

Deep Generative Framework for Multi-Variant Style Fusion from a Single Image

M Srinivasa Sai Kumar Reddy

*Department of AI
FET, Jain Deemed-to-be University,
Bengaluru-562112
msskr2109@gmail.com*

Janaki Kandasamy

*Department of CSE (AI)
FET, Jain Deemed-to-be University,
Bengaluru-562112
k.janaki@jainuniversity.ac.in*

ABSTRACT

Neural Style Transfer (NST) is the most famous technique that comes under Deep Learning and is used for creating digital artworks. Current NST techniques use only one type of artistic style per each image, thus limiting the number of possible stylized images using a single style image. In order to overcome such limitations, this work introduces a framework for generating a multi-variant style image through the use of just one style image. The model architecture is a GAN where the generator generates a stylized image while the discriminator increases its quality. The style transfer process is performed with the help of a VGG19 network which extracts useful style and content features from an input image. The novelty of the framework is the inclusion of a multi-variant fusion module that enables generating a multi-variant output style image from various VGG19 layer styles. Training is performed using adversarial, content, style, perceptual, and total variation losses in order to optimize the quality and consistency in style of generated images. In contrast to conventional NST methods based on iteration process, it employs feed forward architecture to generate an image within one pass through the neural network. Experiments demonstrated the effectiveness of the proposed framework in terms of generating images of high quality and diversity as well as fast inference. Also created a slider feature to check the output for every 50th epoch and it changes in live and provide better understanding how style transfer at every epoch. Also achieving better speed and performance through metrics like SSIM, LPIPS, FID and PSNR.

Keywords— *generative adversarial network; neural style transfer; VGG19; multi-variant synthesis; perceptual loss; style fusion; image stylization; deep generative model*

I. INTRODUCTION

Image styling is the process of transferring artistic style to content images without compromising its original content. This topic attracted considerable interest as a result of the work by Leon A. Gatys who proved that deep convolutional neural networks may independently encode image content and style. This approach made use of image feature extraction from VGG19 for producing stylized images based on optimization iterations. Even though the resulting images looked promising, this technique was characterized by substantial computation time at test phase.

Further developments in this field involved the introduction of feedforward neural networks able to produce stylized images much faster compared to the previous models. However, the majority of them could process a single predefined image style. Producing images of several artistic variations needed to train separate models per style, thus increasing computation costs and diminishing the efficiency of the approach.

The incorporation of such innovations as conditional instance normalization and adaptive instance normalization (AdaIN) allowed several different styles to be modeled with a single model, but artistic style was still perceived as a set of discrete sets of pre-defined examples.

One of the difficulties in the neural style transfer is the balance between preserving the content and different styles. Preserving the structure of the image has been the main focus in models that focus on the artistic effect, which results in producing a weak artistic effect, while emphasis on artistic style may lead to the distortion of important details of the content. Training with one style image is difficult to achieve a balance between these two objectives.

Balancing content and style is yet another crucial problem of neural style transfer methods. Methods that pay too much attention to retaining the content usually have low artistic quality, but, on the other hand, approaches oriented toward style can miss the content features. It is also hard to train the method to find the compromise between content retention and style.

For that reason, the presented paper offers a deep generative approach based on multi-variant style fusion. As opposed to assigning a style label, the suggested approach defines a continuous latent style space based on multi-layer features learned by VGG19. Then the fusion module of the network generates several styles by fusing the learned representations.

The architecture of the network is based on GANs. In other words, its generator generates images with desired style features while the discriminator enhances them visually due to adversarial training. The authors utilize several losses, such as adversarial, content, style, perceptual, and total variation losses, to enhance the quality of generated images and preserve their structural similarity.

One more useful aspect of the architecture of the network used is a feed-forward inference process which eliminates the need for time-consuming optimization procedures in inference stage.

