Style Transfer using Neural Networks

Avadhesh Sisodiya, Ketupati Swargiary, Khushi Bhatt, Naman Nirbhaya, Satyam Kumar
Indian Institute of Technology, Guwahati

***Abstract*—Gatys et al. introduced a neural algorithm that renders a content image in the style of another image, achieving so-called style transfer. However, their framework requires a slow iterative optimization process, which limits its practical application. Fast approximations with feed-forward neural networks have been proposed to speed up neural style transfer. But the speed improvement comes at a cost; the network is usually tied to a fixed set of styles and cannot adapt to arbitrary new styles. AdaIN (Adaptive Instance Normalization), proposed by Huang and Belongie, overcame the limitations of previous methods by allowing real-time style transfer for arbitrary styles. In this report, we implement this method to perform arbitrary style transfer in real-time. In addition, we introduce a modification in the AdaIN method by integrating a standard Gram matrix style loss func- tion, which results in improved style consistency and capturing of complex textures and patterns.**

***Index Terms*—Style transfer, Deep Neural Networks, Adaptive instance normalization, Gram matrix.**

I. Introduction

The pioneering work of Gatys et al. demonstrated that deep neural networks (DNNs) are capable of encoding both the content and style of an image, revealing that these two aspects can be separately manipulated. This discovery led to a flexible neural style transfer approach that can fuse the content of one image with the style of another. However, Gatys et al.'s optimization-based method is computationally intensive, making it slow for practical applications. To address this, researchers have worked to accelerate style transfer, with some approaches training feed-forward neural networks to perform stylization in a single pass. Although effective, these methods are typically limited to a predefined set of styles. More recent techniques attempt to allow for a broader range of styles but are either constrained to specific options or fail to match the efficiency of single-style methods. AdaIN (Adaptive Instance Normalization), proposed by Huang and Belongie, overcame the limitations of previous methods by allowing real-time style transfer for arbitrary styles. The core idea behind AdaIN is to align the mean and variance of the content features with those of the style features. This allows the system to combine any content with any style, while still being computationally efficient. Limitations of this method include reduced texture fidelity, inflexibility for certain artistic styles and potential-oversimplification.

In this report, we introduce a modification in the AdaIN method by integrating a standard Gram matrix style loss function. This results in improved style consistency and capturing of complex textures and patterns.

A. Background

In instance normalization (IN), each feature map is normalized independently by subtracting its mean and dividing by its standard deviation:

$$IN(x) = \gamma(\frac{x - \mu(x)}{\sigma(x)}) + \beta$$

Here, here $\mu(x)$and $\mu(y)$ are computed across spatial dimensions independently for each channel and each sample. AdaIN goes a step further by adapting these statistics based on the style image. AdaIN receives a content input x and a style input y, and simply aligns the channel wise mean and variance of x to match those of y. Unlike instance normalization, AdaIN has no learnable affine parameters. Instead, it adaptively computes the affine parameters from the style input:

$$AdaIN(x, y) = \sigma(y)(\frac{x - \mu(x)}{\mu(x)}) + \mu(y)$$

in which we simply scale the normalized content input with y, and shift it with y. Similar to IN, these statistics are computed across spatial locations. This alignment is done per feature map channel. The result is that the feature representation of the content image is shifted to mimic the style image's texture and style patterns, while still retaining the overall structure of the content image. This method achieves a substantial speedup, making it nearly three orders of magnitude faster than the original approach while retaining the flexibility to apply any new style. In short, AdaIN performs style transfer in the feature space by transferring feature statistics, specifically the channel-wise mean and variance. The AdaIN layer is as simple as an IN layer, adding almost no computational cost.

A simple architecture is followed where the content and style images are passed through a pre-trained VGG-19 network first. The encoder- extracts feature representation from content and style images. AdaIN layer performs the AdaIN operation. After this, a decoder network reconstructs the image in the style of the style image. The decoder uses no normalization layers to allow flexibility across various styles. It also allows for interactive control over the style transfer process.

