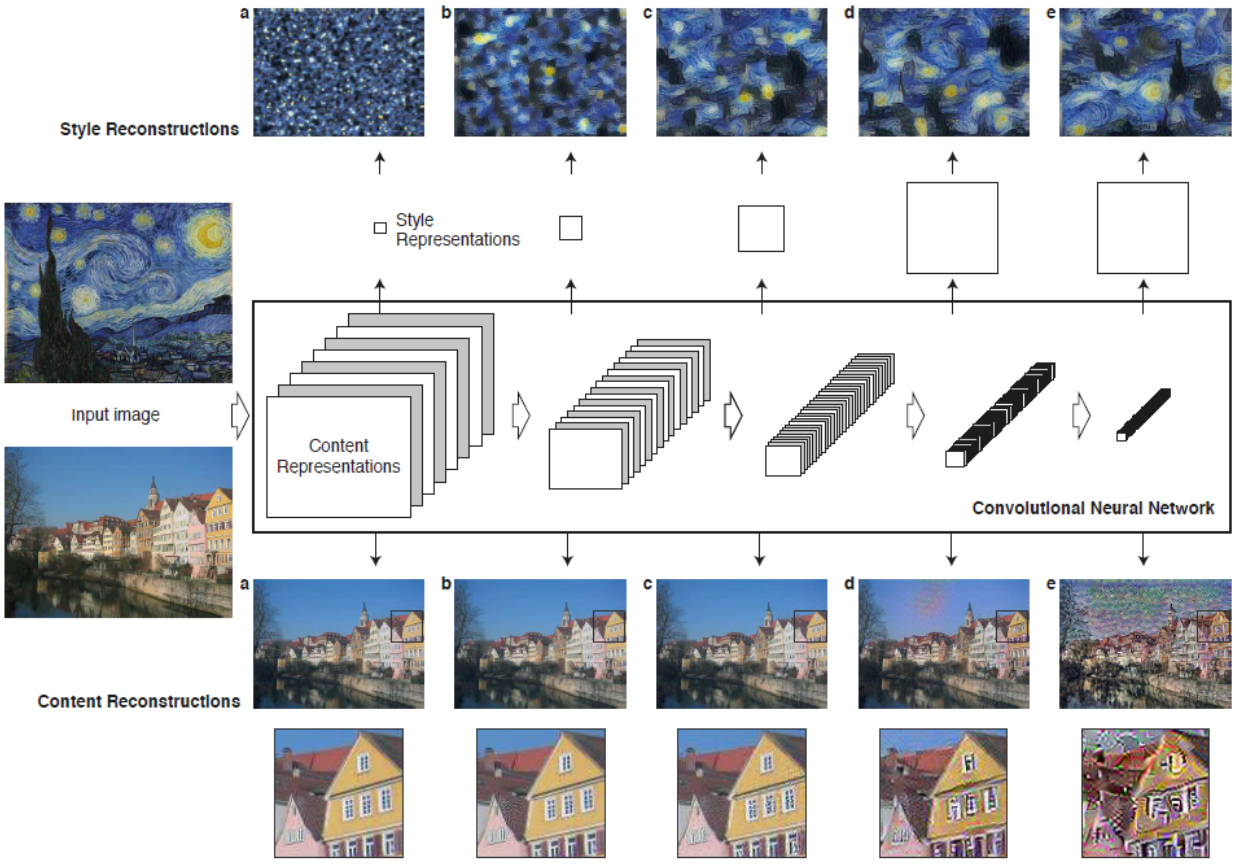

Fig. 1. A system framework of image style transfer using CNN. The input image is filtered by CNN network, and the network's response at a specific layer reconstructs the input image, so as to visualize the information in different processing stages of CNN.

ture extraction and representation ability. Image style transfer can be referred to as a texture transfer problem.

The relation between features of content image and style image extracted from a pretrained VGG network was first shown by Gatys[4]. However, that method is comparatively slow. After that, many researchers worked in style transfer to reduce the computation time. AdaIn [5] is comparatively faster than Gatys and it matches the mean and variance of the content image to the style image. Due to the over simplified nature of this method, the output of AdaIN is not satisfactory. WCT [6] uses covariance in stead of variance and transforms the features of the content into style feature space through a whitening and coloring process. WCT becomes computationally expensive if the features have large number of dimensions. AvatarNet [7] is a patch-based method which considers holistic style distribution as well as local style patterns. However, this method also gives distorted output and fails to balance between global and local style patterns. In this work, we propose an improved SANet with a new restoration loss along with content loss and style loss. This improved model is capable of producing high quality stylized output where the content structure is preserved, the consistency of the content image is maintained as well as the balance between the style and content image is obtained. We have shown the supremacy of our model through experimental results. The paper is organized as follows. Section II describes the related works. In section III, we discuss the baseline methods. Section

IV describes experimental setup while results are demonstrated in section V.Section VI draws the conclusion.

II. Related Works

In this section, we are going to review the works related to image style transfer based on convolutional neural network and generative adversarial network.

In this paper [8], they have discussed some advance the stylization process into two significant ways such as feed-forward texture synthesis and another on is the stylization network. Along with this, this paper also introduced the instance normalization and a structural change which help to make the training process easier for stylization networks. This will also help to achieve lower loss levels for the training process. Furthermore, it also explained the new learning formulation to train the generator networks to sample out from the Julesz ensemble, hence it putting diversity in the desired outputs of this model. It mentioned that during the complete runtime these both above improvements brought a noticeable change in stylization of images as well as in textures.

In this paper [9], they have presented the new CNN-based methods for artistic style transfer that are faster as well as adaptable to arbitrary styles. This method not only records the content but also keeps style details in a single layer of the model. Also, this method is used to train an inverse network which is useful in producing the output in a very minimum amount of time. Moreover, its consistency and intuitive tuning parameters allow a frame-by-frame application transfer to a video. The main advantage of this method is the stylization process using new style images. However, it still lacking in the matter of speed and global style measurement.

In this paper [10], they have presented the Dynamic ResBlock-GAN (Generative Adversarial Network) for artistic style transfer. In the model introduction, "style codes" are designed as the shared parameters, for DRB connecting each of the style encoding network features and the style transfer network to reduce the difference between arbitrary style transfer and collection style transfer in a single model. The examined the style class-aware attention mechanism which captures through the images in style collections and discriminative network makes full use of style information in target-style images which further stimulates the model's ability for the artistic style transfer. Large-scale experimental outputs demonstrate the outstanding performance of the proposed DRB-GAN model in generating synthetic style images with higher quality as compared to the state-of-the-art.