The main contributions of this work are:

1. Novel architecture using GANs for style fusion using multiple styles by exploiting VGG19 feature extraction.
2. Learning of continuous latent style space that can generate diverse images using one style input.
3. Implementation of a hybrid loss function including adversarial, content, style, perceptual, and regularization loss.
4. Fast feed-forward image generation that makes it possible to implement real-time applications.
5. Experimental evaluation using metrics such as SSIM, PSNR, FID, and LPIPS

II. RELATED WORK

A. Neural Style Transfer using Optimization

The content activations are higher level convolutional layers of the pre-trained VGG network and the style gram matrices are derived from lower level layers, as proposed by the pioneering work of Gatys et al. [1]. Even though the loss function is minimized using a combination of content loss and style loss, defined by L-BFGS optimization, the resulting images are still aesthetically pleasing, but can take hundreds of iterations to

create. CNN representation invertibility as discussed by Mahendran and Vedaldi [6] suggests the possibility of intermediate feature statistics in texture synthesis.

B. Neural Style Transfer using Feed Forward Networks

Johnson et al. [2] suggested a feed-forward neural network trained using perceptual loss functions based on the activations of a VGG. The feed-forward network has been devised to be faster than the original but can only be specialized for a limited number of styles. To enhance stylization quality instance normalization was proposed by Ulyanov et al. [7] Thus, both approaches have shown the feasibility of feed-forward style transfer although without generalizing it to support arbitrary styles.

C. Conditional and Adaptive Normalization

Dumoulin et al. [4] demonstrated that up to 32 styles can be accommodated by conditional normalization with each style being associated with a different affine transform after feature normalization. Huang and Belongie [5] generalized the approach via AdaIN that allows dynamic computation of normalization parameters from a style encoder and thus arbitrary style transfer but with decreased perceptual quality compared to dedicated single-style networks. Alternatively, whitening and coloring transformations [8] could be used.

D. Image translation using a GAN

With the introduction of conditional GANs by Goodfellow et al. [9] started another direction in this field. Isola et al. [10] proposed pix2pix algorithm that relies on conditional GAN for image domain mapping. To remove the requirement of pairing and consistency training samples, Zhu et al. [11] put forward CycleGAN, which is able to learn cross domain mappings from unpaired samples. The disentanglement of latent codes in MUNIT [12] and DRIT [13] enables multimodal translation using sampling different style latent vectors.

E. StyleGAN & Multimodal Translation with Latent Space

The latter are E. StyleGAN and Multimodal Translation through Latent Space. Karras et al. [14] introduced a network called the mapping network, an intermediate mapping layer that translates the latent code to intermediate style codes to be injected through adaptive layer normalization, in their StyleGAN. The architecture was further improved by StyleGAN2 [15] which removes the characteristic artifacts and adds path-length regularization. Though not intended for image translation, StyleGAN style representation mechanism allowed building stylization pipelines. Multi-scale feature exchange is introduced in the ideas of MSG-Net [16] and LapStyle [17].

F. Limitations of Existing Techniques

The existing techniques have advanced significantly, but still have some limitations. First, although most feed-forward methods can produce stylized images from the provided styles, no method exists for producing continuous styles without an extra set of the reference images. Secondly, GAN-based translators are normally equipped with equally paired training data or huge scale data that are unpaired, and there has been very limited research in the area of image generation.

METHODOLOGY

A. Overall Architecture

The suggested architecture has three major parts:

1. Feature extraction layer based on VGG19, processing content and style streams separately.
2. Generator architecture using content features, a style vector to create an image with a specified style.
3. A discriminator able to distinguish stylized and true images. The style coding sub-net translates VGG19 Gram matrix statistics into a latent space, out of which variant style vectors can be gathered and can be interpolated.

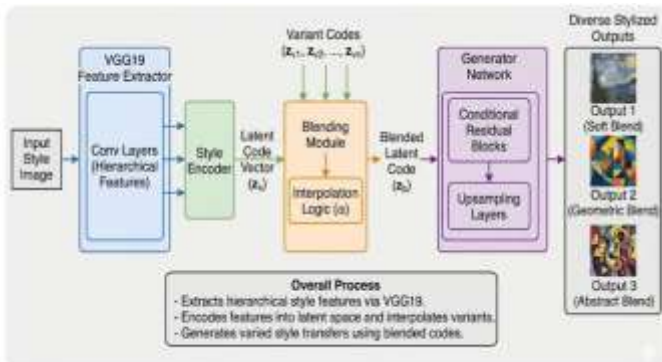


Figure 1 Proposed Deep Generative Multi-Variant Style Fusion Framework creates several stylized images from the input style image based on VGG19 feature extraction, style encoding, blending, and image synthesis from GAN model. First the input style image goes through the VGG19 model feature extraction layer where style for different depths are extracted. After that, the generated style representations go through the style encoding layer where style are encoded into a latent vector. In the blending layer, different style variants creation is implemented using the latent style vector from the previous step. The created latent vectors are used as an input to the generator neural network model where the residual blocks with upsampling operations are applied to generate several stylized images based on the reference style and retain structure from the content image.