B. Challenges and Motivation

Style transfer has many exciting applications across various fields like art and design, medical imaging, marketing and augmented reality. Many social media and photo-editing apps, use style transfer for real-time filters that apply various styles to users' photos and videos. Style transfer can be applied in medical imaging to standardize images from different imaging devices, which helps improve image clarity, consistency, and diagnostic accuracy, especially in training models for medical diagnoses. It helps create immersive experiences by rendering scenes in the style of certain environments or artistic themes, enhancing VR/AR applications for games, training simulations, or storytelling. It can be used to create content for

marketing. Game developers can us style transfer to transform 3D scenes or sprites into particular visual styles and make the games visually distinctive without designing assets manually. It can be used to restore old or damaged film footage or photos by transferring the style of high-quality visuals to the degraded areas, which preserves aesthetic qualities while improving resolution and clarity.

II. Related Works

Style Transfer, This field of style transfer has roots in non-photo-realistic rendering, as first explored in early works on texture synthesis and transfer. Initial methods, such as histogram matching on linear filter responses and non-parametric sampling, relied on low-level image statistics. While effective in some cases, these techniques often failed to capture the broader semantic structures needed for high-quality style transfer. A significant breakthrough came with the work of Gatys et al., who demonstrated that matching feature statistics within the convolutional layers of a deep neural network (DNN) could yield impressive style transfer results.

Building upon this foundation, several enhancements to Gatys et al.'s approach have been proposed. For example, Li and Wand introduced a framework using a Markov random field (MRF) in deep feature space, allowing for more robust local pattern matching. Other methods, such as those from Gatys et al., offered greater control over style elements like color preservation, spatial location, and scale. Addressing video style transfer, Ruder et al. introduced temporal constraints to improve the quality and consistency across frames.

However, a primary limitation of Gatys et al.'s method is its reliance on a slow optimization process, where the image is iteratively adjusted to minimize content and style losses computed by a loss network. Although effective, this approach can be too slow for real-time applications, particularly on devices with limited processing power. To overcome this, many researchers shifted to feed-forward networks trained to minimize similar objectives. These networks achieve style transfer in a single pass, significantly increasing speed—by roughly three orders of magnitude—enabling real-time applications.

Feed-forward approaches have continued to evolve, with innovations aimed at increasing style flexibility. For instance, Wang et al. introduced a multi-resolution architecture to enhance the detail and granularity of feed-forward style transfer, while Ulyanov et al. explored methods to boost sample quality and diversity. Despite these advances, most feed-forward models remain tied to a fixed set of styles, limiting their adaptability. Dumoulin et al. addressed this by designing a network capable of encoding up to 32 styles and their interpolations, and Li et al. later expanded on this with an architecture that could synthesize hundreds of textures and transfer multiple styles. However, these methods are still unable to generalize to arbitrary, unseen styles.

More recently, Chen and Schmidt proposed a style swap layer that allows for arbitrary style transfer within a feed-forward framework. By matching feature activations between content and style images in a patch-wise manner, their style swap layer enables greater flexibility. However, this approach introduces a new computational bottleneck, as the style swap operation consumes over 95

Style Loss Functions the Selecting an effective style loss function is crucial in style transfer. Gatys et al.'s original method achieved style matching by aligning the second-order statistics of feature activations, as represented by the Gram matrix. Since then, other loss functions have been explored to improve style transfer results. These include MRF loss, adversarial loss, histogram loss, CORAL loss, maximum mean discrepancy (MMD) loss, and metrics based on the distance between channel-wise mean and variance. While these functions vary in approach, they all share the common goal of aligning feature statistics between the style and synthesized images.

A. Deep Learning-based

Deep Generative Image Modeling. Beyond style transfer, several frameworks have been developed for generating images through deep learning, including variational autoencoders (VAEs), autoregressive models, and generative adversarial networks (GANs). GANs, in particular, have achieved striking visual quality in image generation tasks. They have seen various improvements over time, such as conditional generation, multi-stage processing, and optimized training objectives. GANs have also been applied to style transfer, expanding the potential of cross-domain image synthesis through high-quality generative model. now in this we have revised the loss function through gram matrix. Gram Matrix is computed by taking dot products across all feature vectors in a layer, producing a C×C matrix that encodes feature correlations without spatial information.