In this paper [11], they have discussed Maximum Mean Discrepancy optimization which offers a closed type surrogate for distinguishing in the framework of adversarial nets. With the use of Bayesian optimization to the parameters, it was detected that this network performed much better than the adversarial network, especially in density estimation. While on the other side, between the MMD nets digits and digits produced by MNIST, it has a clear discrepancy. It signals that KDE is not ready yet for the evaluation of these models. It is also shown that the speed of MMD Nets is quite higher as compared to other which is why it considers best, especially at the beginning of the expensive procedures.

In this paper [12], they defined each Julesz as a textured pattern, as an inversion problem they set texture modeling. In texture images, the perception of the element plays a crucial role in the segmentation of these textures. The modeling texture with filtering and histograms is the first approximation and for the hierarchical system, the study has to be completed yet. Two patterns have been separated here with the help of Bayesian inference, the first one is tree clutter and another one is buildings images. Later on, they studied these typical images to develop an efficient algorithm and measure its performance. And at the end, they have also applied ensembles to get the fundamental bounds for different minor determinations.

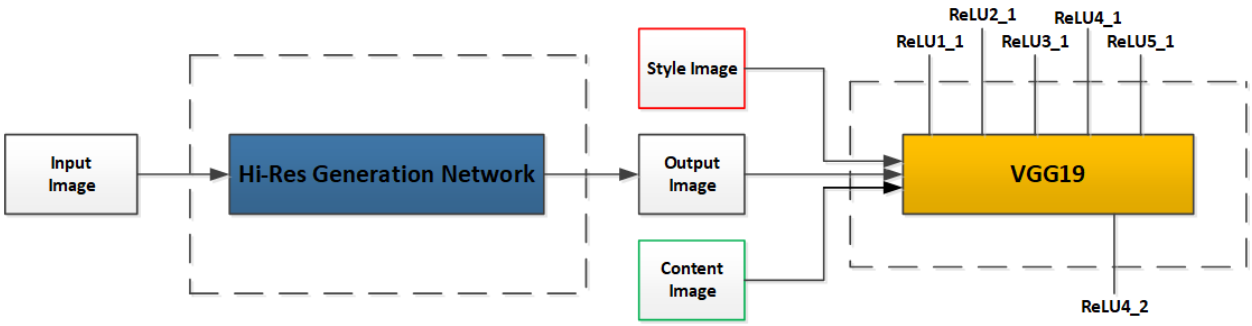

Fig. 2. Gatys Model structure

III. Baseline models

The primary two tasks to address for style transfer are capturing the content representation of one image and the style of another to combine them. In this paper, we have used four models as our baseline models.

A. Gatys

Gatys [13] combined neural network and style transfer, which officially opened the prelude of Neural Style Transfer. Gatys applied the gram matrix to different local feature maps extracted by the VGG-19 as mentioned in the equation 1 network and calculated the correlation between features to form a statistical model. The computation of style features is performed on all convolution layers. A part of local features is directly taken as the content, and the content features are for a convolution layer. Finally, combine the content features and style features of the picture together to form a new picture as mentioned in the equation 2. In the figure 2, we can see the architecture of Gatys Model as mentioned above.

$$E_L = \sum (G^L - A^L)^2 \tag{1}$$

$$L_{total(p,a,x)} = \alpha L_{content}(\vec{p}, \vec{x}) + \beta L_{style}(\vec{a}, \vec{x}) \tag{2}$$

Advantages: Good and usually regarded as a gold standard. Limitation: But it Take a long time, but the results are relatively stable.

B. Universal Style Transfer

Universal Style Transfer [6] via Feature Transforms aims to transfer arbitrary visual styles to content images. The architecture includes the Content Image and style Image as input to the Encoder. As a first step, it is trained with VGG-19 on the Imagenet image classification task. This is used as the Encoder as mentioned in the equation 11 . As a next step, the Encoder is fixed and the decoder as mentioned in the equation 4 is trained to reconstruct the Image. The output of the Encoder is sent into the Whitening and Coloring Transform as mentioned in the equation 5 block whose output is then passed on to the decoder to get the final stylized output. The lower layers capture features like style color whereas the higher layers as mentioned in the equation 6 below capture style structures. In the figure 3, we can see the architecture of Universal Style Transfer Model as mentioned above.

$$L = ||I_o - I_i||_2^2 + \lambda ||\phi(I_o) - \phi(I_i)||_2^2 \tag{3}$$

$$\widehat{f_c} = E_c D_c^{-1/2} E_c^T f_c \tag{4}$$

$$\widehat{f_{cs}} = E_s D_s^{1/2} E_s^T \widehat{f_c} \tag{5}$$

$$\widehat{f_{cs}} = \alpha \widehat{f_{cs}} + (1 - \alpha) f_c \tag{6}$$

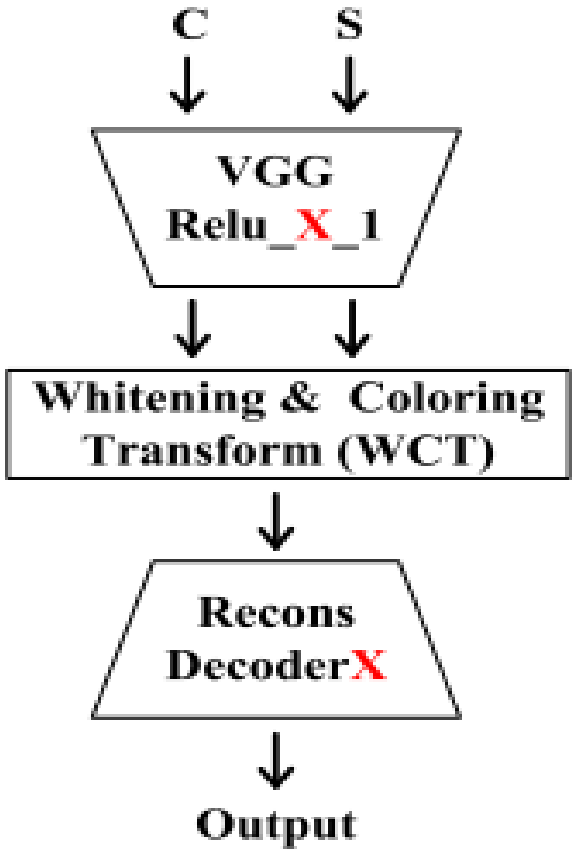

Fig. 3. Universal Style Transfer Model structure

Advantages: Universal style transfer aims to transfer arbitrary visual styles to content images. Existing feed-forward based methods, while enjoying the inference efficiency. Limitations: It distorts the brush strokes and circular patterns

C. Adaptive Instance Normalization(AdaIN)

AdaIN [5] receives a content input and a style input, and simply aligns the channel-wise mean and variance of the content image to match those of the style image. The AdaIN style transfer network takes a content image and an arbitrary style image as inputs and synthesizes an output image transfer network concerning content image and style image that recombines the content and style of the respective input images. The network adopts a simple encoder-decoder architecture, in which the encoder f is fixed to the first few layers of a pre-trained VGG-19. After encoding the content and style images in the feature space, both the feature maps are fed to an AdaIN layer that aligns the mean and variance of the content feature maps to those of the style feature maps, producing the target feature maps. A randomly initialized decoder is trained to invert the feature map back to the image space, generating the stylized image as mentioned in the equation below 7. In the figure 4, we can see the architecture of Adain as mentioned above.

$$L(Input, Style, Out) = \sum_i \lambda_{style,i} ||encoder_i(Input) - encoder_i(Out)||_2^2 + \sum_i \lambda_{style,i} ||stats((encoder)_i(Style)) - stats((encoder)_i(Out))||_2^2 \quad (7)$$

Advantage: AdaIN able to control the strength of the style on the generated image during runtime. Another advantage the method provides is the inference speed over the other models at the time. Limitation: AdaIN cannot keep the color distribution.

