Balancing Content Retention In Neural Style Transfer

A THESIS

submitted by

YASH AGRAWAL

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

Electronics & Communication Engineering
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

May 2025

THESIS CERTIFICATE

This is to certify that the thesis titled **Balancing Content Retention In Neural Style Transfer**, submitted by **Yash Agrawal**, to the INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY, DELHI, for the award of the degree of **Master of Technology**, in Electronics & Communication Engineering with specialization in Machine Learning, is a bonafide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Vinayak Abrol
Thesis Supervisor
Assistant Professor
Dept. of Computer Science Engineering
IIIT Delhi, 110020

Place: New Delhi
Date: May 19, 2025

ACKNOWLEDGEMENTS

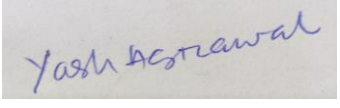
I am immensely grateful to the Computer Science Engineering Department and the Infosys Centre for AI (CAI) at IIIT Delhi. Their generous provision of computational resources and financial support has been crucial to my research.

I extend my deepest thanks to my lab, Cross-Caps Laboratory (I doubt I'll ever have a better desk to sit and work at), and my advisor, Dr. Vinayak Abrol, whose guidance, not only as an advisor but also as a mentor, has been invaluable. Dr. Abrol has been a mentor like no other, supporting me with not just academic guidance but also ensuring that I was motivated throughout my thesis. His boundless enthusiasm for mathematical constructs along with his unparalled work ethic and vast knowledge has been a constant source of inspiration and motivation for me. I could not have asked for any better.

I am especially grateful to my friends who convinced me to opt for thesis and then supported me through my countless rants and frustrations as I navigated complex challenges in my research. Balancing the rigorous coursework and thesis work at IIITD was particularly tough in the beginning, but their support and care were crucial to maintaining my health and managing my workload effectively.

I am also thankful to my parents and my brother, whose unwavering support has been my cornerstone in my master's journey.

Thank you all once again for your invaluable support and belief in my work.



ABSTRACT

KEYWORDS: Convulutional Neural Networks; Style Transfer; Encoder & Decoder; AdaIn; AdaAttN; SANet; MCCNet; Feature Transformation Blocks; Content Loss; Style Loss; Identity Loss; Contrastive Loss

This thesis work focuses on advancements in neural style transfer, a process that enables the blending of content and style features to generate stylized images. It explores feature extraction using two encoders: a VGG19-based encoder and a GLOW-based encoder, the latter improving image reconstruction and reducing content leakage through reversible transformations. Various feature fusion techniques are examined, including Adaptive Instance Normalization (AdaIN), Adaptive Attention Normalization (AdaAttN), Self-Attention Network (SANet), Multi-Channel Correlation Network (MCCNet), and Exact Feature Matching, leveraging statistical matching and attention mechanisms. The study also evaluates the impact of different loss functions such as content loss, style loss, identity loss, and contrastive loss on the quality of the output. Custom transformation blocks are introduced, combining methods like feature concatenation, AdaIN with alternative normalizations, and GLOW-based encoders enhanced with attention modules. Existing architectures, such as AdaIN and Exact Feature Matching, are further refined by integrating additional losses to enhance stylization fidelity and preserve content.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

Abbreviation	Description
AdaIN	Adaptive Instance Normalization
SANet	Style-Attention Network
MCCNet	Multi-Channel Correlation Network
EFM	Exact Feature Matching
ArtFlow	Reversible Flow-Based Style Transfer
VGG	Visual Geometry Group Network
ReLU	Rectified Linear Unit
Sigmoid	Sigmoid Activation Function
Encoder	Network that extracts feature representations
Decoder	Network that reconstructs the stylized image
Gram Matrix	Matrix representation of feature correlations
Attention	Mechanism to focus on specific regions
Normalization	Adjusting data distribution to a standard range
Activation Function	Function that introduces non-linearity in neural networks

NOTATION

$L_{content}$	Content Loss
L_{style}	Style Loss
$L_{identity}$	Identity Loss
$L_{contrastive}$	Contrastive Loss
L_{adv}	Adversarial Loss
L_{TV}	Total Variation Loss
μ	Mean of feature map
σ	Standard Deviation of feature map
ϕ	Activation Function
ϑ	Model Parameters
δ	Loss Coefficient
γ	Learning Rate Decay
α	Content-Style Weight Coefficient
β	Style-Content Weight Coefficient
λ	Regularization Coefficient

CHAPTER 1

INTRODUCTION

1.1 General Overview

Neural networks, a subset of machine learning, have fundamentally transformed how we interact with technology. Originating from the desire to mimic the human brain's architecture and functionality, neural networks have evolved through the years to become a cornerstone of deep learning. This evolution has been marked by the development from simple perceptrons to complex architectures like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), which are capable of processing spatial and sequential data, respectively. Deep learning, which involves neural networks with multiple layers (hence "deep"), has significantly enhanced the capabilities of these models, leading to breakthroughs across various domains such as image recognition, natural language processing, and autonomous driving. The ability of deep learning to extract patterns from large datasets and make intelligent predictions has not only advanced scientific research but also transformed industries by driving innovations that were once thought impossible.

Neural style transfer techniques revolve around the challenge of effectively extracting and blending content and style features to generate visually compelling stylized images. This process leverages powerful feature extractors such as VGG19 and GLOW encoders, which enhance image reconstruction quality and reduce content leakage through reversible transformations. Various feature fusion methods—such as Adaptive Instance Normalization (AdaIN), Adaptive Attention Normalization (AdaAttN), and Self-Attention Networks (SANet)—exploit statistical matching and attention mechanisms to achieve flexible and high-fidelity style blending. Furthermore, the incorporation of diverse loss functions, including content loss, style loss, identity loss, and contrastive loss, helps balance the trade-off between preserving semantic content and achieving faithful stylization, thereby improving both the aesthetic quality and structural coherence of the generated images

Recent advancements also explore custom transformation blocks that combine feature concatenation, normalization techniques, and attention modules to further refine style transfer performance. These enhancements build upon and extend foundational architectures, improving stylization fidelity while preserving detailed content structures. Overall, the evolution of neural style transfer methods reflects a continuous effort to enhance visual quality, efficiency, and adaptability, enabling more precise control over style attributes and paving the way for novel applications in image synthesis and artistic rendering. This thesis explores how we can improve style transfer using different transformation and fusion modules alongside modifications to loss functions.

1.2 Setting up the Flow

In the forthcoming sections, we define the objectives of the thesis and develop a series of research questions that address key challenges and gaps in neural style transfer. We then review current state-of-the-art style transfer architectures and transformation modules to understand their strengths and limitations. Subsequently, we delve into the motivations behind improving style-content trade-offs, computational efficiency, and generalization to unseen styles.

Next, we discuss convolutional neural networks (CNNs) as the foundational technology behind style transfer. We will examine the seminal paper on neural style transfer to gain a thorough understanding of the style transfer pipeline and the essential components involved.

Chapter 3 covers the core components of style transfer pipelines, including encoders, decoders, and transformation modules. Chapter 4 focuses on the various loss functions used to guide and optimize style transfer. Chapter 5 explores prominent architectures within the style transfer community. Chapter 6 presents the experimental setup, datasets, results, and evaluation metrics used to assess the proposed methods.

Finally, the thesis concludes with a comprehensive summary of research questions, objectives, and hypotheses. We synthesize findings across chapters to validate our central contributions, discuss limitations, and propose future directions for advancing neural style transfer in image and video domains.

1.3 Objectives of the Thesis

1.3.1 The Research Gap

While neural style transfer (NST) has seen significant progress since its inception, several critical challenges remain unresolved. Current research largely focuses on improving either stylization quality or computational efficiency, but rarely both simultaneously. The trade-off between preserving content structure and applying artistic style continues to hinder the production of visually coherent outputs. Moreover, existing models often struggle to generalize effectively to unseen styles without retraining, limiting their adaptability in practical applications. Real-time style transfer methods face difficulties in maintaining quality, especially when applied to high-resolution images and videos, where temporal consistency and computational complexity become major bottlenecks. Despite various architectural and loss function innovations, a comprehensive solution that balances style fidelity, content preservation, and inference speed is yet to be established. This research aims to address these gaps by exploring novel transformation and fusion modules alongside improved loss formulations, pushing the boundaries of NST's applicability and performance.

1.3.2 Research Questions

The investigation into advancements in neural style transfer (NST) is guided by several pivotal research questions aimed at uncovering the potential to improve the balance between content preservation and style adaptation, computational efficiency, and generalization capabilities. These questions are structured to direct the research toward practical, measurable outcomes that validate the efficacy of novel transformation modules, loss functions, and architectures within the NST domain.

Can neural style transfer models generalize effectively to unseen styles without requiring retraining? This question seeks to determine whether proposed methods can enable models to adapt to arbitrary new styles efficiently, overcoming the current limitations of fixed-style models that lack robustness and scalability.

How can neural style transfer methods be optimized for real-time performance

while maintaining high-quality stylization? Focusing on computational efficiency, this question examines the trade-offs between inference speed and visual quality, particularly for applications involving high-resolution images and video sequences.

What novel loss functions can be designed to better balance the trade-off between content structure retention and style application? This question investigates the development and integration of advanced loss formulations, such as identity and contrastive losses, to improve stylization fidelity without compromising content details.

How can different transformation and fusion modules be leveraged to enhance selective and localized style transfer? Addressing the architectural aspect, this question explores the potential of attention mechanisms and adaptive normalization techniques to achieve more precise control over stylization across different image regions.

Can multimodal inputs or advanced architectures, such as transformers or diffusion models, further improve the quality and flexibility of style transfer? This research question considers the integration of additional input modalities and novel deep learning frameworks to push the boundaries of user control and creative expression in NST.

1.4 Style Transfer: A little literature Survery

Neural style transfer (NST) has progressed significantly since the pioneering work of Gatys et al. (1), who demonstrated how convolutional neural networks (CNNs) can disentangle and recombine content and style from images. Huang and Belongie (2) introduced Adaptive Instance Normalization (AdaIN), enabling real-time arbitrary style transfer by aligning feature statistics, while Li et al. (4) proposed Whitening and Coloring Transforms (WCT) for improved feature distribution alignment. To capture finer content-style correspondences, Park and Lee (7) developed Style-Attentional Networks (SANet) utilizing self-attention mechanisms, and Liu et al. (10) further refined attention with the AdaAttN module. Flow-based reversible architectures such as ArtFlow (8) addressed content leakage by preserving content details through invertible neural flows. Zhang et al. (11) introduced Exact Feature Distribution Matching (EFDM), leveraging histogram matching to enhance style consistency and reduce artifacts. Multi-scale zero-shot style transfer was explored by Sheng et al. (6), while Chen et al. (9) in-

corporated identity and contrastive loss to balance content fidelity and style diversity. Recent efforts have extended NST to video synthesis with deflickering techniques (13) and diffusion-based style transfer models (12), expanding the applicability of NST in dynamic and multimodal contexts. Collectively, these works reflect a rich and rapidly evolving landscape of NST research aimed at improving stylization quality, computational efficiency, and generalization to diverse styles

1.5 Motivation: Why Style Transfer?

Neural style transfer (NST) has emerged as a powerful technique that bridges the gap between artistic expression and automated image synthesis. Despite rapid advancements, existing NST methods face significant challenges in balancing content preservation with effective style application, achieving real-time performance, and generalizing to unseen styles. These limitations restrict the practical use of NST in dynamic and high-resolution scenarios such as video processing and augmented reality. This thesis focuses on addressing these challenges by exploring novel transformation modules, fusion strategies, and loss functions, aiming to enhance both the visual quality and computational efficiency of style transfer. By improving the adaptability and fidelity of NST models, this research contributes to expanding their applicability across creative industries, interactive media, and real-world multimedia applications.