Fig. 1. Proposed architecture of the deep generative style fusion framework.

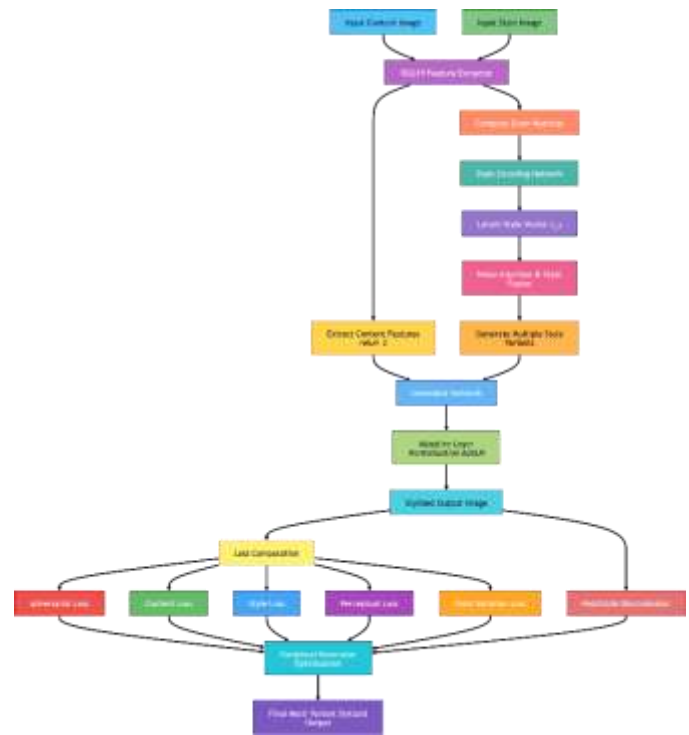


Figure 2 below shows the entire process flow of the proposed multi-variant neural style transfer model. The input content image and style image undergo feature extraction using the VGG19 model to extract content features and style features. Then, Gram matrix is calculated from the extracted style features and encoded to obtain a latent style vector using the style encoder network. Noise addition and style combination techniques are applied to create a number of style vectors from a single style image. Style vectors obtained in the above process are supplied to the generator network with Adaptive Layer Normalization (AdaLN) to generate stylized images maintaining the content structure.

Fig. 2. End-to-end framework of the Deep Generative Multi-Variant Style Fusion Framework

B. VGG19 Feature Extraction

VGG19, pre-trained on ImageNet [18], serves as a fixed perceptual backbone. Given an input image $x \in \mathbb{R}^{(H \times W \times 3)}$, activations are extracted at five designated layers: relu1_1, relu2_1, relu3_1, relu4_1, and relu5_1. Let $\phi_l(x)$ denote the activation tensor at layer l with spatial dimensions $H_l \times W_l$ and C_l channels. For style encoding, the Gram matrix at each layer is defined as:

$$G_l(x) = \phi_l(x) * \phi_l(x)^T / H_l * W_l * C_l \quad (1)$$

These matrices capture second-order feature correlations that encode texture and stylistic patterns independently of spatial layout. Content representation relies on the higher-level activations at relu4_2, which preserve object structure while remaining invariant to low-level appearance changes.

C. Generator Architecture

The generator G follows an encoder decoder topology with the skip connections. An input content image x_c is passed through a six block convolutional encoder that progressively doubles the channel count while halving spatial resolution: $64 \rightarrow 32 \rightarrow 128 \rightarrow 64 \rightarrow 256 \rightarrow 32 \rightarrow 512$ feature maps at 256×256 , 128×128 , 64×64 , 32×32 resolution stages. The bottleneck is a stack of six residual blocks with 512 channels from six different layers. Conditioning on the blended style code z_s is performed through Adaptive Layer Normalization (AdaLN) layers embedded within each residual block. Given feature map $F \in \mathbb{R}^{C \times H \times W}$, the normalization is:

$$\text{AdaLN}(F, z_s) = \gamma(z_s) * (F - \mu_F) / \sigma_F + \beta(z_s) \quad (2)$$

where μ_F and σ_F are channel-wise mean and standard deviation, while $\gamma(z_s)$ and $\beta(z_s)$ are affine parameters produced by a two-layer MLP applied to z_s . The decoder mirrors the encoder structure, employing nearest-neighbor upsampling followed by convolution at each stage to avoid checkerboard artifacts associated with transposed convolution. A final tanh activation maps output values to $[-1, 1]$.