D. Avatar-Net

Zero-shot artistic style transfer is an important image synthesis problem aiming at transferring arbitrary style into content images. However, the trade-off between the generalization and efficiency in existing methods impedes a high-quality zero-shot style transfer in real-time. Proposing an efficient yet effective Avatar-Net [7] that enables visually plausible multi-scale transfer for arbitrary style. The proposed Avatar-Net employs an hourglass network with multi-scale style adaptation modules that progressively fuse the styles from the encoded features into the corresponded decoded features, thus it enables multi-scale style transfer in one feed-forward pass. The

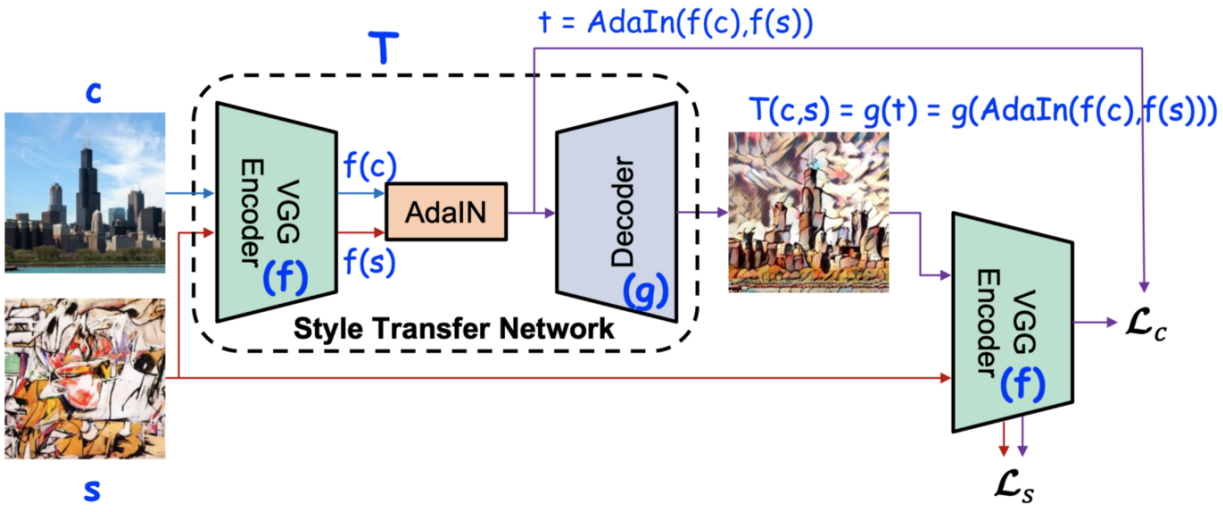

Fig. 4. AdaIN Model structure

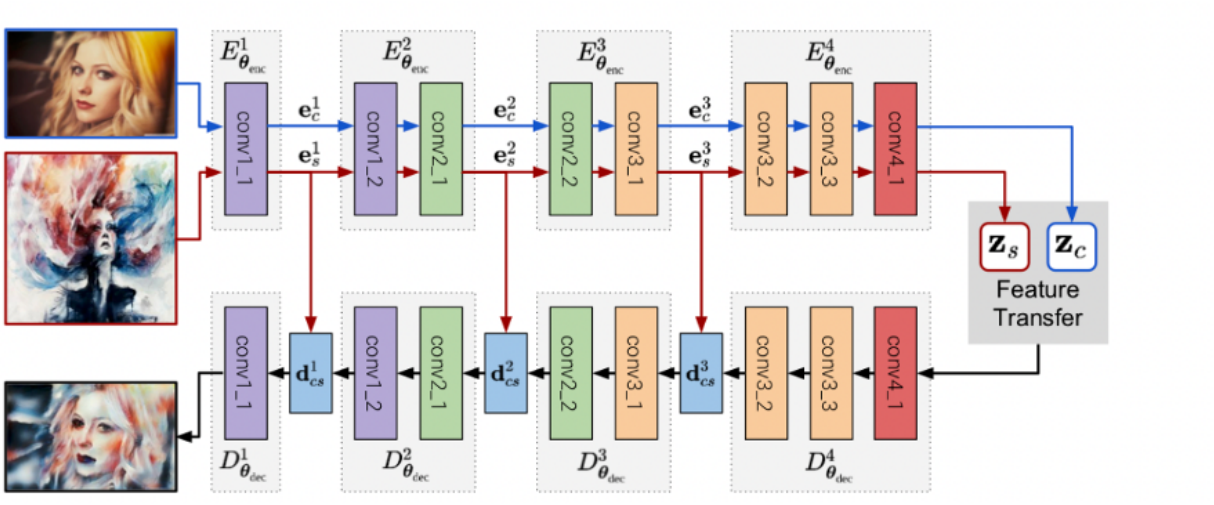

Fig. 5. Avatarnet Model structure

style transfer requires a special procedure to fit the proposed network architecture. At first, Avatar-Net takes a content image and an arbitrary style image as inputs and extracts content feature and style feature in the bottleneck layer through the encoder module. In the meanwhile, the style image also bypasses the multi-scale encoded features. Secondly, the content feature is then transferred based on the style features through the proposed style decorator module. In the end, the stylized image is inverted by the decoded module with multiple style fusion modules that progressively modify the decoded features under the guidance of multi-scale style patterns.8 In the figure 5, we can see the architecture of Avatar-net Model as mentioned above.

$$F_{SF}(d^l_{cs}; e^l_s) = \sigma(e^l_s) \circ \frac{(d^l_{cs} - \mu(d^l_{cs})}{(\sigma(d^l_{cs})} + \mu(e^l_{cs}) \tag{8}$$

Advantages: The result by Avatar-Net receives concrete multi-scale style patterns (e.g. color distribution, brush strokes and circular patterns in the style image). Limitations: It