CHAPTER 2

Style Transfer Preliminaries

2.1 Convolutional Neural Networks (CNNs)

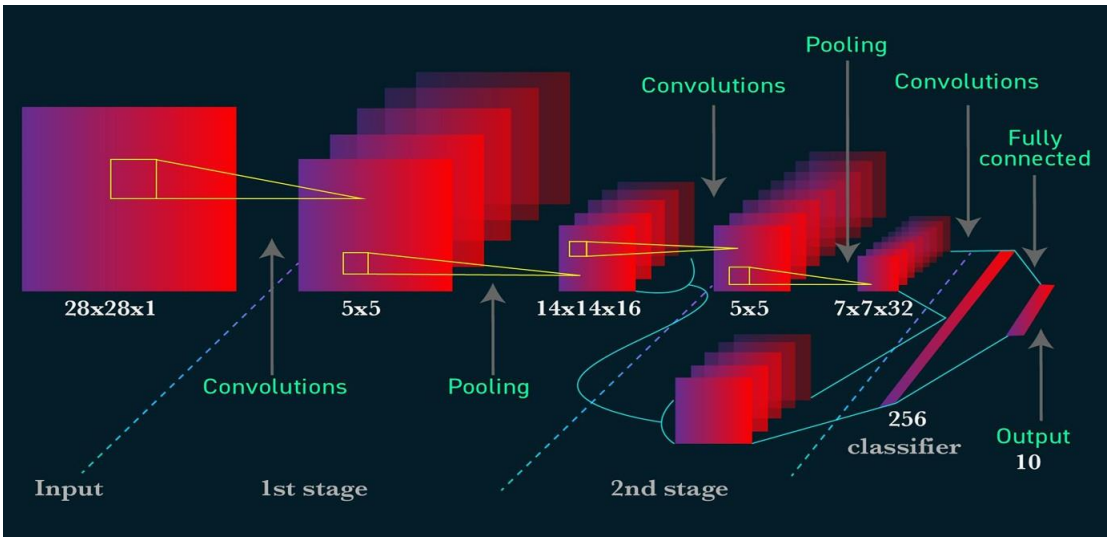


Figure 2.1: Illustration of a Convolutional Neural Network (CNN) architecture showing the input layer, convolutional stages, pooling layers, fully connected layers, and the output layer. Best viewed in color.

Convolutional Neural Networks (CNNs) are a specialized class of artificial neural networks that excel in processing data with grid-like structures, such as images. As shown in Figure 2.1, CNNs are designed to automatically and adaptively learn spatial hierarchies of features through the use of convolutional layers, pooling layers, and fully connected layers. The primary components and functionalities of CNNs are as follows:

Convolutional Layers: The fundamental building block of CNNs is the convolutional layer, which applies filters to input data to extract spatial features. Each filter, also known as a kernel, slides over the input data to produce a feature map that highlights specific patterns such as edges, textures, or shapes. In the architecture depicted in Figure 2.1, the input image is of size $28 \times 28 \times 1$, representing a grayscale image. Initially, filters of size 5×5 are applied, generating a set of feature maps. These feature maps are further processed through subsequent convolutional stages, extracting more complex patterns and hierarchical features while expanding the depth of the network.

Pooling Layers: Pooling layers are incorporated after each convolutional stage to reduce the spatial dimensions of feature maps, as illustrated in Figure 2.1. This dimensionality reduction lowers the computational load while retaining the most relevant features. Max pooling extracts the maximum value from each region covered by the filter, effectively highlighting the most prominent features. Average pooling computes the average value, promoting generalization and reducing overfitting. Min pooling, though less commonly used, selects the minimum value, providing additional filtering mechanisms. In the presented architecture, pooling operations reduce the spatial dimensions of the feature maps, leading to a final size of $14 \times 14 \times 16$.

Fully Connected Layers: Following the convolutional and pooling stages, the feature maps are flattened into a single vector and passed through fully connected layers, as indicated in Figure 2.1. These layers are responsible for aggregating the learned features and making classification decisions. The first fully connected layer consists of 256 neurons, each receiving inputs from all preceding feature map values, effectively forming a dense network. The final output layer comprises 10 neurons, each representing a specific class, and applies a softmax activation function to generate class probabilities.

Activation Functions: Activation functions introduce non-linearity to the network, enabling CNNs to learn complex patterns and decision boundaries. As shown in Figure 2.1, the Rectified Linear Unit (ReLU) applies the function $f(x) = \max(0, x)$, effectively retaining positive values while discarding negative ones. PReLU (Parametric ReLU) extends ReLU by incorporating a learnable parameter, providing greater flexibility in activation. The softmax function, primarily used in the output layer, calculates the probability distribution over multiple classes. Sigmoid activation maps inputs to a range between 0 and 1, commonly used in binary classification tasks.

Weight Sharing and Hierarchical Learning: CNNs leverage weight sharing to significantly reduce the number of trainable parameters. Each filter applies the same set of weights across different spatial locations, as illustrated in Figure 2.1, allowing the network to efficiently learn localized patterns. This approach fosters hierarchical learning, where lower layers capture basic features like edges and textures, while deeper layers learn more complex, abstract structures such as shapes and object components. This hierarchical feature extraction is fundamental to CNNs' ability to effectively model complex visual data.

2.2 Overview of Style Transfer

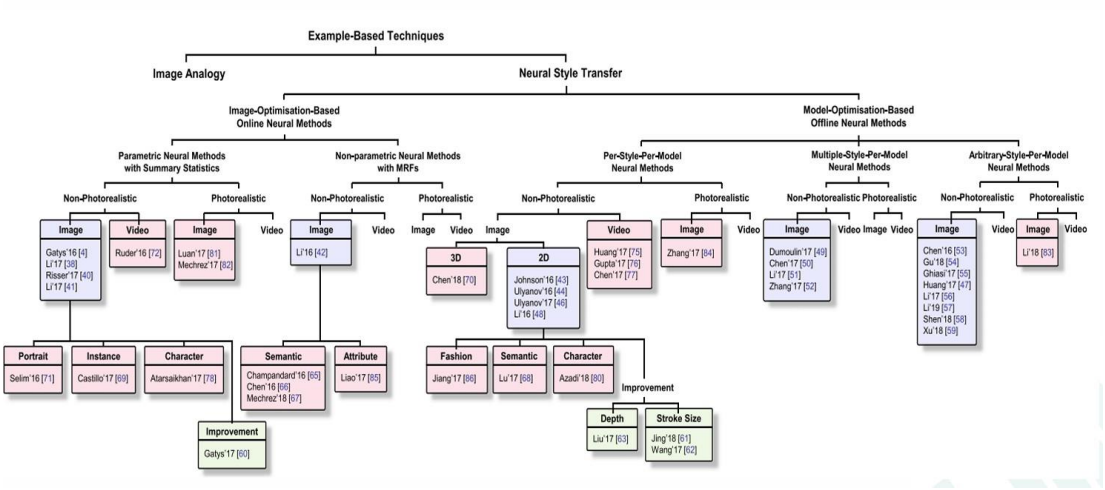


Figure 2.2: Taxonomy of neural style transfer techniques categorized by example-based, image-optimization, and model-optimization methods across different content types and application domains.

The landscape of neural style transfer techniques can be broadly categorized based on the underlying methodology and target application domains, as depicted in Figure 2.2. Style transfer approaches are generally classified into two categories: Example-Based Techniques and Neural Style Transfer (NST). Example-based methods, such as Image Analogy, transfer visual patterns based on predefined examples. In contrast, NST methods are divided into Image-Optimization-Based and Model-Optimization-Based approaches. The former includes parametric methods utilizing summary statistics and non-parametric methods employing Markov Random Fields (MRFs). These techniques are applied to both non-photorealistic and photorealistic content across images, videos, and 3D data. Offline neural methods comprise per-style-per-model, multiple-style-per-model, and arbitrary-style-per-model frameworks, each targeting specific content types and visual effects. Optimization strategies impact computational complexity and visual quality, with per-style-per-model approaches delivering customized outputs at the cost of higher processing time, while arbitrary-style-per-model methods focus on generalization. Additionally, advancements in depth estimation, stroke size control, and semantic preservation have enabled more context-aware stylization. The integration of adaptive normalization and attention mechanisms has further enhanced visual coherence, expanding style transfer applications to include video synthesis and 3D rendering.

2.3 Neural Style Transfer

Neural Style Transfer (NST) was first introduced by Gatys et al. (2016) (1), demonstrating the capability of Convolutional Neural Networks (CNNs) to separate and recombine content and style information from distinct images. The architecture leverages a pre-trained VGG-19 network to extract content and style features from different layers, as illustrated in Figures 2.3 and 2.4 (1).

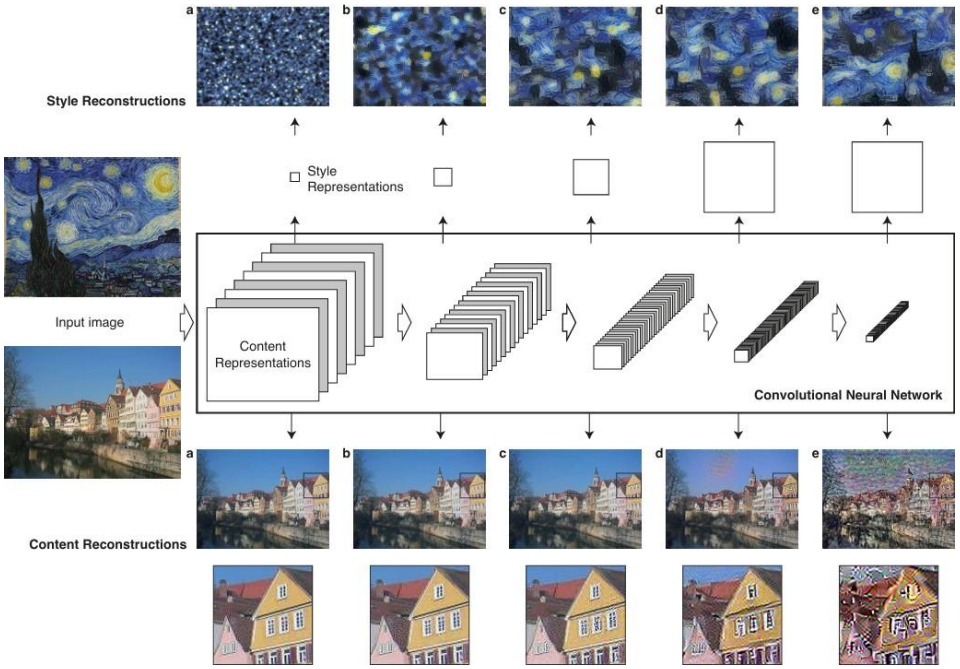


Figure 2.3: Neural Style Transfer framework as proposed by Gatys et al. (2016) (1). The content and style representations are extracted using a pre-trained CNN, with each layer capturing distinct visual features. The lower layers capture finer details, while the higher layers capture more abstract semantic content. Style representations are obtained using Gram matrices to capture texture information across multiple layers.

In the NST framework, the content and style are processed through the VGG-19 network to extract feature maps from specific layers. As shown in Figure 2.3, the content representation is derived from the deeper layers (e.g., `conv4_2`), focusing on the structural and semantic features. In contrast, style representations are computed using Gram matrices across multiple layers, capturing the correlation between feature maps to encode texture information (1).

Optimization Process: The objective of NST is to generate a new image $\mathbf{x}$ that preserves the content of a target image $\mathbf{p}$ while adopting the style of a style image $\mathbf{a}$. This is achieved by minimizing the total loss function (1):

$$L_{total} = \alpha L_{content} + \beta L_{style}$$

where α and β control the trade-off between content preservation and style adaptation. The content loss $L_{content}$ is calculated as the squared difference between the content representation F^l of the target image and the content representation of the generated image at layer l:

$$L_{content} = \sum_l (F^l - P^l)^2$$

The style loss L_{style} is defined using Gram matrices G^l, which encode the correlation between feature maps at layer l:

$$L_{style} = \sum_l w_l (G^l - A^l)^2$$

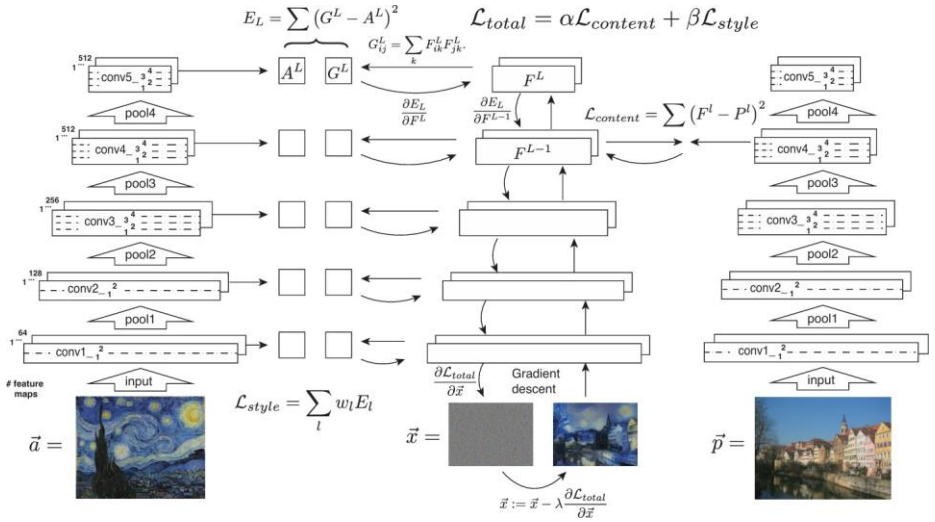


Figure 2.4: Optimization process in Neural Style Transfer (1). The content and style losses are calculated independently across different layers, and the total loss is minimized using gradient descent. The intermediate images demonstrate the progression of style transfer, illustrating the gradual incorporation of style patterns into the content structure.