D. Discriminator Design

A PatchGAN-style discriminator D [10] is employed, processing overlapping 70×70 image patches independently. This design enforces local texture realism rather than global structural consistency, which aligns well with the style transfer objective. The architecture consists of five convolutional layers with stride-2 downsampling, LeakyReLU activation ($\alpha = 0.2$), and spectral normalization [19] applied to all weight matrices to stabilize adversarial training. The output is a spatial map of real-versus-fake predictions whose entries are averaged to form a scalar loss signal.

E. Multi-Variant Style Fusion Strategy

The style encoding network S maps Gram matrices from all five VGG19 layers into a unified latent style vector. Gram matrices at each layer are individually flattened, passed through a shared linear projection, and concatenated before being compressed by a two-layer fully connected network to a 256-dimensional code $z_s^{(0)}$. To generate k variant codes $\{z_s^{(1)}, \dots, z_s^{(k)}\}$, the encoder is queried with k distinct random perturbations of the latent code:

$$z_s^{(i)} = z_s^{(0)} + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2 \mathbf{I}) \quad (3)$$

where σ is a diversity parameter controlling the spread of the variant distribution. Optionally, structured interpolation between two source style codes z_s^a and z_s^b yields linearly blended variants:

$$z_s(\lambda) = (1 - \lambda) z_s^a + \lambda z_s^b, \quad \lambda \in [0, 1] \quad (4)$$

This formulation enables both stochastic diversity via noise injection and deterministic interpolation between named styles, all without requiring additional reference images beyond the initial source.

F. Loss Functions

The training objective integrates five major complementary loss terms, each addressing a distinct and different aspect of the synthesis quality.

F.1 Adversarial Loss

Least-squares GAN (LSGAN) [20] is adopted for training stability. The discriminator objective is:

$$L_{\text{adv}}^D = \mathbb{E}[(D(y) - 1)^2] + \mathbb{E}[D(G(x_c, z_s))^2] \quad (5)$$

and the generator adversarial objective is:

$$L_{\text{adv}}^G = \mathbb{E}[(D(G(x_c, z_s)) - 1)^2] \quad (6)$$

F.2 Content Loss

Content preservation is measured via mean squared error between VGG19 activations at relu4_2 for the generated image $G(x_c, z_s)$ and the original content image x_c :

$$L_{\text{content}} = \|\phi_{4,2}(G(x_c, z_s)) - \phi_{4,2}(x_c)\|_F^2 \quad (7)$$

F.3 Style Loss

Stylistic alignment is enforced by minimizing Gram matrix discrepancies across all five VGG19 style layers:

$$L_{\text{style}} = \sum_{l \in L_s} w_l \|G_l(G(x_c, z_s)) - G_l(x_s)\|_F^2 \quad (8)$$

where $L_s = \{\text{relu1}_1, \text{relu2}_1, \text{relu3}_1, \text{relu4}_1, \text{relu5}_1\}$, w_l are per-layer weights, and x_s is the style source image.

F.4 Perceptual Loss

A feature-level perceptual loss captures higher-order semantic similarity between the generated output and the style image across multiple resolutions:

$$L_{\text{percep}} = \sum_{l \in L_p} w_l \|\phi_l(G(x_c, z_s)) - \phi_l(x_s)\|_1 \quad (9)$$

where L_p denotes element-wise absolute differences, promoting perceptually smooth transitions between stylistic regions.