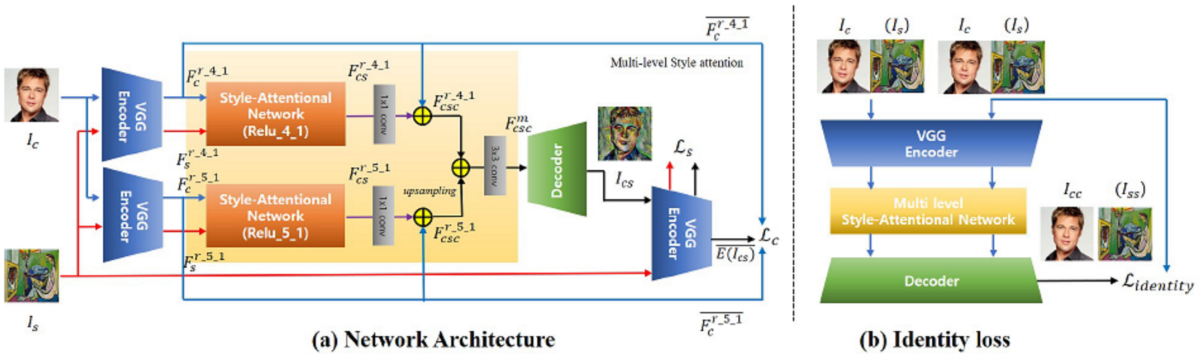

Fig. 6. SANet Architecture

frequently cannot represent the local and global style patterns at the same time due to its dependency on the patch size and also cannot keep the content structure in most cases.

IV. Experimental Setup

Figure 6 shows an overview of the SANet. Codes and the pretrained models are available here.

A. Backbone Architecture

We have used SANet [14] as our backbone model. SANet takes a content image I_c and a style image I_s and produces an output I_cs. It uses encoder-decoder module and two SANets. In this paper, they are following the decoder settings of AdaIN [5] and a pretrained VGG-19 network as encoder. In this network, they trained two Style Attention Networks and integrated them by taking VGG feature maps. These feature maps are encoded from Relu_4_1 and Relu_5_1 and working as input of the SANets. They first extracted the VGG feature maps $F_c = E(I_c)$ and $F_s = E(I_s)$ from the content image and style image respectively at a certain layer and then fed them to a SANet module to again extract their corresponding features from F_c and F_s and produced the following output feature map:

$$F_{cs} = SANet(F_c, F_s) \tag{9}$$

Then they obtained F_{csc} as shown in eqn 19 applying 1×1 convolution to F_{cs} and summing F_c and F_{cs}.

$$F_{csc} = F_c + W_{cs}F_{cs} \tag{10}$$

They combined two feature maps $F_{csc}^{r\_4\_1}$ and $F_{csc}^{r\_5\_1}$ to F_{csc}^{m} as shown in eqn 11 by $3\times$ convolution and $F_{csc}^{r\_5\_1}$ is added to $F_{csc}^{r\_4\_1}$ after upsampling.

$$F_{csc}^{m} = conv_{3\times3}(F_{csc}^{r\_4\_1} + upsampling(F_{csc}^{r\_5\_1})) \tag{11}$$

The output stylized image I_{cs} is synthesized by feeding F_{csc}^{m} into the decoder as shown in eqn 12

$$I_{cs} = D(F_{csc}^{m}) \tag{12}$$

SANet architecture takes inputs as the feature map F_c and F_s from content and style images from VGG-19 encoder and after normalizing them, makes transformation to feature spaces f and g in order to compute the attention between content and style feature maps $\overline{F_c^i}$ and $\overline{F_c^i}$ as shown in eqn 22 and 7. Here, feature spaces are calculated as $f(\overline{F_c}) = W_f\overline{F_c}$, $g(\overline{F_s}) = W_g\overline{F_s}$ and $h(F_s) = W_hF_s$. W_f, W_g and W_h are the learned weighted matrices. Normalized factor $C(F)$ is calculated as $C(F) = \sum_{\forall_j} exp(f(\overline{F_c^i})^T g(\overline{F_s^j}))$.

$$F_{cs}^{i} = \frac{1}{C(F)} \sum_{\forall_j} exp(f(\overline{F_c^i})^T g(\overline{F_s^j}))h(F_s^j) \tag{13}$$

To compute the loss function to train the decoder and the SANets, they used pretrained