Figure 2.4 illustrates the optimization process. Initially, a white noise image is generated as the starting point. During each iteration, the generated image is updated by minimizing the total loss using gradient descent. As shown in the intermediate images, the content structure is progressively retained while the style patterns are gradually

incorporated, producing a visually coherent output (1).

Applications and Impact: The NST framework demonstrated the potential of CNNs to effectively blend content and style information, paving the way for artistic style transfer, texture synthesis, and image-to-image translation tasks. The use of Gram matrices to capture style representations proved to be a key contribution, allowing the separation of content and style across multiple layers. Subsequent works have further refined this approach, incorporating attention mechanisms, adaptive normalization techniques, and multi-scale processing to enhance stylization quality and computational efficiency (1).

2.4 Basic Working of Style Transfer

Style transfer is a technique that involves blending the content of one image with the style of another to generate a visually stylized output. The fundamental architecture of style transfer, as illustrated in Figure 2.5, consists of three main components: an encoder, a transformation module, and a decoder.

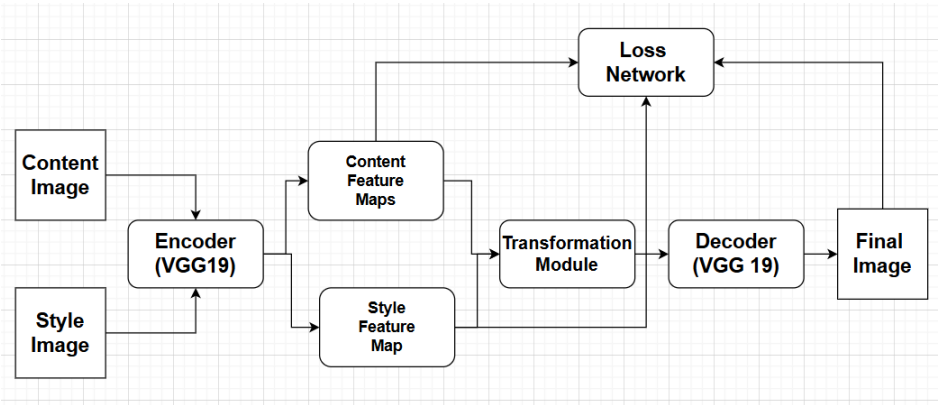


Figure 2.5: Basic architecture of style transfer. The content and style images are processed through the encoder to extract feature maps. These feature maps are then fed to the transformation module, which blends the content and style features. The decoder reconstructs the final stylized image based on the transformed features, and the loss network calculates content and style losses for optimization.

The style transfer pipeline begins with the input of a Content Image and a Style Image, both of which are processed by a pre-trained encoder network, typically a VGG-19 model. The encoder extracts content and style feature maps from intermediate layers. The content feature maps primarily capture structural and spatial information, while the

style feature maps focus on texture and artistic patterns through Gram matrices.

Next, the extracted content and style features are passed to the Transformation Module. This module is responsible for blending the content and style representations. The specific blending method can vary, ranging from simple concatenation to more advanced techniques like Adaptive Instance Normalization (AdaIN) or Attention Mechanisms. The output of this module is a set of transformed feature maps that combine the content structure with the desired style patterns.

Following the transformation stage, the feature maps are passed through the Decoder, which reconstructs the final image by synthesizing a new visual representation that merges content and style attributes. The decoder is trained to generate realistic outputs that maintain structural coherence while adopting the stylistic elements from the style image.

The Loss Network, typically another VGG-19 model, calculates the content and style losses to guide the optimization process. The content loss measures the difference between the content feature maps of the target image and the generated image, while the style loss quantifies the difference between the Gram matrices of the style and generated images. The overall loss is expressed as:

$$L_{total} = \alpha L_{content} + \beta L_{style}$$

where α and β control the relative importance of content and style. The transformation module is iteratively updated through gradient descent to minimize the total loss, ensuring that the generated image effectively preserves the content structure while incorporating the desired stylistic elements.

Thus, the style transfer framework effectively disentangles content and style features, allowing for the synthesis of artistic renditions while maintaining semantic coherence, as depicted in Figure 2.5.

2.5 Components of Style Transfer

The components of style transfer can be broadly categorized into four main sections: architecture, encoder/decoder, transformation module, and loss function, as depicted in Figure 2.6. The architecture forms the foundational structure of style transfer models and includes pipelines such as Art Flow, WCT (Whitening and Coloring Transform), and LST (Linear Style Transfer), each targeting specific stylistic and content preservation objectives. The encoder/decoder module typically leverages pre-trained networks like VGG-based architectures and Art Flow, which extract content and style features and reconstruct the final stylized output. The transformation module, a critical aspect of style transfer, involves methods such as AdaIN (Adaptive Instance Normalization), AdaATTn (Adaptive Attention), MCCNet, SANet, EFM (Exact Feature Matching), and AvatarNet, each incorporating distinct feature fusion techniques to achieve enhanced stylization. Lastly, the loss function is responsible for guiding the optimization process and includes specialized pipelines like MCCNet and IEST (Image Embedding Style Transfer), which refine content-style trade-offs by adjusting style consistency and content fidelity through tailored loss formulations.

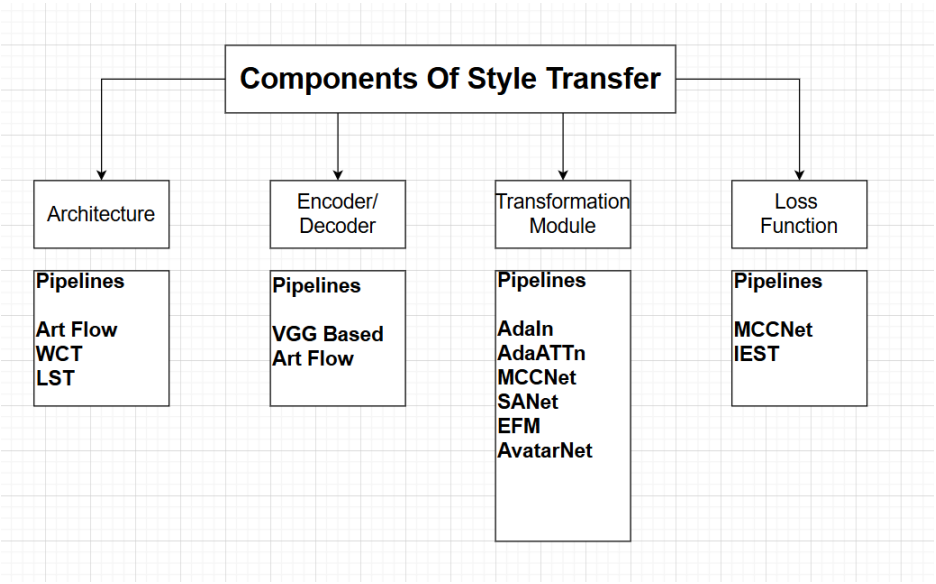


Figure 2.6: Overview of the key components of style transfer, categorized into architecture, encoder/decoder, transformation module, and loss function, along with their respective pipelines.

CHAPTER 3

Encoders, Decoders And Transformation Modules

3.1 Encoders & Decoders

Encoders and decoders form the fundamental backbone of style transfer architectures, serving as key components for extracting and reconstructing image features. The encoder is responsible for transforming the input images into a latent feature space, capturing content and style information at multiple hierarchical levels. It typically utilizes pre-trained convolutional neural networks to effectively extract spatial and semantic patterns. The decoder, in contrast, reconstructs the stylized image from the fused content and style features, mapping them back to the original image space. While the encoder is often fixed and pre-trained, the decoder is trained specifically to generate stylized outputs that maintain content structure while incorporating stylistic elements. Commonly used encoder-decoder frameworks in style transfer include VGG19-based networks and GLOW-based architectures, each offering distinct approaches for content and style fusion.

3.1.1 VGG19-Based Encoder and Decoder in Style Transfer

The VGG19 network, as depicted in Figure 3.1, is a deep convolutional neural network initially trained on the IMAGENET dataset for object classification. In the context of style transfer, VGG19 serves as the **encoder** to extract feature maps representing both content and style from input images (1). The architecture comprises multiple convolutional layers followed by max-pooling layers, allowing it to hierarchically capture spatial and semantic features across different layers.

The **encoder** leverages specific layers of VGG19 to isolate content and style information. For content representation, the deeper layers such as `conv4_2` are employed, capturing higher-level structural and semantic details. For style representation, lower and intermediate layers such as `conv1_1`, `conv2_1`, `conv3_1`, `conv4_1`,

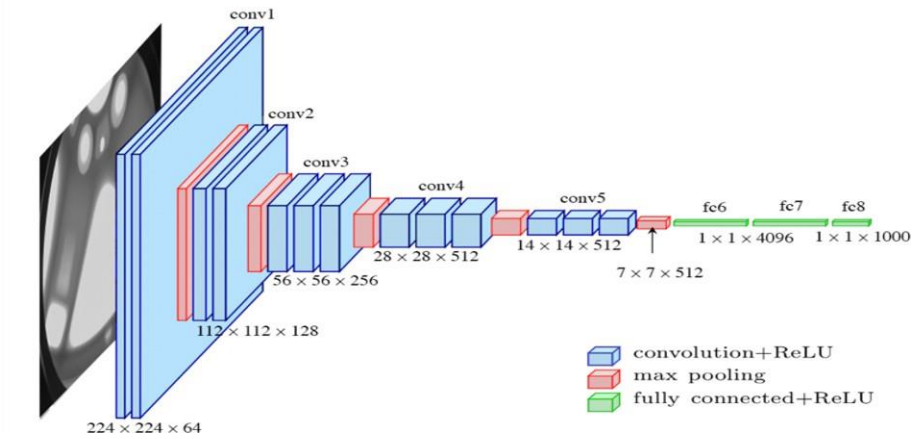


Figure 3.1: VGG19 architecture illustrating the convolutional and fully connected layers. The network is pre-trained on the IMAGENET dataset and is commonly utilized for extracting content and style features in style transfer (1).

and `conv5_1` are utilized. These layers are used to compute Gram matrices, effectively encoding texture and spatial patterns across multiple levels of abstraction (1).

The **decoder**, in contrast, is a network that mirrors the encoder architecture but operates in reverse. It receives the fused content and style features as input and reconstructs the stylized image. Unlike the encoder, the decoder is trained to synthesize images that retain the content structure while integrating the style patterns extracted from the style image. The decoder is not pre-trained but is designed to symmetrically reverse the encoding process, effectively mapping feature representations back to the pixel space (1).

The encoder is kept fixed during the style transfer process, serving as a feature extractor, while the decoder is trained to map these features into the final stylized output. This design choice ensures that the content and style features are effectively disentangled and reassembled, maintaining structural integrity while incorporating stylistic attributes (1).