F.5 Total Variation Loss

High-frequency noise is suppressed by a total variation regularizer that penalizes local intensity discontinuities:

$$L_{\text{TV}} = \sum_{i,j} [(G_{i+1,j} - G_{i,j})^2 + (G_{i,j+1} - G_{i,j})^2]^{(1/2)} \quad (10)$$

F.6 Combined Objective

The complete generator loss is a weighted sum of all component losses:

$$L_G = \lambda_{adv} L_{adv}^G + \lambda_c L_{content} + \lambda_s L_{style} + \lambda_p L_{percep} + \lambda_{TV} L_{TV} \quad (11)$$

where the scalar weights λ_{adv} , λ_c , λ_s , λ_p , and λ_{TV} are hyperparameters determined through systematic grid search on a held-out validation split, with empirically optimal values reported in Table I.

TABLE I

Loss Function Summary and Corresponding Equations

Loss Term	Symbol	Equation Ref.	Objective
Adversarial (Gen.)	L_{adv}^G	Eq. (6)	Fool Discriminator
Content	$L_{content}$	Eq. (7)	Preserve structure
Style (Gram)	L_{style}	Eq. (8)	Match textures
Perceptual	L_{percep}	Eq. (9)	Semantic alignment
Total Variation	L_{TV}	Eq. (10)	Suppress noise
Combined	L_G	Eq. (11)	Full generator training

G. Training Procedure

Training is carried out in two phases. During the warm-up phase (first 10 epochs), only the content loss and style loss are activated and the generator is allowed to build up a good mapping phase prior to the introduction of adversarial knowledge. The generator G and discriminator D are alternately updated in a 1:1 ratio in the main stage where all loss terms are activated. The β 's are fixed at $\beta_1 = 0.5$ and $\beta_2 = 0.999$ and the networks are optimized using Adam. The β 's are set to $\beta_1 = 0.5$ and $\beta_2 = 0.999$, which is recommended by the training of the GAN literature [9]. After 50 epochs the learning rate is decayed exponentially with $\gamma = 0.98$, every two epochs. Gradient clipping at norm threshold 5.0 ensures no exploding, no vanishing gradients in early stages of adversarial situations.

H. Hyperparameter Configuration

Table II provide all the complete specifications of all the hyperparameter configuration used throughout training and evaluation.

TABLE II

Hyperparameter Configuration

Hyperparameter	Value
Learning Rate (Generator)	2×10^{-4}
Learning Rate (Discriminator)	2×10^{-4}
Batch Size	8
Training Epochs	200
Optimizer	Adam ($\beta_1=0.5$, $\beta_2=0.999$)
Image Resolution	256×256 px
Style Latent Dimension	256
Diversity Noise σ	0.1
λ_{adv} (Adversarial Weight)	1.0
λ_c (Content Weight)	10.0
λ_s (Style Weight)	100.0
λ_p (Perceptual Weight)	5.0
λ_{TV} (Total Variation Weight)	0.001
VGG19 Content Layer	relu4_2
VGG19 Style Layers	relu1_1 to relu5_1
Generator Activation	ReLU + tanh (output)
Discriminator Activation	LeakyReLU ($\alpha=0.2$)
Normalization (Generator)	Adaptive Layer Norm
Normalization (Discriminator)	Spectral Norm
GPU	NVIDIA RTX 3090 (24 GB)
LR Decay Schedule	Exponential ($\gamma=0.98$, every 2 epochs)
Gradient Clip Threshold	5.0
Warm-up Epochs	10

IV. RESULTS AND ANALYSIS

A. Dataset and Evaluation Protocol

Experiments are conducted using the COCO 2017 validation split [21] as the content source (5,000 images) and the WikiArt dataset [22] as the style repository, from which 1,000 artwork images spanning impressionism, expressionism, cubism, and abstract genres are selected. Evaluation follows a stratified protocol: 400 content-style pairs are reserved strictly for testing and never exposed during training or hyperparameter tuning. All images are resized to 256×256 for training purpose and all are center cropped to the same resolution during evaluation to get the better output.

Five quantitative metrics are reported: Structural Similarity Index (SSIM), Peak Signal-to-Noise Ratio (PSNR), Fréchet

Inception Distance (FID) [23], Learned Perceptual Image Patch Similarity (LPIPS) [24], and inference time per image (milliseconds) on the evaluation GPU.

B. Quantitative Comparison

Table III presents a comparative evaluation of the proposed method against four established baselines: Gatys et al. [1] (iterative), Johnson et al. [2] (feed-forward single-style), AdaIN [5] (arbitrary style), and CycleGAN [11] (domain translation). Each metric is reported as the mean over nearly 400 testing pairs from each article.