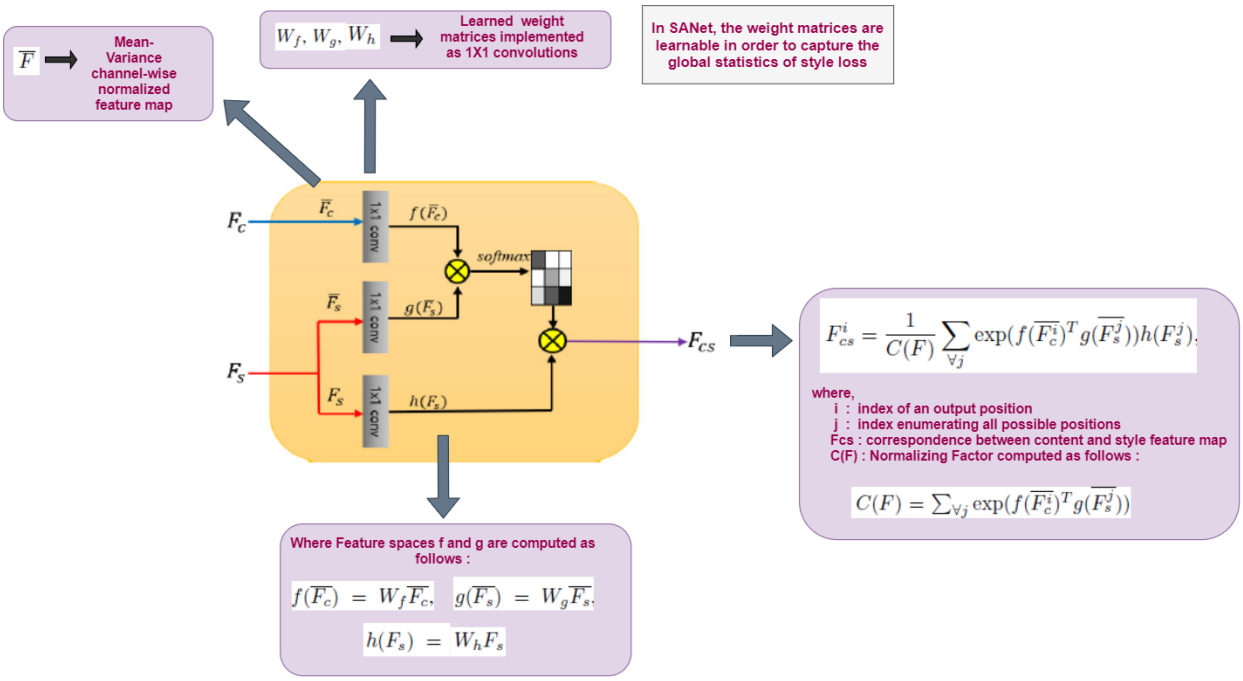

Fig. 7. Style Attention Module

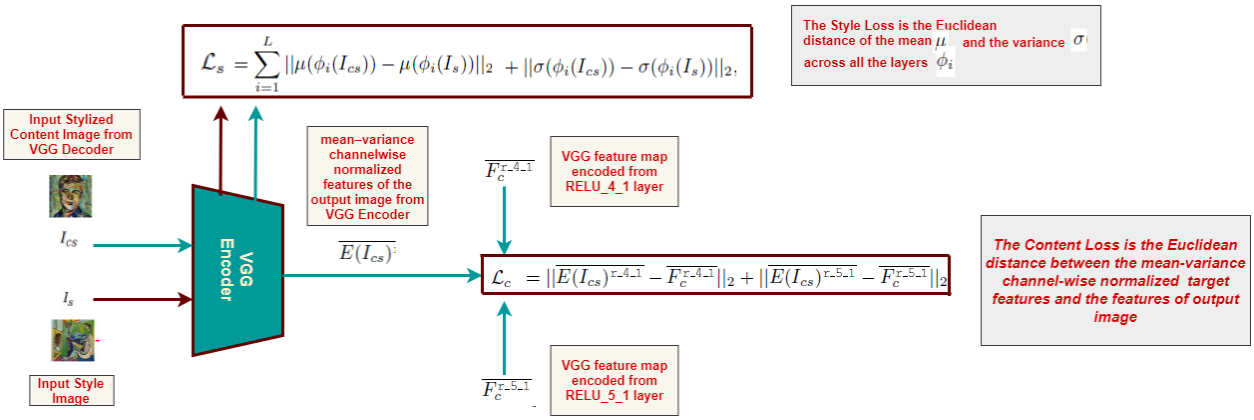

Fig. 8. SANet Loss Function

vgg-19 encoder. Total loss is calculated as eqn 14.

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \mathcal{L}_{identity} \tag{14}$$

The content loss $\mathcal{L}_c$ is the Euclidean distance between the mean-variance channel-wise normalized target features, $\overline{F_c^{r\_4\_1}}$ and $\overline{F_c^{r\_5\_1}}$ and mean-variance channel-wise normalized features of the output image VGG features, $\overline{E(I_{cs})^{r\_4\_1}}$ and $\overline{E(I_{cs})^{r\_5\_1}}$, as follows:

$$\mathcal{L}_c = ||\overline{E(I_{cs})^{r\_4\_1}} - \overline{F_c^{r\_4\_1}}||_2 + ||\overline{E(I_{cs})^{r\_5\_1}} - \overline{F_c^{r\_5\_1}}||_2 \tag{15}$$

The style loss $\mathcal{L}_s$ is defined as follows:

$$\mathcal{L}_s = \sum_{i=1}^{L} ||\mu(\phi_i(I_{cs})) - \mu(\phi_i(I_s))||_2 + ||\sigma(\phi_i(I_{cs})) - \sigma(\phi_i(I_s))||_2 \tag{16}$$

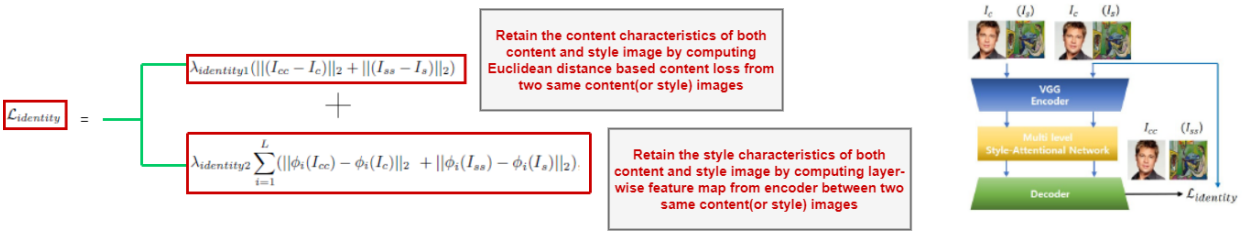

Fig. 9. SANet Identity Loss

They have used Relu_1_1, Relu_2_1, Relu_3_1, Relu_4_1 and Relu_5_1 and AdaIN style loss [5]. Figure 8 demonstrates the loss calculation. In the SANet, transfer model can be trained by considering only the global statistics of the style loss $\mathcal{L}_c$. A identity loss is defined in SANet[5] as shown in eqn 17. This identity loss can consider both global statistics and semantic local mapping between content and style features.

$$\begin{aligned}\mathcal{L}_{identity} = \lambda_{identity1}(||(I_{cc} - I_c)||_2 \\ + ||(I_{ss} - I_s)||_2) \\ + \lambda_{identity2} \sum_{i=1}^{L} (||\phi_i(I_{cc}) - \phi_i(I_c)||_2 \\ + ||\phi_i(I_{ss}) - \phi_i(I_s)||_2) \end{aligned} \quad (17)$$

An overview of the computation of identity loss is shown in figure 9

B. Architecture of Improved Style Attention Network

However, our model works in two-phase. In the first phase, it takes a content image I_c and a style image I_s and produces an output I_cs in a similar way to SANet. In the second phase, we are again sending I_cs to the model as content image and I_c as the style image. It is giving us the restored output I_r. In this phase, we extracted the VGG feature maps $F_r = E(I_{cs})$ and $F_s = E(I_c)$ from the stylized image and content image respectively and then fed them to a SANet module to again extract their corresponding features from F_r and F_s and produced the following output feature map:

$$F_{rc} = SANet(F_r, F_s) \quad (18)$$

Then we obtained F_{rcsc} applying 1×1 convolution to F_{rc} and summing F_r and F_{rc}.

$$F_{rcsc} = F_r + W_{cs} F_{rc} \quad (19)$$

We combined two feature maps $F^{r\_4\_1}_{rcsc}$ and $F^{r\_5\_1}_{rcsc}$ to F^{m}_{rcsc} by $3\times$ convolution and $F^{r\_5\_1}_{rcsc}$ is added to $F^{r\_4\_1}_{rcsc}$ after upsampling.

$$F^{m}_{rcsc} = conv_{3\times3}(F^{r\_4\_1}_{rcsc} + upsampling(F^{r\_5\_1}_{rcsc})) \quad (20)$$

The output stylized image I_r is synthesized by feeding F^{m}_{rcsc} into the decoder.