3.2 Glow-Based Encoder and Decoder in Style Transfer

The Glow network, as illustrated in Figure 3.2, is a flow-based generative model designed to perform lossless and reversible transformations. Unlike traditional autoencoder-based methods like VGG19, Glow employs a projection network that includes **Activation Normalization (ActNorm)**, **Invertible 1x1 Convolutions**, and **Additive Cou-**

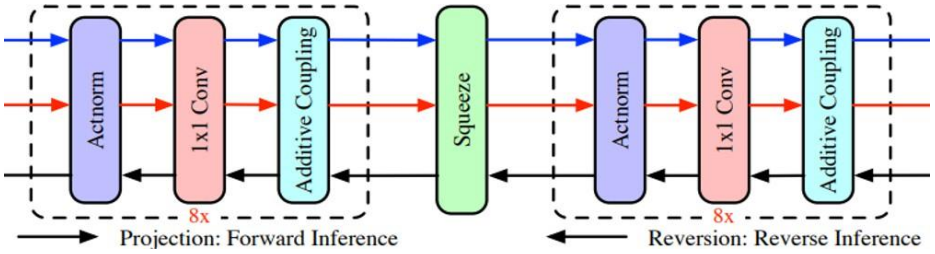


Figure 3.2: Glow-based architecture consisting of activation normalization, invertible 1x1 convolution, and additive coupling layers. The network is capable of reversible transformations, ensuring lossless content and style representation (8).

pling Layers (8). These components facilitate precise content and style feature extraction without introducing reconstruction loss. By maintaining a bijective mapping between the input and latent space, Glow ensures that every transformation is invertible, allowing for precise recovery of the input content during the decoding process.

The **encoder** in the Glow framework projects the content and style images into a latent feature space using the flow-based structure. The projection network leverages invertible convolutions and additive coupling layers, ensuring that the extracted features retain all spatial information without degradation. This mechanism prevents content leakage, a common issue observed in traditional encoder-decoder frameworks. Unlike VGG19, where the encoder is fixed and pre-trained, Glow actively learns the encoding function through its flow-based architecture, allowing for adaptive feature extraction that is both lossless and reversible (8).

The **decoder**, in contrast, performs reverse inference by mapping the fused content and style features back to the pixel space, reconstructing the stylized image. The key advantage of the Glow decoder lies in its ability to perfectly invert the encoded features, maintaining exact content and stylistic attributes without introducing artifacts. This reversibility is achieved through the coupling layers and invertible convolutions, which apply learned transformations in both forward (encoding) and reverse (decoding) passes. Additionally, the decoder does not require explicit training, as the transformations are inherently reversible, effectively preventing accumulated reconstruction errors and ensuring structural integrity throughout the style transfer process (8).

Moreover, the Glow framework offers distinct advantages over conventional autoencoders, such as precise image reconstruction and the ability to prevent content leakage.

By operating as a bijective network, Glow maintains a one-to-one mapping between the input and output, allowing it to capture intricate stylistic patterns without compromising spatial coherence. Additionally, Glow-based decoders can generate higher quality stylized outputs by preserving the exact feature representation in both forward and reverse passes, making them well-suited for high-resolution style transfer applications (8).

3.3 Transformation Modules

Transformation modules are integral components in style transfer pipelines, responsible for fusing content and style features to generate the stylized output. Unlike encoders and decoders, which primarily focus on feature extraction and reconstruction, transformation modules perform content-style blending by applying specific transformation techniques. Common transformation modules include **Adaptive Instance Normalization (AdaIN)**, **Adaptive Attention (AdaATTn)**, **MCCNet (Multi-Channel Correlation Network)**, **SANet (Self-Attention Network)**, **EFM (Exact Feature Matching)**, and **AvatarNet**. Each of these modules introduces distinct mechanisms for feature alignment and fusion.

AdaIN achieves content-style fusion by aligning the mean and variance of content features to match the style features, effectively capturing stylistic patterns while preserving structural content. AdaATTn extends this concept by incorporating attention mechanisms, allowing the network to selectively focus on critical style regions, thereby enhancing visual coherence in the stylized output. MCCNet, on the other hand, employs multi-channel correlation to establish fine-grained content-style correspondence, maintaining spatial integrity during the transformation process.

EFM utilizes feature matching techniques to directly align content and style representations in the latent space, ensuring precise style adaptation without compromising content structure. Similarly, SANet leverages self-attention to capture contextual dependencies between content and style features, resulting in globally consistent stylization. AvatarNet introduces spatially-aware transformations that enable the generation of stylized outputs with enhanced texture and structure preservation.

Transformation modules in style transfer blend content and style features. In the below sections we will be discussing about different transformation modules.

3.3.1 Adaptive Instance Normalization (AdaIN)

Adaptive Instance Normalization (AdaIN), introduced by Huang and Belongie (2), is a transformation module that enables effective style transfer by aligning the statistical properties of content and style features. Unlike conventional normalization methods that apply fixed affine transformations, AdaIN adaptively adjusts the mean and variance of content features to match those of the style features, effectively transferring stylistic patterns while maintaining structural content.

The transformation process in AdaIN is mathematically expressed as:

$$\text{AdaIN}(x, y) = \sigma(y) \frac{x - \mu(x)}{\sigma(x)} + \mu(y) \tag{3.1}$$

where x represents the content features and y represents the style features. The channel-wise mean $\mu(\cdot)$ and standard deviation $\sigma(\cdot)$ of the content are adjusted to align with those of the style features. This process ensures that the content structure is preserved while stylistic textures are seamlessly integrated.

AdaIN facilitates **arbitrary style transfer**, allowing for real-time adaptation to any style without the need for additional training. This capability is achieved by leveraging a pre-trained VGG19 encoder to extract content and style features, followed by a decoder that reconstructs the stylized output based on the transformed features. Unlike traditional style transfer methods that rely on fixed-style models, AdaIN offers dynamic style adaptation, making it computationally efficient and highly flexible.

A key advantage of AdaIN is its user-controlled style interpolation mechanism, allowing for fine-grained adjustments in the degree of stylization. By controlling the style-content trade-off parameter, users can influence the extent to which the style is applied, enabling customized stylized outputs. Furthermore, AdaIN is particularly effective in minimizing content leakage, ensuring that the content structure remains intact while the stylistic attributes are seamlessly incorporated.

The introduction of AdaIN has significantly advanced the field of style transfer by simplifying the transformation process while maintaining high visual quality. Its ability to achieve real-time style transfer, combined with content-style trade-off control, makes it a versatile and widely adopted transformation module in style transfer frameworks.

3.3.2 AvatarNet

AvatarNet, proposed by Sheng et al. (6), introduces a zero-shot style transfer model that effectively decorates content features with stylistic patterns from arbitrary style images. Unlike other methods that rely on pre-trained style models or extensive training, AvatarNet employs a patch-based feature manipulation module known as the *style decorator*. This module robustly aligns content features with semantically similar style patches, ensuring that the stylized output maintains both structure and texture consistency.

The style transfer process in AvatarNet is divided into three key stages. The first stage, known as the **Projection Step**, projects content and style features into a common feature space by whitening and scaling operations. This normalization step ensures that the content and style features are statistically aligned, reducing the impact of domain discrepancies. The transformation process can be mathematically represented as:

$$\bar{z}_c = W_c \otimes (z_c - \mu(z_c)), \qquad \bar{z}_s = W_s \otimes (z_s - \mu(z_s)) \tag{3.2}$$

In the second stage, **Matching and Reassembly**, the content features are matched with the most semantically relevant style patches. This alignment is achieved by measuring the similarity between content and style patches using normalized cross-correlation. The reassembled features $\bar{z}_{cs}$ effectively incorporate detailed style patterns while preserving spatial content distribution:

$$\bar{z}_{cs} = \Phi(\bar{z}_s)^\top \otimes B(\Phi(\bar{z}_s) \otimes \bar{z}_c) \tag{3.3}$$

The final stage, **Reconstruction**, applies a coloring transformation to the reassembled features to ensure that the stylized output matches the overall second-order statistics of the style image. The transformed features are expressed as:

$$z_{cs} = C_s \otimes \bar{z}_{cs} + \mu(z_s) \tag{3.4}$$

AvatarNet provides several advantages over conventional methods such as AdaIN and WCT. It enables multi-scale style adaptation in a single feed-forward pass, reducing computational cost and enhancing processing efficiency. Additionally, it preserves

fine-grained style details through patch-based reassembly, making it particularly effective for maintaining intricate patterns and textures. Moreover, the model's hourglass network architecture ensures that both local and global style patterns are integrated without distorting content structure.

Overall, AvatarNet achieves a balance between computational efficiency and stylization quality, making it a robust framework for zero-shot style transfer across diverse visual content (6).

3.3.3 SANet

Style-Attentional Network (SANet) is a self-attention-based transformation module proposed by Park and Lee (7), designed to address the challenges of capturing both global and local style patterns in arbitrary style transfer tasks. Unlike patch-based methods such as AvatarNet, which rely on hard attention, SANet employs a learnable soft-attention mechanism that allows for more flexible style-content matching.

SANet utilizes two SANet modules, each corresponding to different VGG layers (ReLU 4_1 and ReLU 5_1), to effectively capture both fine-grained textures and broader stylistic patterns. The feature embedding process in SANet can be expressed as:

$$F_{cs}^{i} = \frac{1}{C(F)} \sum_{j} \exp(f(F_c^i)^T g(F_s^j)) h(F_s^j) \tag{3.5}$$

Here, F^i_{cs} represents the transformed content-style feature map, while $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are learned transformation functions that map the content and style features to a common embedding space. By integrating soft attention weights, SANet adaptively matches content features to semantically similar style patterns, achieving more nuanced and context-aware stylization.

A key advantage of SANet is its ability to handle both global and local style patterns simultaneously, a limitation faced by other methods like AdaIN and WCT. Additionally, SANet leverages an identity loss function that preserves the structure of the content image while ensuring that the style is accurately transferred. This identity loss is defined as:

$$L_{identity} = \lambda_{identity1} \| I_{cc} - I_c \|^2 + \lambda_{identity2} \sum_i \| \phi_i(I_{cc}) - \phi_i(I_c) \|^2 \quad (3.6)$$

The inclusion of the identity loss function prevents structural distortion in the content image, allowing for better content preservation while transferring style patterns effectively.

SANet achieves real-time processing speeds, outperforming other state-of-the-art methods in terms of both quality and computational efficiency. It processes images at approximately 18-24 frames per second at 512px resolution, making it highly suitable for real-time applications. The use of soft-attention mechanisms also ensures that SANet maintains semantic consistency across different style regions, providing a more coherent stylization output without compromising content structure (7).

3.3.4 AdaATTn

The Adaptive Attention Normalization (AdaAttN) module addresses the challenge of achieving fine-grained style transfer by leveraging both shallow and deep features from content and style images (10). Unlike traditional methods that focus on global feature statistics, AdaAttN normalizes content features using per-point weighted statistics derived from the attention-weighted output of all style feature points. This approach allows for local feature adaptation, significantly enhancing the stylization quality.

The core of AdaAttN lies in computing spatial attention scores from both shallow and deep features of content and style images. These scores are used to calculate weighted statistics where a style feature point represents a distribution of the attention-weighted output. By aligning content features to these per-point weighted style statistics, AdaAttN ensures that local feature details are preserved, resulting in more visually coherent and locally adaptive style transfer outputs.

Furthermore, AdaAttN incorporates a novel local feature loss to enhance the visual quality of the output by minimizing discrepancies between the transformed content features and the desired style feature distribution. The module's ability to maintain detailed local patterns while ensuring global style consistency makes it highly effective for both image and video style transfer. It also supports video style transfer with improved tem-

poral consistency by introducing cosine similarity-based attention and cross-image similarity loss. This capability makes AdaAttN versatile and robust in diverse stylization scenarios (10).

3.3.5 Exact Feature Distribution Matching (EFDM)

Exact Feature Distribution Matching (EFDM) introduces a precise methodology for aligning feature distributions in arbitrary style transfer and domain generalization tasks (11). Unlike conventional methods that rely on Gaussian assumptions, EFDM leverages empirical Cumulative Distribution Functions (eCDFs) to achieve exact matching of feature distributions. This approach ensures that the style transfer process accurately aligns not only the mean and standard deviation but also higher-order statistics, leading to more faithful style transfer results. By employing the Exact Histogram Matching (EHM) algorithm, EFDM conducts feature distribution alignment through a sorting-based mechanism, effectively minimizing discrepancies between input and target feature distributions. This mechanism preserves both local and global feature structures, mitigating potential artifacts and distortions.

EFDM also integrates a computationally efficient algorithm known as Sort-Matching, which implements histogram matching in a plug-and-play manner without introducing additional model parameters. This enables EFDM to operate at a higher frame rate, achieving up to 256 FPS for images of size 512×512, making it applicable for real-time style transfer applications. The robustness of EFDM in handling complex feature distributions beyond Gaussian is demonstrated in its ability to maintain structural integrity while preserving detailed stylization, effectively preventing content leak and ensuring semantic consistency across generated outputs.