B. Our Contributions

Revised Style Loss: In style transfer, the dot product serves as a key measure for similarity between feature vectors, capturing how certain features co-occur within an image. This measure is essential for understanding texture relationships, which form the basis of style transfer, focusing on patterns and textures rather than spatial layout. The dot product between feature vectors highlights style characteristics, such as brush strokes and color distributions. To formalize these relationships, a Gram Matrix is constructed by taking dot products across all feature vectors in a layer, producing a C×C matrix that encodes feature correlations without spatial information. This allows the Gram Matrix to capture the essence of an image's style while disregarding specific positional details.

Style Loss is then defined as the Mean Squared Error (MSE) between the Gram matrices of the style and generated images, ensuring that the generated image mirrors the texture and pattern qualities of the style image. Minimizing Style Loss guides the generated image to adopt these stylistic features while retaining its own spatial structure, allowing for a seamless blend of content and style. This optimization approach effectively transforms the generated image, aligning its visual features with the textures and patterns of the chosen style.

III. Proposed Work

We propose a modification in this process by integrating a standard Gram matrix style loss function. Gram Matrix is computed by taking dot products across all feature vectors in a layer, producing a CC matrix that encodes feature correlations without spatial information.

A. Network Architecture

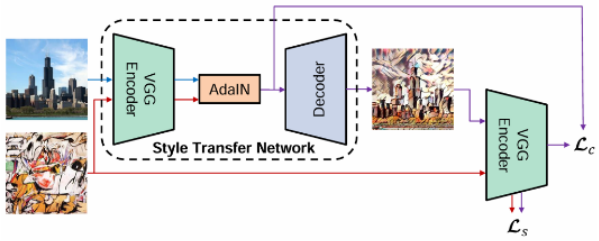
Our style transfer network T takes a content image c and an arbitrary style image s as inputs, and synthesizes an output image that recombines the content of the former and the style of the latter. We adopt a simple encoder-decoder architecture, in which the encoder f is fixed to the first few layers (up to relu4_1) of a pre-trained VGG-19. After encoding the content and style images in feature space, we feed both feature maps 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 t:

$$t = AdaIN(f(c), f(s))$$

A randomly initialized decoder g is trained to map t back to the image space, generating the stylized image $T(c, s)$:

$$T(c, s) = g(t)]$$

Figure 1. An overview of our style transfer algorithm.



B. Cost Functions

Content loss:

$$\Theta(content) = (F(g(t)) - t)^2$$

where $\Theta(content)$ is the feature representation of the content and $g(t)$ is the stylized image.

Style loss:

$$\Theta(style) = \sum_{i=1}^{l} \Big(\|\mu(\phi_i(g(t))) - \mu(\phi_i(s))\|^2 + \|\sigma(\phi_i(g(t))) - \sigma(\phi_i(s))\|^2 \Big)$$

where μ and σ are the mean and variance, and θ are the feature maps of the network at different layer

Total loss:

$\Theta = \Theta(content) + \lambda\Theta(style)$

where λ controls the balance between content and style

Now in this we have revised style loss , In style transfer, the dot product serves as a key measure for similarity between feature vectors, capturing how certain features co-occur within an image. This measure is essential for understanding texture relationships, which form the basis of style transfer, focusing on patterns and textures rather than spatial layout. The dot product between feature vectors highlights style characteristics, such as brush strokes and color distributions. To formalize these relation- ships, a Gram Matrix is constructed by taking dot products across all feature vectors in a layer, producing a C×C matrix that encodes feature correlations without spatial information. This allows the Gram Matrix to capture the essence of an image's style while disregarding specific positional details

IV. Experimental Details

A. Datasets and Training Details

The revised model underwent training for 10000 iterations, aiming to effectively transfer artistic styles onto content images. For this task, 1,019 content images were sourced from the ImageNet dataset, accessible via Kaggle, while a selection of style images, including various paintings and artwork, provided the artistic textures and patterns. The model was optimized using the Adam optimizer, which allowed for efficient and stable updates across iterations.

At the end of training, the model achieved a final total loss of 0.7213, indicating reasonable success in replicating style elements onto the content images. The model demonstrated a noticeable degree of style transfer, capturing the intended textures and patterns from the style images onto the content images. However, further refinement in style transfer might be achieved by expanding the content dataset and extending the training duration, which could allow the model to capture even finer details and more complex style characteristics.