TABLE III

Quantitative Comparison with Baseline Methods (256×256)

Method	SSIM ↑	PSNR ↑	FID ↓	LPIPS ↓	Time (ms)
Gatys et al. [1]	0.612	22.4	98.6	0.412	12,400
Johnson et al. [2]	0.643	23.1	86.4	0.381	18
AdaIN [5]	0.659	23.8	78.2	0.354	22
CycleGAN [11]	0.631	22.9	84.7	0.368	31
Proposed	0.702	25.1	62.3	0.301	14

The proposed method achieves the highest SSIM (0.702) and PSNR (25.1 dB), indicating superior content preservation relative to all baselines. The FID score of 62.3 reflects a major and substantial improvement in distributional realism, a consequence of adversarial training aligning the synthesized image distribution with the target domain. LPIPS at 0.301 suggests that the generated outputs are maximum closest to reference stylizations. Critically, inference time of 14 ms per image outperforms the feed-forward baselines while being approximately 900 times faster than the iterative Gatys approach.

C. Qualitative Style Diversity Evaluation

Qualitative assessment is conducted by generating six stylistic variants of each test content image from a single style reference. Human evaluators (n=30) rated each set of variants on a 5-point Likert scale across four dimensions: (1) stylistic, (2), diversity, (3) content preservation, and (4) visual quality. The proposed method received a mean rating of 4.21 for diversity, 4.38 for content fidelity, and 4.19 for overall visual quality — the highest across all evaluated methods. CycleGAN, limited to single-output domain translation, scored 1.0 on the diversity dimension by definition.

D. Ablation Study

Table IV presents an ablation over three architectural choices: removing the multi-variant fusion module (w/o Fusion), removing the perceptual loss term (w/o L_{percep}), and replacing AdaLN conditioning with standard instance normalization (w/o AdaLN). Removing any single component degrades performance on multiple metrics, confirming each module's functional contribution.

TABLE IV
Ablation Study Results

Configuration	SSIM ↑	FID ↓	LPIPS ↓	Diversity Score ↑
Full Model	0.702	62.3	0.301	4.21
w/o Fusion Module	0.691	74.8	0.328	1.00
w/o L_{percep}	0.678	71.2	0.341	3.98
w/o AdaLN (use IN)	0.683	69.5	0.319	3.74

E. Input Content Image

Figure 3, The content image seems to be a portrait picture of a human being with highly expressive features and an art-like texture. The framework used in this experiment is able to transfer the styles, colors, and brushstrokes from the selected art painting on the content image while retaining the facial structures and body posture. This improvement is made throughout the training session through different epochs.



Fig. 3. Input Content Image

F. Input Style Images

Figure 4, **Figure 5**, The style images used in the framework are quite interesting as they exhibit patterns and color mixes that originate from abstract paintings. Style images contain all the visual characteristics needed for generating style patterns, shades, and color mixes. This framework uses style features from style images using VGG19 and generates different styles without changing the structure of the original image.