Additionally, EFDM's plug-and-play nature allows it to be seamlessly integrated into existing style transfer architectures, extending its applicability to diverse tasks without requiring extensive model reconfiguration. This flexibility, coupled with its computational efficiency, makes EFDM a practical solution for real-time applications, where maintaining visual quality and stylistic accuracy are critical. The combination of exact feature matching, histogram alignment, and computational speed establishes EFDM as a robust framework for accurate and efficient style adaptation (11).

CHAPTER 4

Loss Functions

4.1 What is Loss Function?

In style transfer, a loss function serves as a guiding objective that evaluates how well the generated output preserves content structure while integrating stylistic patterns from a reference image. It ensures that the output maintains visual coherence by minimizing discrepancies between the generated image and target content and style features. Content loss focuses on preserving spatial and structural integrity, preventing distortion of the original content. Style loss captures stylistic elements such as texture, color, and artistic patterns, aligning the generated output with the desired style. Contrastive loss enhances the distinction between content and style features, ensuring that stylistic patterns remain distinct without compromising content fidelity. Identity loss further reinforces structural consistency by preventing excessive transformation, maintaining content integrity throughout the stylization process. Together, these loss functions provide a comprehensive framework for balancing content preservation and style adaptation, ensuring visually coherent and semantically consistent style transfer outputs.

4.2 Content Loss in Style Transfer

Content loss is a critical objective function in style transfer that measures the structural similarity between the content image and the stylized image. It ensures that the essential spatial structure and semantic information of the content image are retained in the stylized output while allowing for the incorporation of stylistic patterns from the style image. Content loss is typically calculated using feature maps extracted from a pre-trained convolutional neural network, such as VGG-19, at specific layers that capture high-level content information (1).

Mathematically, the content loss is defined as the Mean Squared Error (MSE) between the feature maps of the content image and the stylized image. It is expressed

as:

$$L_{content} = \sum_{i} \| F_c^i - F_s^i \|^2 \tag{4.1}$$

where F_c^i and F_s^i are the feature maps extracted from the i^{th} layer of the pre-trained encoder for the content and stylized images, respectively. The content loss ensures that the spatial layout and semantic structure of the content image are preserved in the generated output (1).

Additionally, content loss can also be computed using statistical measures such as mean and variance to capture both first-order and second-order statistics of the feature maps. The statistical content loss is formulated as:

$$L_{mean} = \| \mu(F_c) - \mu(F_s) \|^2 \tag{4.2}$$

$$L_{variance} = \| \sigma(F_c) - \sigma(F_s) \|^2 \tag{4.3}$$

$$L_{stat} = L_{mean} + L_{variance} \tag{4.4}$$

where $\mu(\cdot)$ and $\sigma(\cdot)$ represent the mean and variance of the feature maps. This statistical content loss provides a more comprehensive measure by capturing both pixel-level differences and overall structural alignment (1).

Another important variation of content loss is the **Total Variation Loss (TV Loss)**, which focuses on maintaining smoothness and reducing artifacts in the stylized output. It is defined as:

$$L_{TV} = \sum_{i,j} (I_{i+1,j} - I_{i,j})^2 + (I_{i,j+1} - I_{i,j})^2 \tag{4.5}$$

TV Loss penalizes abrupt intensity changes between adjacent pixels, promoting smooth transitions and minimizing visual artifacts in the generated output (1).

Overall, content loss serves as a vital component in style transfer frameworks, guid-

ing the model to preserve the spatial structure and content information while allowing for artistic stylization. By employing feature map comparisons, statistical alignment, and smoothness constraints, content loss effectively balances content preservation and stylistic adaptation, resulting in visually coherent and semantically consistent stylized images.

4.3 Style Loss in Style Transfer

Style loss is a critical component in style transfer that quantifies the visual similarity between the style image and the generated image. It ensures that the generated output effectively captures the stylistic patterns, textures, and color distributions of the reference style image while maintaining the content structure. Style loss is primarily calculated by comparing the feature maps of the style and generated images, typically extracted using a pre-trained encoder network such as VGG-19 (1).

A widely used method for computing style loss is the **Gram Matrix-Based Style Loss**. The Gram matrix, $G(F)$, represents the correlation between different feature maps and is defined as:

$$G(F)_{ij} = \sum_{k} F_{ik} F_{jk} \tag{4.6}$$

The style loss is then computed as the Mean Squared Error (MSE) between the Gram matrices of the style image and the generated image:

$$L_{style} = \sum_{i} \| G(F_s^i) - G(F_g^i) \|^2 \tag{4.7}$$

This formulation captures the spatial dependencies between feature maps, effectively encoding stylistic patterns such as textures, brushstrokes, and artistic styles. By aligning the Gram matrices, the generated image is encouraged to adopt the style patterns of the reference image while preserving its own content structure (1).

Another approach for calculating style loss involves **Mean and Variance Matching**. Instead of relying solely on the Gram matrix, this method aligns the statistical

properties of the style and generated images by matching their mean and variance values:

$$L_{style} = \|\mu(F_s) - \mu(F_g)\|^2 + \|\sigma(F_s) - \sigma(F_g)\|^2 \tag{4.8}$$

Here, $\mu(\cdot)$ and $\sigma(\cdot)$ represent the mean and standard deviation of the feature maps, respectively. This method effectively captures global style patterns, making it computationally efficient for real-time style transfer (1).

Overall, style loss serves as a guiding objective in style transfer, enabling the model to learn and apply stylistic patterns while maintaining visual coherence. By incorporating both Gram matrix-based correlations and statistical alignment, style loss ensures that the generated output retains the stylistic essence of the reference image while adhering to its structural content (1).

4.4 Identity Loss in Style Transfer

Identity loss is a crucial objective function in style transfer that ensures the consistency of the content and style features when the content image and style image are identical. Its primary purpose is to prevent unnecessary transformations when the input content and style images are the same, thereby maintaining content structure and stylistic characteristics without introducing artifacts (9).

Mathematically, identity loss is designed to minimize the discrepancy between the input and output images when no transformation is required. It comprises two main components: content preservation and style preservation. The content preservation term penalizes deviations between the original content image (I_c) and the output image (I_{cc}), while the style preservation term ensures that the style features remain unchanged when the style image is passed through the network without any transformation. The identity loss function can be expressed as:

$$L_{identity} = \lambda_1 \left\| I_{cc} - I_c \right\|^2 + \lambda_2 \left\| I_{ss} - I_s \right\|^2 + \sum_i \lambda_3 \left\| \phi_i(I_{cc}) - \phi_i(I_c) \right\|^2 + \lambda_4 \left\| \phi_i(I_{ss}) - \phi_i(I_s) \right\|^2 \tag{4.9}$$

where I_{cc} and I_{ss} represent the generated content and style images, I_c and I_s are the original content and style images, $\phi_i(\cdot)$ denotes the feature map extracted at the i^{th} layer of the encoder, and λ_1, λ_2, λ_3, λ_4 are hyperparameters controlling the balance between content and style preservation.

Identity loss effectively helps the network learn a consistent transformation function that ensures that when the input content and style images are identical, the generated output remains unchanged. This mechanism is particularly important in preventing unnecessary stylistic distortions and maintaining content integrity during the stylization process. Additionally, identity loss serves as a regularization technique, guiding the model to retain essential content features while allowing for controlled style adaptation (9).

4.5 Contrastive Loss in Style Transfer

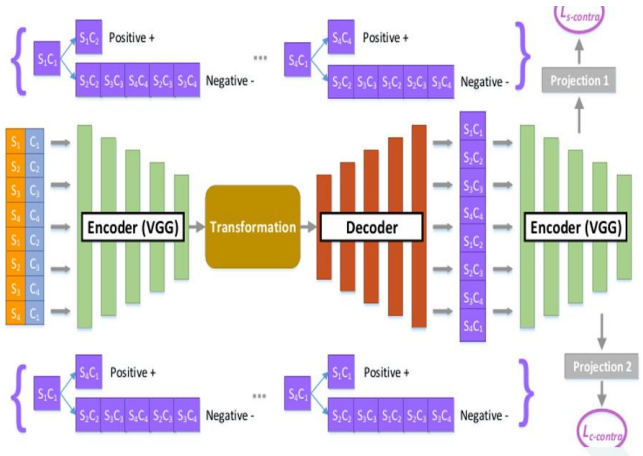


Figure 4.1: Architecture for implementing contrastive loss in style transfer. Positive and negative pairs are identified based on content and style similarities, with projections to a lower-dimensional space for contrastive alignment.

Contrastive loss is a pivotal objective in style transfer that focuses on enhancing style consistency and preventing content distortion across multiple stylizations. It achieves this by enforcing similarity between images with the same content or style while main-

taining distinctiveness between dissimilar pairs. The objective is to group style-consistent images together and separate those with different stylistic characteristics, thus maintaining a clear boundary between content and style features (9).

In style transfer frameworks, contrastive loss is implemented through two primary components: **Style Contrastive Loss** and **Content Contrastive Loss**. Style contrastive loss ensures that images sharing similar style features are mapped closer in the feature space, effectively maintaining stylistic coherence across different images. Conversely, content contrastive loss prevents content distortion by aligning content features across stylizations, ensuring that the structural integrity of the content is preserved despite varying style applications.

The architecture for implementing contrastive loss involves encoding the content and style images through a VGG-based encoder, followed by a transformation module and a decoder. The encoded feature representations are then projected to a lower-dimensional space, where positive pairs (images with similar content/style) are brought closer, and negative pairs (images with differing content/style) are pushed apart. By leveraging contrastive loss, the model learns to maintain style consistency and content integrity across multiple stylizations, effectively preventing undesirable distortions while enhancing the stylistic fidelity of the generated outputs (9).

4.6 Loss Function Combination in Style Transfer

The overall loss function in style transfer is a linear combination of multiple loss components, each weighted by a specific coefficient to control its relative influence. By adjusting these weights, the model can determine how much emphasis to place on content preservation, style adaptation, or structural consistency. The general form of the loss function can be expressed as:

$$L_{total} = \alpha L_{content} + \beta L_{style} + \gamma L_{contrastive} + \delta L_{identity}$$

where $\alpha, \beta, \gamma, \delta$ are weight coefficients that define the contribution of each loss term. By varying these coefficients, the model can prioritize content retention, enhance style fidelity, minimize content distortion, or maintain structural consistency.

CHAPTER 5

Architectures In Style Transfer

5.1 Architectures

In style transfer, various architectures are employed to effectively blend content and style while maintaining visual coherence. Two prominent architectures commonly used in this domain are **ArtFlow** (8) and **WCT (Whitening and Coloring Transform)** (4). ArtFlow leverages reversible neural networks to achieve style transfer without content distortion, ensuring high-quality outputs by preventing content leak and preserving spatial structures. On the other hand, WCT utilizes a feature transformation mechanism to align content and style distributions through whitening and coloring operations, effectively capturing stylistic patterns while maintaining content integrity. Both architectures offer distinct approaches to handling content and style interactions, providing flexibility in style transfer applications.

5.2 Whitening and Coloring Transform (WCT) Architecture

The Whitening and Coloring Transform (WCT) architecture is a notable style transfer pipeline that employs feature transformation to effectively align content and style distributions. The WCT framework leverages the VGG-based encoder-decoder structure along with whitening and coloring operations to achieve style transfer without content distortion. The overall pipeline, as illustrated in Figure 5.1, can be divided into three main stages: Reconstruction, Single-Level Stylization, and Multi-Level Stylization.