$$I_r = D(F^{m}_{rcsc}) \quad (21)$$

In this phase, this architecture takes inputs as the feature map F_r and F_s from stylized image from the first phase and content image as style images from VGG-19 encoder and after normalizing them, makes transformation to feature spaces f and g in order to compute the attention between content and style feature maps $\overline{F^i_r}$ and $\overline{F^i_s}$. Here, feature spaces are calculated as $f(\overline{F_r}) = W_f \overline{F_r}$, $g(\overline{F_s}) = W_g \overline{F_s}$ and $h(F_s) = W_h F_s$. W_f, W_g and W_h are the learned weighted matrices. Normalized factor $C(F)$ is calculated as $C(F) = \sum_{\forall_j} exp(f(\overline{F^i_r})^T g(\overline{F^j_s}))$.

$$F^i_{rcs} = \frac{1}{C(F)} \sum_{\forall_j} exp(f(\overline{F^i_r})^T g(\overline{F^j_s})) h(F^j_s) \quad (22)$$

In the first phase, it calculates the loss function as 14. However, in the second phase, the new loss function is calculated as followed:

$$\mathcal{L} = \lambda_c \mathcal{L}_c + \lambda_s \mathcal{L}_s + \mathcal{L}_{identity} + \lambda_r \mathcal{L}_r \quad (23)$$

Here, $\mathcal{L}_c$, $\mathcal{L}_s$,$\mathcal{L}_{identity}$,$\mathcal{L}_r$ are content loss, style loss, identity loss and restoration loss respectively. λ_c, λ_s, and λ_r are different weights

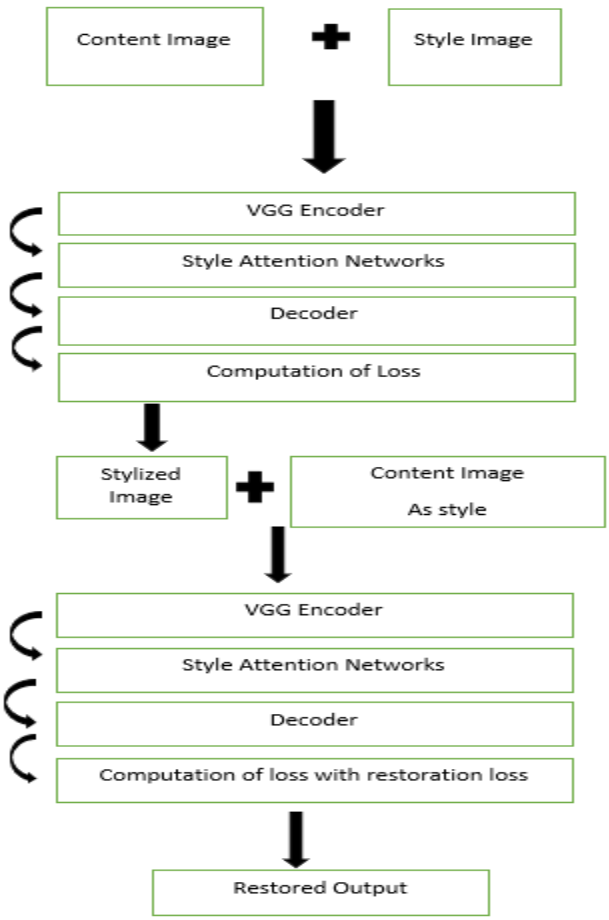

Fig. 10. Improved SANet architecture

for the losses. The restoration loss $\mathcal{L}_r$ is the Euclidean distance between the mean-variance channel-wise normalized target features, $\overline{F_r^{r\_4\_1}}$ and $\overline{F_r^{r\_5\_1}}$ and mean-variance channel-wise normalized features of the output image VGG features, $\overline{E(I_r)^{r\_4\_1}}$ and $\overline{E(I_r)^{r\_5\_1}}$, as follows:

$$\mathcal{L}_r = ||\overline{E(I_r)^{r\_4\_1}} - \overline{F_r^{r\_4\_1}}||_2 + ||\overline{E(I_r)^{r\_5\_1}} - \overline{F_r^{r\_5\_1}}||_2 \quad (24)$$

Detailed architecture of improved model is shown in figure **??**. The new restoration loss keeps the consistency and stability of the original image. As a result, in final output we get a better stylized image where the image is stylized as well as the features of the original image is maintained.

C. Training

We trained our model using MS-COCO [15] dataset as the content images and Painter by Number Dataset [16] as styles images. Many of the images in this dataset were obtained from WikiArt [17]. Additional paintings were provided by other artists. Both the datasets contain roughly $80,000$ training images. We have trained our model with Adam optimizer[18] with a learning rate of 0.0001. We took a batch size of five content-style image pairs. During training time, we re-scaled the dimension of the images to 512 as well as preserved the aspect ratio. Then we randomly cropped a region of size 256×256 pixels. Our model can test images with any input size because it is fully