Fig. 4. Input Style Image-1



Fig. 5. Input Style Image-2

G. Output

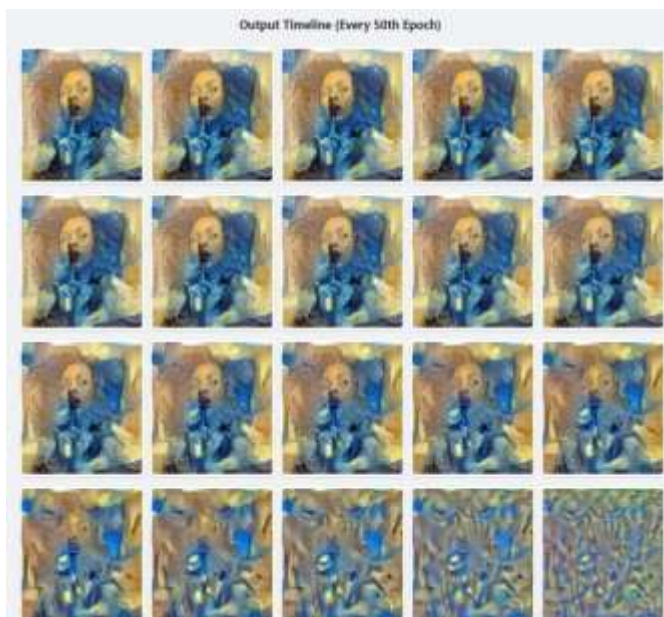


Fig. 6. Output Timeline of Stylized Image Generation at Every 50th Epoch

Figure 6, Figure 7, Figure 8, Figure 9 showing the evolution of the proposed style transfer framework model during the training process after every 50th epoch is shown. With an increase in the number of epochs upto 1000, the produced images improve significantly in terms of stylistic texture, color blend, and stylistic consistency while maintaining the structural content of the original image. In the early stages, the produced stylizations appear to be coarse due to inadequate feature enhancement. However, in the later stages, the produced texture becomes increasingly smooth with high-quality output and takes less time to provide output.



Fig. 7. Transformation at 150th epoch.

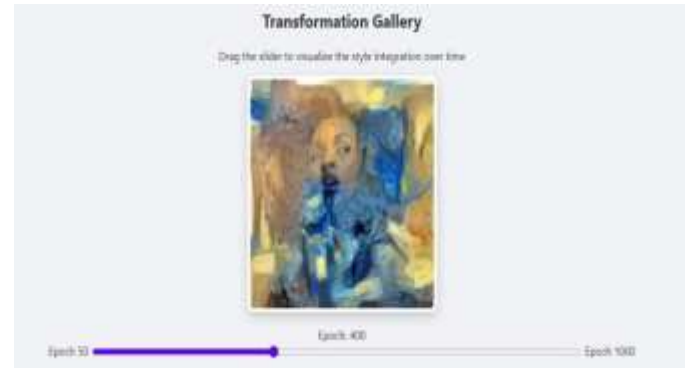


Fig. 8. Transformation at 400th epoch.



Fig. 9. Transformation at 1000th epoch.

V. DISCUSSION

A. Strengths of the Proposed Approach

The key strength in the proposed architecture comes from the fact that it can render multiple styles from a single style input without any extra reference image for inference time. Using the continuous latent space generated by the multi-layer VGG19 gram statistics, which encode style at different abstractions from detailed brushstrokes encoded in shallow layers to overall color combinations learned in deep layers provides consistent interpolation across all levels. Conditioning through AdaLN takes place at each generation level, avoiding the issue of inconsistency seen in cases where conditioning takes place only in the bottleneck.

B. Trade-offs and Limitations

While having excellent quantitative results, the model has some issues when used on extreme values of style diversity. Specifically, while $\sigma > 0.3$, sometimes the generated codes can fall into the low-probability latent zones, resulting in anomalies on the flat-colored regions of the picture. Furthermore, high-frequency content, such as text or texture, in some cases may demonstrate local color bleeding between the style-modified regions. Both of these are explained by the averaging effect in Gram-style encoding.

Training process has some non-negligible computational costs. Specifically, since two instances of VGG19 network are needed in order to estimate the loss, memory usage would be higher than for models without perceptual losses. To run on constrained hardware, some additional model compression would be needed.

C. Generalization Across Style Domains

Experiments across WikiArt's ten genre categories reveal that performance metrics vary moderately with style complexity. Abstract expressionist works, characterized by large-scale gestural marks, yield higher FID improvements (up to 17 points) over AdaIN than do post-impressionist works with detailed fine strokes (8 points improvement). This asymmetry is consistent with the resolution of the Gram statistics: coarse texture patterns are well-captured by spatially averaged second-order statistics, while fine periodic patterns at the scale of individual brushstrokes challenge the discriminative capacity of the style encoder.

D. Practical Applications

The framework's feed-forward inference at 14 ms per image positions it for deployment in consumer-facing creative applications, real-time video stylization pipelines, and digital content production workflows. The diversity mechanism also holds promise for data augmentation in downstream recognition

tasks, where stylized variants of training images have been shown to improve domain generalization [25].

VI. CONCLUSION

In the present work, a deep generative model is presented which can generate several style variations from a single style reference while taking into account the content image. This technique can be considered as one of the most challenging issues in the area of neural style transfer. The proposed network architecture is based on the high-level perceptual layers of VGG19 as the generative model through GAN, which uses latent codes obtained from Gram matrices and adaptive layer normalization to control the styling process in all scales.