Reconstruction Stage: The first stage in the WCT pipeline is the reconstruction phase, where the input content image is passed through a VGG-based encoder network to extract feature maps at different layers. These feature maps are then decoded using corresponding reconstruction decoders, allowing the network to reconstruct the content

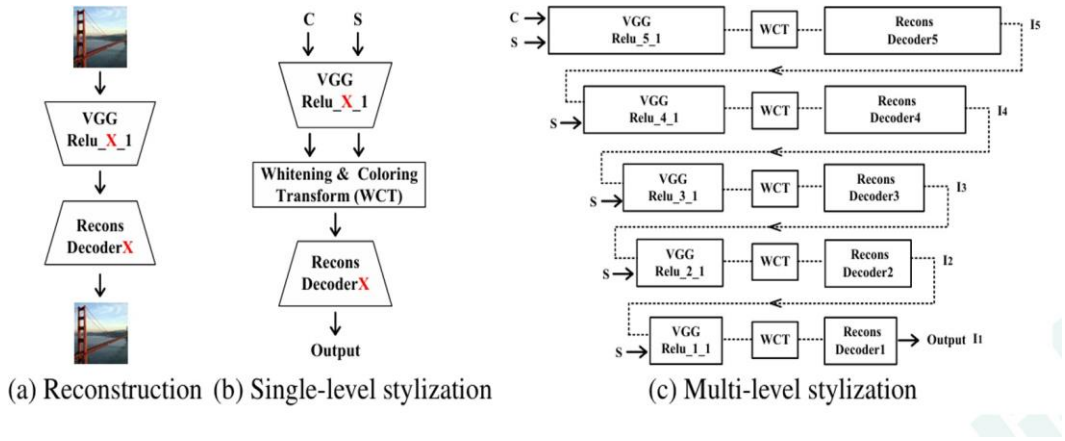


Figure 5.1: WCT Architecture: (a) Reconstruction, (b) Single-level stylization, and (c) Multi-level stylization. Each stage employs the Whitening and Coloring Transform to align content and style features effectively.

image. This stage establishes a baseline for content representation in the style transfer framework.

Single-Level Stylization: In the single-level stylization stage, the WCT operation is applied at a specific layer (e.g., ReLU_X_1) to align content and style feature distributions. The WCT operation performs the following two steps: - **Whitening:** The content feature map is whitened to remove its original statistics, effectively neutralizing content-specific information. - **Coloring:** The whitened feature map is then re-colored using the statistical attributes of the style image, effectively transferring style features while maintaining content structure.

The stylized output is then passed through the decoder to generate the final output at that particular layer. This single-level approach provides localized style adaptation based on specific feature maps.

Multi-Level Stylization: To achieve a more comprehensive stylization effect, the WCT architecture employs multi-level stylization. In this stage, the WCT operation is applied at multiple layers of the VGG network (e.g., ReLU_1_1, ReLU_2_1, ReLU_3_1, ReLU_4_1, ReLU_5_1). This hierarchical approach ensures that both shallow and deep feature maps are stylized, capturing fine-grained textures as well as high-level structural patterns.

After each WCT operation, the intermediate stylized output is passed through a corresponding decoder to reconstruct the image at that specific level. The outputs from

each level are then progressively merged to produce the final stylized image, effectively integrating multi-scale style features throughout the network.

WCT architecture effectively addresses content-style alignment by employing whitening and coloring transformations, allowing for effective style adaptation while preserving content integrity (4).

5.3 ArtFlow Pipeline

ArtFlow is a style transfer framework specifically designed to address the content leak problem that is prevalent in conventional encoder-decoder architectures. Content leak refers to the phenomenon where the content structure of the input image is distorted or lost during the style transfer process. This occurs due to excessive transformation of content features in an attempt to align them with the style features, leading to unintended blending and loss of essential content information. ArtFlow mitigates this issue by employing a reversible flow-based architecture that maintains content integrity throughout the transfer process. The overall pipeline is illustrated in Figure 5.2.

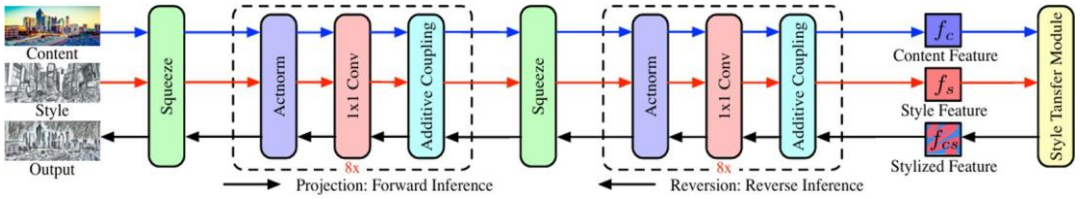


Figure 5.2: ArtFlow Pipeline: The architecture consists of three main stages - Projection, Transfer, and Reversion. The Projection Flow Network (PFN) extracts content and style features using reversible transformations, while the Transfer module merges the features to produce stylized output. The Reversion stage reconstructs the stylized image using reverse inference through the PFN.

The ArtFlow pipeline is structured into three primary stages as illustrated in Figure 5.2:

5.3.1 Projection

In the projection stage, the content and style images are passed through the **Projection Flow Network (PFN)**, which comprises a series of reversible transformations. These

transformations include: - **Squeeze Layer:** Reduces spatial dimensions to focus on relevant feature regions. - **Actnorm Layer:** Stabilizes feature distributions by applying activation normalization. - **Invertible 1x1 Convolution:** Ensures that the transformation remains reversible, preserving spatial structure. - **Additive Coupling Layer:** Splits the input into two parts, allowing localized feature transformation without altering global structure.

The PFN extracts content features (f_c) and style features (f_s) without loss of information due to its reversible nature. This reversibility is crucial in preventing content leak, as the original content features can be fully recovered during the reversion stage.

5.3.2 Transfer

The transfer stage applies the **Unbiased Style Transfer Module**, which combines the content and style features while minimizing content distortion. Unlike conventional methods that overemphasize stylistic adaptation, ArtFlow employs an unbiased transfer mechanism that ensures the stylistic patterns are transferred without overpowering the content structure. This module operates in the latent space, adjusting the stylistic patterns while maintaining the integrity of the content features.

5.3.3 Reversion

In the reversion stage, the combined features (f_{cs}) are passed through the PFN in reverse order to reconstruct the stylized image. Since the PFN is fully reversible, the reversion process effectively mitigates content leak by preserving the original content structure while incorporating stylistic elements. The network ensures that no content information is lost during transformation, achieving a balanced stylization effect.

5.3.4 Addressing Content Leak in ArtFlow

Content leak is a common issue in style transfer where the structural information of the content image is compromised during the transfer process. Traditional encoder-decoder architectures often distort content features by overly emphasizing style patterns, leading to unintended blending and visual artifacts. ArtFlow addresses this problem by

employing a flow-based reversible network that preserves the original content features throughout the projection and reversion stages. By maintaining a bijective mapping between the input and output feature spaces, ArtFlow prevents content distortion and ensures structural integrity, effectively preventing content leak and maintaining a balanced content-style transformation.

Overall, ArtFlow leverages reversible flows and unbiased transfer modules to effectively address the content leak problem, producing high-quality stylized outputs without compromising content integrity. The proposed pipeline demonstrates robust stylization performance while maintaining structural consistency throughout the projection-transfer-reversion process (8).

5.4 Transformation Modules as Style Transfer Architectures

Transformation modules in style transfer, such as **AdaIN** (2), **AdaATTn** (10), **SANet** (7), **MCCNet** (9), **AvatarNet** (6), and **EFM** (11), implement distinct strategies for content-style fusion while maintaining structural integrity. AdaIN aligns the mean and variance of content features to match those of the style, achieving seamless style adaptation without altering content structure. AdaATTn employs adaptive attention to selectively blend style patterns with content features, enhancing stylistic coherence. SANet leverages self-attention to capture spatial dependencies, allowing for context-aware style transfer across different regions. MCCNet maintains structural integrity through multi-channel correlation, ensuring fine-grained style integration. AvatarNet utilizes patch-based matching to preserve spatial consistency, preventing content misalignment, while EFM employs histogram matching to align content and style distributions, effectively preventing content leakage. Collectively, these modules not only perform content-style blending but also address key challenges such as content leak, structural distortion, and stylistic imbalance, making them integral components in style transfer pipelines.

CHAPTER 6

Datasets , Experiments , Results And Metrics

In this chapter, we will explore the datasets utilized for training and evaluating style transfer models, detailing the content and style image collections employed to achieve diverse stylistic effects. The experimental setup will outline the implementation frameworks, model configurations, and parameter settings used for evaluating various style transfer architectures, including AdaIN, SANet, and ArtFlow. We will also present the quantitative and qualitative results obtained across different models, comparing the effectiveness of each transformation module in maintaining content structure while achieving stylistic consistency. Additionally, we will discuss the evaluation metrics used to assess style transfer performance, such as structural similarity, perceptual loss, providing a comprehensive analysis of the effectiveness and limitations of each style transfer technique.

6.0.1 Datasets for Style Transfer

For the style transfer tasks, two primary datasets were utilized to provide a diverse range of content and style images: **MS COCO 2014** (14) and **Wiki Art** (15). These datasets were selected based on their extensive variability in content scenes and artistic styles, enabling comprehensive evaluation of style transfer techniques.

MS COCO 2014 (Microsoft Common Objects in Context) is a large-scale dataset widely used for image recognition, segmentation, and style transfer tasks. It comprises a total of 328,000 images, with 82,783 images designated for training and 40,504 images for validation. The dataset features a broad array of real-world scenes, encompassing objects, human activities, and complex backgrounds, serving as the primary content dataset for style transfer experiments (14).

Wiki Art serves as the primary style dataset, containing over 80,000 artworks spanning 25 distinct artistic styles. This dataset includes paintings from various art movements such as Impressionism, Cubism, Abstract, and more, offering a diverse spectrum

of artistic patterns and textures. By leveraging these artistic styles, the Wiki Art dataset enables the generation of visually compelling and stylistically consistent outputs (15).

Incorporating these datasets ensures a comprehensive evaluation of style transfer models across diverse content and style domains, facilitating robust analysis of transformation modules and their ability to maintain content structure while achieving artistic stylization.

Figure 6.1: Sample content image 1 from MS COCO dataset.

Figure 6.2: Sample content image 2 from MS COCO dataset.

Figure 6.3: Sample style image 1 from Wiki Art dataset.

Figure 6.4: Sample style image 2 from Wiki Art dataset.

Figure 6.5: Sample images from the datasets. Top row: MS COCO content images. Bottom row: Wiki Art style images.

6.1 Experiments

This section explores the experimental setups and modifications applied to the style transfer pipeline to address key challenges such as content leakage, reconstruction loss accumulation, and stylistic consistency. The experimentation is divided into three major areas: ArtFlow with attention-based modules, enhancing style transfer with identity and contrastive loss, and variations in normalization methods within the ADAIN framework.

6.1.1 ArtFlow with Attention-Based Modules

ArtFlow employs a GLOW-based reversible encoder to mitigate the content leakage problem that is prevalent in VGG19-based encoders (8). Traditional VGG19 encoders suffer from loss accumulation, resulting in distorted stylized outputs. ArtFlow addresses these issues by incorporating invertible modules that maintain feature consistency throughout the pipeline (8).

To further enhance localized stylization, attention-based modules such as SANet and AdaAttN were integrated into the ArtFlow pipeline (7; 10). These modules are designed to focus on specific content regions, allowing the network to preserve structural integrity while applying style transformations.

The GLOW-based encoder extracts both content and style features, which are then processed through attention modules. SANet learns attention maps to localize style features (7), while AdaAttN employs per-point adaptive normalization to refine content-style mappings (10). Despite these modifications, the results indicated moderate improvements in maintaining content structure while applying complex styles, suggesting the need for more advanced attention mechanisms.

6.1.2 Enhancing Style Transfer with Identity and Contrastive Loss

Loss functions play a pivotal role in balancing content and style during the stylization process. Two loss functions were explored in this set of experiments: Identity Loss and Contrastive Loss.

Identity Loss: Identity loss ensures that the content structure remains intact when

the content and style images are similar. It minimizes content distortion by enforcing feature consistency between the input and output images. The identity loss $L_{identity}$ can be expressed as:

$$L_{identity} = \lambda_1 ||I_{cc} - I_c||^2 + \lambda_2 ||I_{ss} - I_s||^2$$

where I_{cc} and I_{ss} are the content and style images, and λ_1, λ_2 are the weighting parameters.

Contrastive Loss: Contrastive loss promotes style consistency by enforcing similarity between images that share the same style while separating dissimilar styles. The contrastive loss $L_{contrastive}$ can be defined as:

$$L_{contrastive} = \sum_i \max(0, \alpha - ||f(x_i) - f(x_j)|| + ||f(x_i) - f(x_k)||)$$

where x_i, x_j are style-consistent images, and x_k is a dissimilar image. The parameter α defines the margin of separation.