Increasing the number of content images and training iterations would provide the model with more examples and time to learn nuanced relationships between content and style. This, in turn, could enhance the quality and fidelity of style transfer, helping the generated images appear even closer to the desired artistic effect while preserving essential content structure.

B. Baseline Methods

In style transfer models, adaptive instance normalization (AdaIN) is often used to align the mean and standard deviation of features in the generated image with those of the style image. This technique allows the model to adaptively shift the features in a way that captures the unique textures and color patterns of the target style. AdaIN adjusts the intermediate feature maps by normalizing them to match the statistics of the style image, enhancing the stylistic transfer while preserving the original content structure.

The model's loss function is designed to guide this transformation through a Total Loss, which is the sum of Content Loss and Style Loss. The Total Loss can be defined as:

$$TotalLoss = ContentLoss + \lambda(StyleLoss)$$

where λ is a weighting factor that controls the influence of the style. Style Loss is calculated as the average Mean Squared Error (MSE) between the Gram matrices of feature maps extracted from different layers of both the generated and style images. This comparison of Gram matrices, which capture

Figure 2. Example style transfer results. All the tested content and style images are never observed by our network during training.

feature correlations, helps the model reproduce the stylistic elements such as brush strokes and color tones, while Content Loss ensures the structural integrity of the original content.

V. Results

A. Comparison with State-of-the-art Methods

1) Qualitative Analysis: In the Figure 2., it shows comparison qualitatively our model is performing better than AdaIn. In previous model we can clearly see there is , Reduced Content Clarity: here style is more prominent, the content features of the original image may become less distinguishable. This can lead to a visually compelling style but may obscure key aspects of the content, making it harder to recognize objects or structures in the original image. But our model has solved this problem.

VI. Justification and Discussion

A. Failure Cases

Limitations in Style Diversity: Although AdaIN enables arbitrary style application, it still depends on the statistical alignment of features, which might not capture the nuanced characteristics of very diverse or complex styles. This could result in less stylistic richness for highly detailed or non-standard artistic styles.

Dependency on Pretrained Networks: This method often requires a pretrained encoder-decoder structure, which could limit flexibility in customization. If an application demands specialized modifications or higher artistic control, adapting these models might require complex retraining or fine-tuning.

Possible Overfitting to Training Data: If AdaIN-based models are trained on a limited range of style images, they may not generalize as effectively to styles beyond the training data, resulting in visual artifacts or lack of consistency in outputs when novel styles are applied.

VII. Conclusion

In this project, we explored the application of Neural Style Transfer (NST), a method for transferring the style of one image to the content of another. The technique leverages Adaptive Instance Normalization (AdaIN), a powerful approach for real-time style transfer . This real-time processing was enhanced by integrating a standard Gram matrix style loss function to better capture the textures and patterns inherent in artistic styles.

Our study demonstrated that the Gram matrix, calculated through the dot product of feature vectors, effectively captures the correlations between features, enabling the preservation of complex textures and patterns. The Style Loss, defined as the Mean Squared Error (MSE) between the Gram matrices of the style and the generated images, ensures that the generated image retains the unique texture, patterns, and ”feel” of the style, while maintaining the content’s spatial layout.

Key findings from our work include:

1.The Gram matrix loss is particularly effective in capturing intricate details such as brush strokes and repetitive textures.

2.By focusing on feature correlations across layers, our approach ensures improved style consistency, keeping the global structure of the style intact, regardless of the content’s spatial configuration.

VIII. References

1) X. Huang and S. Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. *arXiv:1703.06868*, 2017.
2) N. Chung, X. Rong, S. Li and Z. Wenjun. Improving Semantic Style Transfer Using Guided Gram Matrices. *10.1007/978-981-13-8138-6_14*, 2019.
3) J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
4) M.Elad and P. Milanfar. Style-transfer via texture-synthesis. *arXiv preprint arXiv:1609.03057*, 2016.
5) Surreal Symphonies (A Dataset Of Diverse Art): https://www.kaggle.com/datasets/cyanex1702/surreal-symphonies-a-dataset-of-diverse-art
6) ImageNet-Sketch Dataset: https://www.kaggle.com/datasets/wanghaohan/imagenetsketch