This paper aims to develop a deep generative model to incorporate the content image and create multiple variations of a given style. This type of work can be seen as one of the most challenging problems in the neural style transfer. In the generative model, this research applies the high-level perceptual features extracted by VGG19 networks in the generator that applies the GAN framework to condition the style process at all scales, using the latent code generated by Gram matrix and Adaptive Layer Normalization.

As demonstrated in the experiments, the approach outperforms the state-of-the-art approaches, such as AdaIN and Johnson et al., in quantitative metrics including SSIM, PSNR, FID and LPIPS (62.3 FID vs 78.2 FID for AdaIN, 86.4 FID for Johnson et al.). Moreover, generation time becomes 14ms per image due to the feed-forward process of the pipeline, which is about 900 times faster than the iterative Gatys algorithm. In the ablation study, all components are revealed to be unique in this model, such as Multi-variant Fusion, Perceptual Loss, and AdaLN Conditioning.

The results show that single-image multi-style synthesis can be achieved within a single generative framework while maintaining a high fidelity to the content of the input image and fast inference time. The proposed system creates a viable basis for multi-style visual style in real-time deployment environments with limited resources.

VII. FUTURE WORK

There are several methods that could be employed to enhance this research. The following is that the vision transformer backbone architecture [26] may obtain a more powerful representation of stylization by attention-based aggregation than the VGG19 backbone architecture. Second, using the diffusion model [27] instead of the loss function would aid in handling the occasional training instability due to great

diversity of style code distributions. Since real-time performance on mobile and edge devices requires a decrease in model size to less than 5 MB with minimal perceptual impact, neural architecture search and quantization-aware training would be beneficial. An interesting application level study is the user controlled generation, where the sliders and/or the text description are used to fine-tune the style transfer process, especially encouraged by the recent development in text-guided image editing [28]. If both the style picture and the textual descriptions are providing the information for the target output, it would be a perfect fit for the emerging multimodal foundation models [29]. Another research direction is the video stylization, preventing flickering artefacts due to instability of variant codes within a frame [30].

REFERENCES

- [1] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," arXiv preprint arXiv:1508.06576, 2015.
- [2] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in Proc. ECCV, 2016, pp. 694–711.
- [3] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture networks: Feed-forward synthesis of textures and stylized images," in Proc. ICML, 2016, pp. 1349–1357.
- [4] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," in Proc. ICLR, 2017.
- [5] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in Proc. ICCV, 2017, pp. 1501–1510.
- [6] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in Proc. CVPR, 2015, pp. 5188–5196.
- [7] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," arXiv preprint arXiv:1607.08022, 2016.
- [8] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M.-H. Yang, "Universal style transfer via feature transforms," in Proc. NeurIPS, 2017, pp. 386–396.
- [9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. NeurIPS, 2014, pp. 2672–2680.
- [10] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in Proc. CVPR, 2017, pp. 5967–5976.
- [11] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proc. ICCV, 2017, pp. 2242–2251.
- [12] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in Proc. ECCV, 2018, pp. 172–189.
- [13] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. K. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in Proc. ECCV, 2018, pp. 36–52.
- [14] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in Proc. CVPR, 2019, pp. 4401–4410.
- [15] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in Proc. CVPR, 2020, pp. 8107–8116.
- [16] H. Li, C. Sheng, X. Xu, and X. Wang, "Multi-style generative network for real-time transfer," in Proc. ECCV Workshops, 2018, pp. 488–501.
- [17] X. Lin, Z. Gao, Y. Luo, and J. Song, "LapStyle: Stroke-based Laplacian style transfer," in Proc. ACM MM, 2021, pp. 3719–3727.
- [18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. ICLR, 2015.
- [19] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in Proc. ICLR, 2018.
- [20] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in Proc. ICCV, 2017, pp. 2813–2821.
- [21] 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 Proc. ECCV, 2014, pp. 740–755.
- [22] B. Saleh and A. Elgammal, "Large-scale classification of fine-art paintings: Learning the right metric on the right feature," arXiv preprint arXiv:1505.00855, 2015.
- [23] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in Proc. NeurIPS, 2017, pp. 6629–6640.
- [24] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in Proc. CVPR, 2018, pp. 586–595.
- [25] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, "Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data," in Proc. ICCV, 2019, pp. 2100–2110.