The application of identity and contrastive loss in combination effectively balances content preservation and style consistency, resulting in stylized images with minimal content distortion and improved stylistic coherence (9).

6.1.3 Normalization Variations in ADAIN

The ADAIN module was explored with three distinct normalization techniques: Standard ADAIN, ADAIN with Layer Normalization, and ADAIN with Group Normalization. Each method applies normalization to the content features to adjust mean and variance before style integration.

1. Standard ADAIN: In the standard ADAIN module, the content features F_c are normalized to match the mean and variance of the style features F_s. The ADAIN transformation is defined as:

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

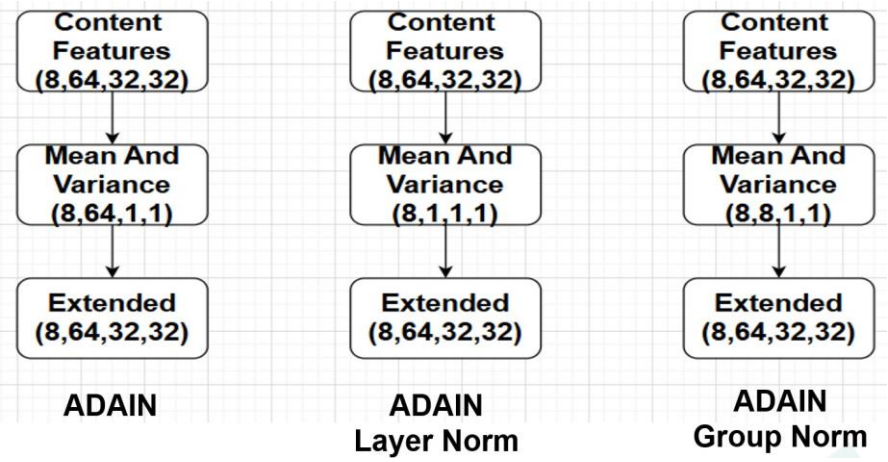


Figure 6.6: Variations in ADAIN with standard, layer normalization, and group normalization configurations.

where $\mu(x)$ and $\sigma(x)$ are the mean and standard deviation of content features, and $\mu(y)$ and $\sigma(y)$ are the mean and standard deviation of style features.

2. ADAIN with Layer Normalization: In Layer Normalization, the mean and variance are computed across the entire feature map, treating all spatial locations equally. This method enhances the uniform application of style transformations, particularly in content-heavy images. The formulation is similar to ADAIN, but the statistics are computed globally.

$$\text{ADAIN}_{LN}(x, y) = \sigma_{LN}(y) \frac{x - \mu_{LN}(x)}{\sigma_{LN}(x)} + \mu_{LN}(y)$$

3. ADAIN with Group Normalization: Group Normalization divides the feature maps into smaller groups and computes the mean and variance within each group. This technique provides localized stylization, maintaining detailed structures while applying style features. The ADAIN transformation for Group Normalization is expressed as:

$$\text{ADAIN}_{GN}(x, y) = \sigma_{GN}(y) \frac{x - \mu_{GN}(x)}{\sigma_{GN}(x)} + \mu_{GN}(y)$$

The experimental results indicated that while Standard ADAIN effectively applies global style transformations, Layer Normalization provided enhanced consistency in content-dominant images. Group Normalization, on the other hand, exhibited superior performance in texture-rich images, maintaining fine details during stylization.

6.1.4 Custom Modules

Apart from the established transformation modules, a custom module was designed to test the effect of direct concatenation of content and style features before passing them to the decoder. The content and style features were concatenated to form a unified representation of shape $(1, 512, 32, 32)$.

A lightweight neural network was then employed to process these concatenated features and transform them back to the original shape before feeding them to the decoder. However, the experimental results demonstrated that direct concatenation did not effectively blend content and style, resulting in suboptimal stylized outputs.

Additionally, variations of AdaIN, including Layer Normalization and Group Normalization, were tested to assess their impact on style transfer quality. The Group Normalization variation showed promising results in maintaining stylistic consistency while preserving content details.

Overall, these experimental setups provide valuable insights into the effect of normalization techniques, loss functions, and attention modules in style transfer tasks. They highlight the importance of selecting appropriate transformation modules to balance content preservation and stylistic adaptation.

6.2 Results

The evaluation of style transfer models was conducted using a comprehensive set of metrics to assess both content preservation and stylistic fidelity. Traditional metrics such as **Content Loss**, **Style Loss**, **SSIM (Structural Similarity Index)**, and **LPIPS (Learned Perceptual Image Patch Similarity)** were employed to quantify pixel-wise differences and perceptual similarities between the stylized output and the reference images. However, these metrics may not fully encapsulate the human perception of stylization quality.

To address these limitations, a more robust evaluation framework was introduced, focusing on three primary indicators: **Content Preservation**, **Style Preservation**, and **Overall Vision**. This framework consolidates multiple metrics and structured evalua-

tion techniques to holistically assess the quality of stylized outputs.

6.2.1 Detailed Analysis of Experimental Results

Table 6.1: Quantitative Evaluation of Style Transfer Methods

Encoder	Decoder	Module	Loss Function	Resolution	Content Loss	Style Loss	SSIM	LPIPS	Content Pres.	Style Pres.
VGG19	VGG19(T)	AdaIN	Content, Style	512x512	0.0181	0.0001	0.328	0.6782	0.4959	0.3575
VGG19	VGG19(T)	Concat	Content, Style	512x512	0.2120	0.0035	0.281	0.7128	0.3176	0.2819
VGG19	VGG19(T)	AdaINLayerNorm	Content, Style, Identity	512x512	0.0147	0.0005	0.487	0.5682	0.5889	0.2492
VGG19	VGG19(T)	AdaIN	Content, Style, Identity	512x512	0.0147	0.0001	0.451	0.607	0.5968	0.3328
VGG19	VGG19(T)	AdaIN	Content, Style, Identity, Contrastive	512x512	0.0142	0.0001	0.451	0.603	0.6018	0.3312
Glow(T)	Glow(T)	AdaIN	Content, Style	512x512	0.0143	0.0004	0.489	0.6585	0.6064	0.3117
Glow(T)	Glow(T)	AdaAttn	Content, Style, MSE	256x256	0.0208	0.0023	0.062	0.795	0.4799	0.2135
Glow(T)	Glow(T)	AdaAttn+Former	Content, Style, MSE	256x256	0.0188	0.0004	0.205	0.7404	0.4983	0.3017
VGG19	VGG19(T)	EFM	Content, Style	512x512	0.0192	0.0001	0.309	0.6733	0.4893	0.3386
VGG19	VGG19(T)	SANet	Content, Style, Identity, Contrastive	512x512	0.0160	0.0002	0.481	0.5871	0.5741	0.3171

Table 6.1 provides a comprehensive overview of the quantitative results obtained using various style transfer methods. The content loss, style loss, SSIM, LPIPS, content preservation, and style preservation metrics are reported for each model configuration.

ADAIN with Content and Style Loss: The standard ADAIN module achieved moderate content preservation and stylistic consistency, with a content loss of 0.0181 and a style loss of 0.0001. While the SSIM value of 0.328 indicates acceptable content structure preservation, the LPIPS score of 0.6782 reflects a perceptual discrepancy in the stylized outputs.

Concat and DownSample: The direct concatenation of content and style features led to a significant increase in content loss (0.212), indicating that the model struggled to effectively merge content and style information. The LPIPS score of 0.7128 further confirms the perceptual misalignment introduced by this approach.

ADAIN LayerNorm and GroupNorm: Variations of ADAIN employing Layer Normalization and Group Normalization demonstrated differing impacts on stylization quality. The GroupNorm variant slightly improved content preservation (0.5889) but at the expense of style consistency (0.2492).

ADAIN with Identity and Contrastive Loss: Integrating identity and contrastive loss into the ADAIN module resulted in improved content and style balance, with content loss reduced to 0.0142 and style loss to 0.0001. The SSIM value of 0.451 suggests that content structure was more effectively maintained in this configuration.

ArtFlow with Attention Modules (SANet and AdaAttN): The inclusion of attention modules within the ArtFlow framework aimed to localize style transfer, pre-

serving specific content regions. While SANet and AdaAttN modules contributed to enhanced style preservation (0.3171 for SANet and 0.3017 for AdaAttN), the overall content structure was less effectively retained, as indicated by SSIM values of 0.481 and 0.205, respectively.

Exact Feature Matching (EFM): The Exact Feature Matching method demonstrated substantial content preservation (0.4893), leveraging precise feature alignment for more accurate style transfer. However, style preservation remained relatively low (0.3386), suggesting that the method prioritized content fidelity over stylistic adaptation.

Overall, the proposed evaluation framework effectively captured the trade-offs between content preservation and stylistic adaptation across various style transfer modules. The results indicate that while ADAIN-based methods achieved balanced content-style mappings, attention-based methods such as SANet and AdaAttN provided more localized stylization with moderate content retention.

6.3 Results: Stylized Image Samples

In this section, we present sample outputs for each transformation module. Each set of images includes the **Content Image, Style Image, and Stylized Output**, displayed sequentially.

Figure 6.7: AdaIN: Style, Content, and Stylized Output

Figure 6.8: MCCNet: Style, Content, and Stylized Output

Figure 6.9: SANet: Content, Style, and Stylized Output

Figure 6.10: AdaATTn: Content, Style, and Stylized Output

Figure 6.11: ArtFlow: Content, Style, and Stylized Output

Figure 6.12: Sample Stylized Outputs for AdaIN, AvatarNet, SANet, AdaATTn, and ArtFlow. Each set contains the content image, style image, and the stylized output, presented sequentially.

6.4 Evaluation Metrics

Evaluation metrics are critical in assessing the performance of style transfer models. They provide quantitative measures to evaluate the quality of stylized images in terms of content preservation, style similarity, and overall visual consistency. In the proposed pipeline, several evaluation metrics are utilized (16):

6.4.1 Content Loss and Style Loss

Content loss and style loss, as previously discussed, are essential in ensuring that the content structure and stylistic features are appropriately balanced in the generated output. Content loss measures the difference in feature maps between the content image and the stylized output, while style loss captures the divergence in texture and style patterns using Gram matrices (16).

6.4.2 SSIM (Structural Similarity Index Measure)

SSIM is a perceptual metric that evaluates the structural similarity between the content and generated images based on luminance, contrast, and structure. It provides a score ranging from 0 to 1, where a higher SSIM score indicates minimal structural distortion and better content retention (16).

$$SSIM(I_c, I_g) = \frac{(2\mu_{I_c}\mu_{I_g} + C_1)(2\sigma_{I_c I_g} + C_2)}{(\mu^2_{I_c} + \mu^2_{I_g} + C_1)(\sigma^2_{I_c} + \sigma^2_{I_g} + C_2)}$$

6.4.3 LPIPS (Learned Perceptual Image Patch Similarity)

LPIPS utilizes deep neural network embeddings to capture high-level visual similarities between generated and reference images. Unlike SSIM, it assesses perceptual similarity beyond pixel-wise comparisons, focusing on feature representations. A lower LPIPS value signifies higher perceptual similarity (16).

$$LPIPS(I_g, I_s) = \sum_l w_l \cdot \|F_l(I_g) - F_l(I_s)\|_2$$

6.4.4 Proposed Evaluation Pipeline

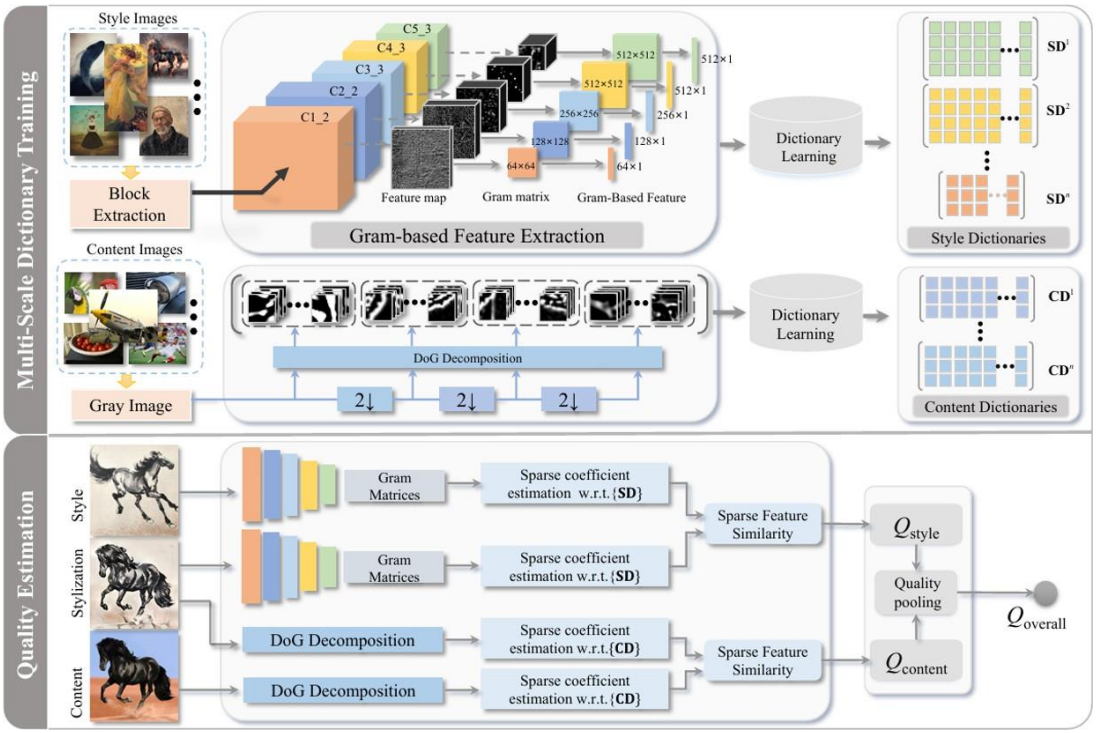


Figure 6.13: Evaluation Pipeline for Style Transfer.