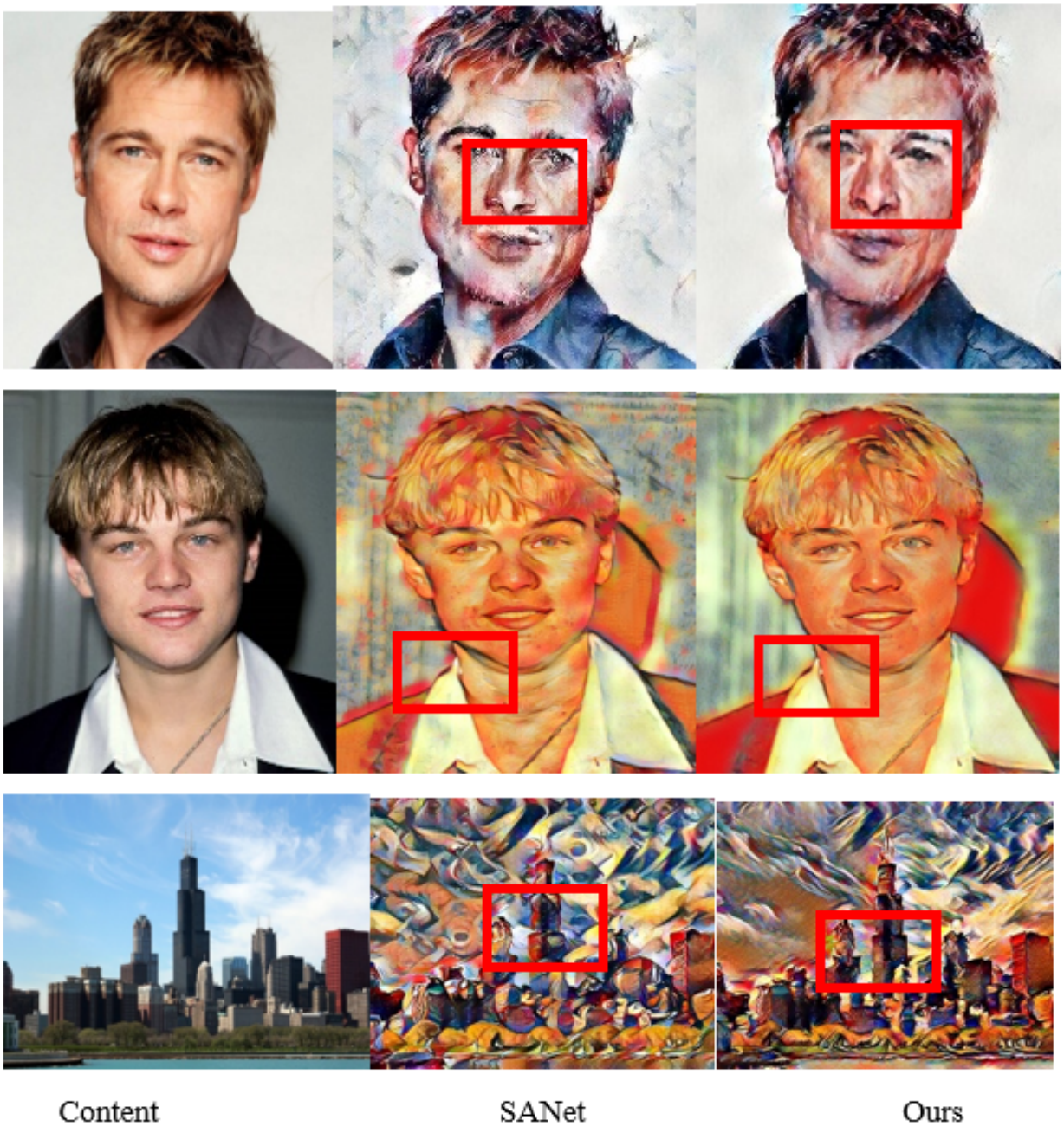

Fig. 11. Qualitative Analysis

convolutional. The weighting parameters are set $\lambda_c = 1$, $\lambda_s = 3$, $\lambda_r = 2$, $\lambda_{identity} = 1$ and $\lambda_{identity} = 50$ in our experiments.

V. Results

To evaluate the performance of our model, we compared our model with three types of arbitrary style transform methods: the iterative optimization method proposed by Gatys[4], two feature transformation-based methods WCT [6] and AdaIN [5] and the patch-based method AvatarNet[7].

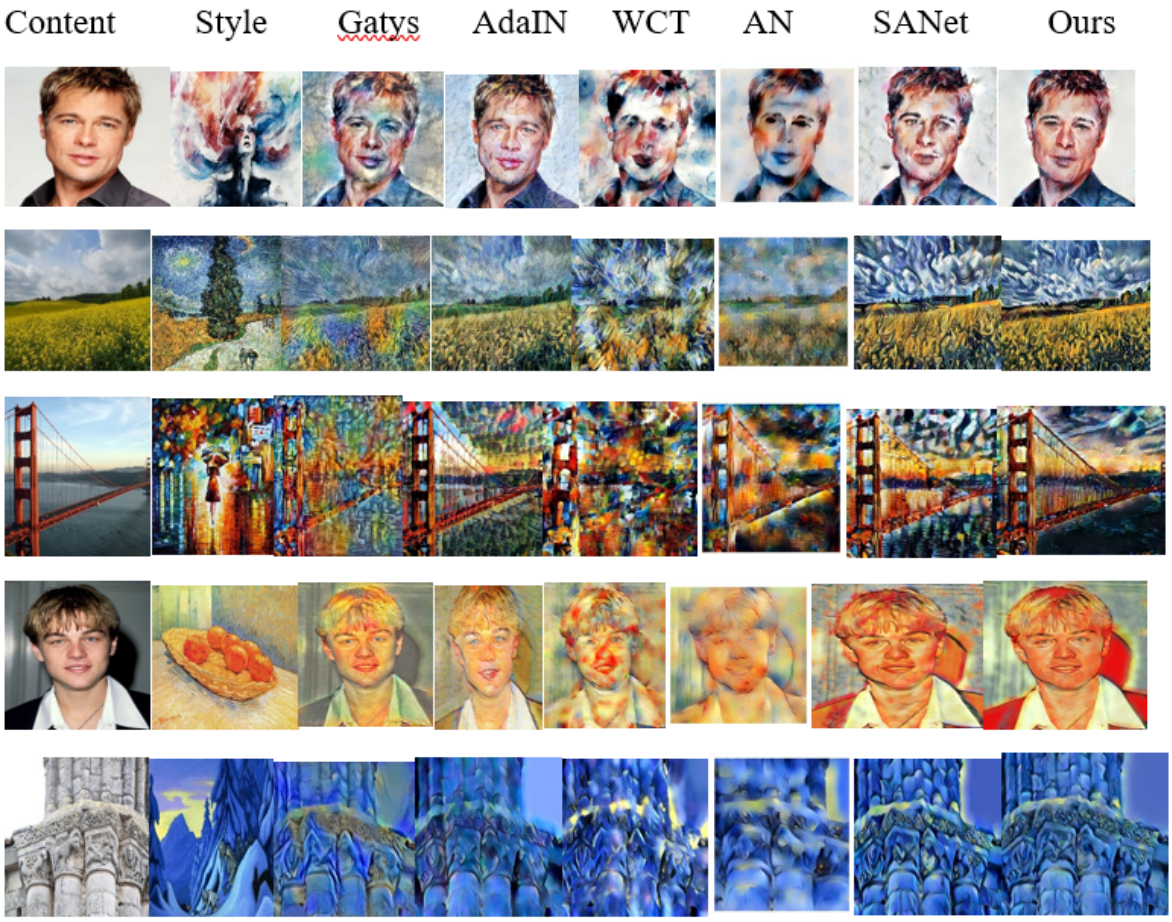

Fig. 12. Qualitative Analysis

A. *Qualitative Analysis*

First of all, we will show the qualitative comparison between SANet and our model. As shown in fig 11, it is observed that, for SANet, the shape of the nose and eye of the human portrait got distorted. However, our model preserved the shape of the nose and eye satisfactorily. In the second row of figure 11, it is observed that, the part of neck got deformed, however, in our model this edge of the neck is preserved properly. In the third row, we can see the edges of the building are sharper in the output of our model than the output of SANet. It also maintained the consistency of the buildings more than SANet output image.

In the figure 12, it is observed that, the output of our model is way better than the other models we have used for testing. The output from Gatys is somewhat stable but fails to fullfill the requirement. For AdaIn, it has distorted the color. WCT is over stylizing the output and deforming the original image. AN or AvatarNet could not keep content structure throughout the stylized image because of its dependency on the style image patch sizes. SANet is giving better output than other models as it parses diverse style patterns such as global color distribution, texture, and local style patterns while maintaining the structure of the content. However, it is providing better output as it is also distorting the shapes and features of the content image. Lastly, we can see that the output of our model is well structured as well as stylized with the given style image.

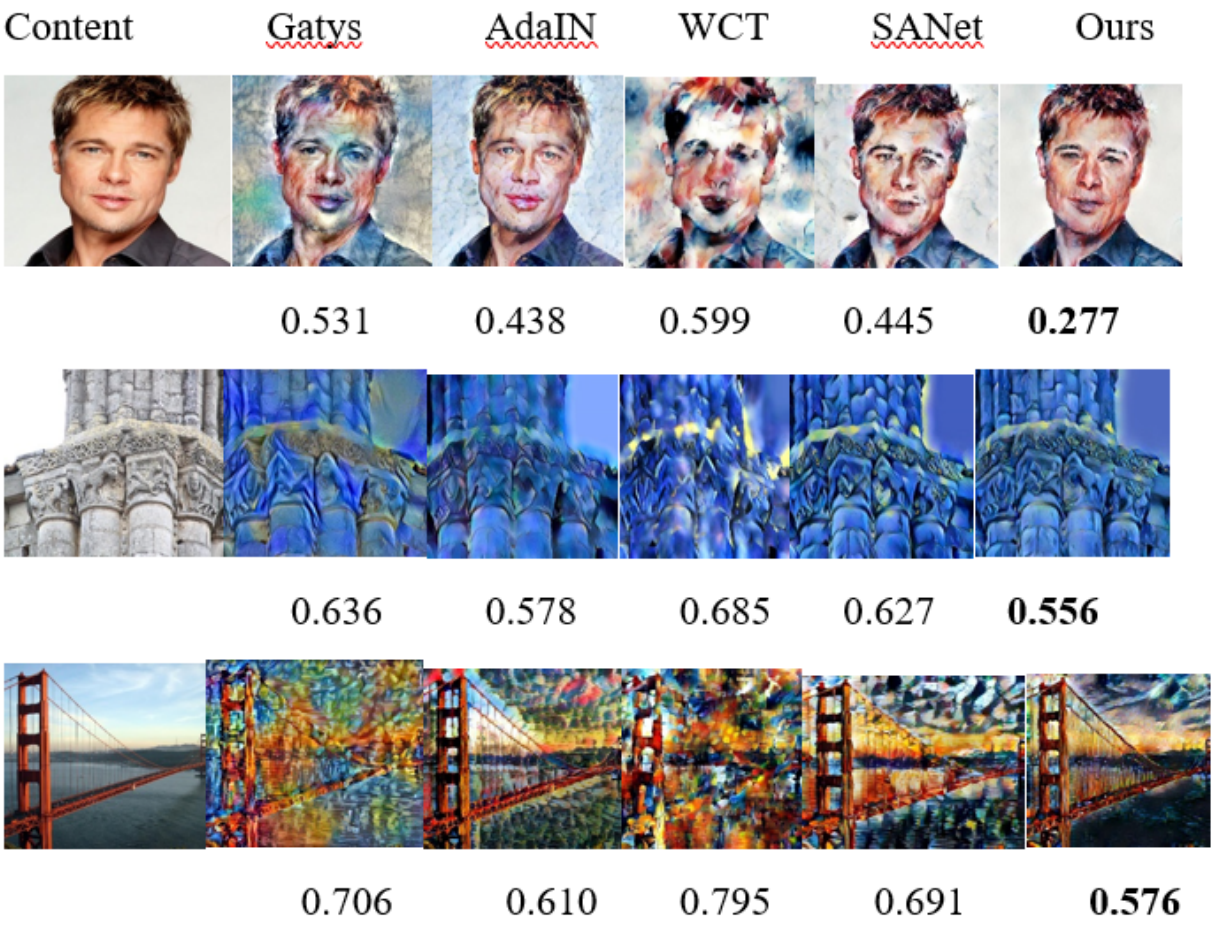

Fig. 13. Quantitative Analysis

B. *Quantitative Analysis*

To evalute the performance of the models numerically, we have used LPIPS[19] distance. LPIPS is a widely used metric to measure diversity. In this paper, we employ LPIPS to measure the stability and consistency of two given images by computing the average perceptual distances between two images. Here, we expect a lower LPIPS value to achieve better stability and consistency. We tested several pairs of content images and style images to get the LPIPS distance. From the results shown in the fig 13, we have seen that our approach obtains the best score among all other methods.

VI. Conclusion

In this paper, we have introduced a new restoration loss to achieve consistency and stability in the output stylized image. Our model achieved a lower LPIPS value which demonstrates that the output is preserving the features of the content image as well as the structure. As a result, in the final output, we get a better-stylized image where the image is stylized as well as the features of the original image are maintained. We have used SANet as our backbone model as it already provides better-stylized images than the other artistic style models which we were covered in the baseline model section. We have noticed that our model does not provide better output when

the style image has more brush strokes, such as, van gogh's paintings. In future, we will try to remove this limitation. Our future work will also include video stylization to propagate the style from a few selected keyframes to the rest of the sequence where a specific part of a moving object will be stylized.

ACKNOWLEDGEMENT

The authors would like to Dr. Basu and our mentor Yingnan Ma for their support and suggestions about the project.

REFERENCES

[1] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, "Neural style transfer: A review," *IEEE transactions on visualization and computer graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.

[2] N. Jin, "Cnn-based image style transfer and its applications," in *2020 International Conference on Computing and Data Science (CDS)*. IEEE, 2020, pp. 387–390.

[3] L. Liu, Z. Xi, R. Ji, and W. Ma, "Advanced deep learning techniques for image style transfer: a survey," *Signal Processing: Image Communication*, vol. 78, pp. 465–470, 2019.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[5] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1501–1510.

[6] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," *Advances in neural information processing systems*, vol. 30, 2017.

[7] L. Sheng, Z. Lin, J. Shao, and X. Wang, "Avatar-net: Multi-scale zero-shot style transfer by feature decoration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8242–8250.

[8] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6924–6932.

[9] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," *arXiv preprint arXiv:1612.04337*, 2016.

[10] W. Xu, C. Long, R. Wang, and G. Wang, "Drb-gan: A dynamic resblock generative adversarial network for artistic style transfer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6383–6392.

[11] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," *arXiv preprint arXiv:1505.03906*, 2015.

[12] S. C. Zhu, X. W. Liu, and Y. N. Wu, "Exploring texture ensembles by efficient markov chain monte carlo-toward a" trichromacy" theory of texture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 554–569, 2000.

[13] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[14] D. Y. Park and K. H. Lee, "Arbitrary style transfer with style-attentional networks," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5880–5888.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[16] "Painter by number kaggle dataset," https://www.kaggle.com/c/painter-by-numbers/data.

[17] F. Phillips and B. Mackintosh, "Wiki art gallery, inc.: A case for critical thinking," *Issues in Accounting Education*, vol. 26, no. 3, pp. 593–608, 2011.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.