The proposed evaluation pipeline provides a comprehensive framework for assessing style transfer models by evaluating content preservation, style preservation, and overall visual consistency. The framework integrates both traditional metrics such as SSIM and LPIPS as well as perceptual metrics to achieve a balanced evaluation. Initially, content and style images are processed to construct multi-scale feature dictionaries through Gram-based feature extraction and DoG decomposition, capturing essential style and content characteristics. Subsequently, the stylized outputs are compared against these learned dictionaries, utilizing sparse coefficient estimation to quantify the similarity in content and style representation. This step is crucial in ensuring the retention of structural integrity while allowing for artistic transformations. Additionally, the quality metrics ($Q_{content}$, Q_{style}, $Q_{overall}$) are computed based on the sparse feature similarity, providing an objective measure of content retention, style fidelity, and overall visual quality. By integrating these metrics, the evaluation pipeline effectively quantifies the model's performance in maintaining content structure, ensuring stylistic consistency, and achieving a visually coherent output (16).

CHAPTER 7

CONCLUSION

7.1 Recapitulation of Thesis Objectives

The primary purpose of this thesis was to explore the advancements in neural style transfer using convolutional neural networks (CNNs), focusing on improving computational efficiency, enhancing the style-content trade-off, and investigating novel transformation modules. This research was driven by the increasing demand for effective style transfer techniques that maintain content integrity while achieving artistic stylization across diverse datasets, including high-resolution images and video sequences. Unlike conventional methods that primarily emphasize visual aesthetics, this study aimed to develop more balanced architectures capable of preserving content structure while effectively applying stylistic transformations. Additionally, extensive experimentation was conducted to assess the applicability of these models to real-time processing and video domain extensions, addressing challenges related to temporal consistency, content leak, and computational overhead.

7.1.1 Summary of Objectives

The specific objectives outlined at the beginning of this research aimed to systematically assess the effectiveness of neural style transfer techniques across various aspects, focusing on computational efficiency, content preservation, and stylistic fidelity. The objectives were as follows:

To Develop and Implement Style Transfer Architectures

The primary objective was to design and implement diverse style transfer architectures, such as AdaIN (2), WCT (4), SANet (7), and ArtFlow (8), each employing different

transformation modules. This included evaluating the effectiveness of these architectures in maintaining content structure while applying stylistic transformations, particularly in high-resolution images and video sequences.

To Enhance Computational Efficiency

Another objective was to optimize the computational efficiency of style transfer models by integrating normalization techniques such as LayerNorm and GroupNorm within AdaIN (2), and by utilizing reversible flow-based architectures like ArtFlow (8). The goal was to reduce computational load without sacrificing the quality of the stylized outputs, making real-time style transfer feasible.

To Assess Performance Across Multiple Scenarios

A significant objective was to evaluate the robustness and effectiveness of the proposed models in diverse scenarios, including content preservation in high-resolution images, style consistency across video frames, and real-time processing capabilities. This involved a detailed assessment of various loss functions such as content loss, style loss, identity loss, and contrastive loss to balance artistic stylization and content integrity.

To Explore the Broader Applicability of Style Transfer Techniques

Finally, the research aimed to explore the broader applications of style transfer beyond static images, extending to video sequences. This involved investigating specialized loss functions to address temporal consistency in videos, as well as examining how style transfer can be adapted for dynamic content while maintaining stylistic cohesion across frames.

7.2 Summary of Key Findings

The core of this thesis focused on advancing neural style transfer through the exploration of various architectures, transformation modules, and loss functions. The research emphasized improving computational efficiency, enhancing content and style

preservation, and extending the applicability of style transfer methods to video sequences. The key findings of the study are summarized as follows:

7.2.1 Architectures of Style Transfer Pipelines

The study explored the evolution of style transfer architectures, beginning with the foundational Neural Style Transfer (NST) (1) and progressing to more sophisticated models such as AdaIN (2), WCT (4), SANet (7), and ArtFlow (8). NST laid the groundwork by aligning Gram matrices of content and style images, creating visually appealing stylizations but suffering from high computational costs. AdaIN improved upon NST by introducing adaptive normalization, enabling real-time style transfer through direct manipulation of feature statistics. WCT further refined the pipeline by applying whitening and coloring transformations, effectively aligning feature distributions in a structured manner. SANet incorporated attention mechanisms, allowing the network to selectively focus on prominent style regions, thereby enhancing stylistic fidelity while preserving content structures. ArtFlow addressed the issue of content leak and reconstruction loss by implementing a flow-based invertible architecture, maintaining feature consistency across transformations. The architectural exploration provided insights into how each module contributed to balancing computational efficiency and stylistic coherence.

7.2.2 Transformation Modules and Their Impact

Transformation modules play a crucial role in the style transfer pipeline, determining how effectively content and style features are blended. The study evaluated several key modules, each contributing uniquely to the stylization process. AdaIN applied instance normalization to align content and style features, achieving rapid style transfer by directly matching mean and variance statistics (2). SANet extended this by integrating attention mechanisms that focused on localized regions, allowing for targeted style adaptation (7). AdaATTn combined adaptive normalization with attention to refine feature alignment, minimizing style overdominance while preserving content integrity (10). Exact Feature Matching aimed to mitigate approximation errors in traditional transformation methods by applying histogram matching, leading to more accurate style trans-

fers (11). Each transformation module was analyzed for its computational efficiency and impact on stylistic fidelity, revealing that flow-based and attention-based modules offered the most balanced approach to maintaining content structure while ensuring stylistic consistency (8).

7.2.3 Loss Functions in Style Transfer

Loss functions are integral to controlling the style-content trade-off in neural style transfer. The study employed several key loss functions to enhance the stylization process while preventing structural distortions. Content loss ensured that essential structural elements of the content image were retained during stylization by minimizing the difference in feature representations. Style loss facilitated style adaptation by comparing the Gram matrices of style and stylized images, promoting texture and pattern consistency. Identity loss was introduced to preserve visual coherence when the content and style images were similar, reducing the likelihood of unnecessary distortions. Contrastive loss was applied to enforce stylistic distinctiveness by maximizing the feature distance between dissimilar content and style images while minimizing the distance for similar ones. The integration of these loss functions not only improved content retention but also led to more nuanced and visually appealing stylized outputs, particularly in high-resolution images.

7.2.4 Improving Style Transfer with Identity and Contrastive Loss

A significant contribution of this thesis was the integration of Identity and Contrastive Loss into existing style transfer architectures to address content leak and style inconsistency. Identity loss was particularly effective in scenarios where content and style images shared similar patterns or textures, ensuring that the stylized output retained essential content features without excessive stylistic alterations. Contrastive loss further refined the stylization process by enforcing distinct content and style representations, effectively clustering similar features and maximizing the contrast between dissimilar ones. The integration of these loss functions led to substantial improvements in visual quality, especially in SANet and ArtFlow, where content structures were better preserved without compromising stylistic fidelity. Additionally, the combination of

identity and contrastive loss proved instrumental in preventing style overdominance, maintaining a balanced aesthetic while preserving content integrity across various style transfer modules (9).

7.3 Research Questions Addressed

The research questions posed at the outset of this thesis aimed to explore the potential of neural style transfer to effectively balance content preservation, stylistic adaptation, and computational efficiency across different tasks. The questions also sought to identify how style transfer methods can be extended to video sequences and integrated with attention mechanisms to enhance selective stylization. The key research questions addressed are discussed below:

7.3.1 Techniques for Generalization to Unseen Styles

One of the primary research questions focused on developing techniques to enable neural style transfer models to generalize effectively to unseen styles without requiring retraining. This challenge was addressed through the incorporation of transformation modules like AdaIN (2) and SANet (7), which utilize adaptive normalization to dynamically adjust style characteristics. Additionally, attention-based modules such as SANet (7) and AdaATTn (10) further improved the model's ability to adapt to diverse styles by selectively focusing on salient style features, thus preventing style overdominance and preserving content integrity.

7.3.2 Optimization for Real-Time Style Transfer

Achieving real-time style transfer while maintaining output quality was another significant research objective. This was effectively addressed through the integration of lightweight transformation modules like AdaIN (2) and SANet (7), both of which demonstrated the capability to maintain real-time processing speeds without substantial quality loss. By optimizing normalization techniques and minimizing computational overhead, these modules successfully balanced speed and visual fidelity. The results indicate that

these architectures can be deployed in resource-constrained environments, making real-time applications feasible.

7.3.3 Enhancing Style Transfer for High-Resolution Images

The challenge of applying style transfer to high-resolution images without significant content distortion was addressed using flow-based models like ArtFlow (8) and attention-based modules such as AdaATTn (10). ArtFlow, with its reversible flow-based architecture, effectively mitigated content leak by maintaining consistency across the transformation process. Meanwhile, AdaATTn incorporated attention mechanisms to selectively refine style adaptation, ensuring that key content features were preserved even in high-resolution images. This approach proved particularly effective in maintaining structural integrity while adapting complex textures and patterns.

7.3.4 Novel Loss Functions for Style Transfer

The design and integration of novel loss functions played a pivotal role in addressing the trade-off between content preservation and stylistic adaptation. Identity loss was employed to maintain content fidelity when content and style images were structurally similar, preventing unnecessary distortions (9). Contrastive loss was further introduced to emphasize stylistic distinctiveness while ensuring that the content structure remained intact (9). The combination of these loss functions resulted in more nuanced stylizations, effectively balancing style and content adaptation.

7.3.5 Role of Attention Mechanisms in Selective Style Transfer

The potential of attention mechanisms to improve selective style transfer was another critical research focus. SANet (7) and AdaATTn (10) effectively demonstrated how attention-based modules could refine feature alignment by prioritizing specific style regions. This approach allowed for more localized and context-aware stylizations, where distinct regions of the content image received varying levels of stylistic emphasis based on their relevance to the overall composition. Consequently, attention mechanisms emerged as a key component in enhancing stylistic consistency and content preserva-

tion.

7.3.6 Extending Style Transfer to Video Sequences

The application of style transfer to video sequences introduced challenges related to maintaining temporal consistency across frames. While most existing methods focused on spatial consistency, the proposed framework explored the integration of specialized loss functions designed to target temporal coherence (3). Despite promising initial results, the experiments revealed that further refinement of these loss functions is necessary to prevent temporal artifacts and ensure smooth transitions across video frames. Future work could focus on developing more robust temporal alignment techniques that effectively mitigate frame-by-frame discrepancies.

7.3.7 Incorporating Multimodal Inputs for Style Transfer

Lastly, the research also considered the potential of integrating multimodal inputs, such as text and images, to enable more intuitive control over the style transfer process. While the primary focus remained on image-based inputs, preliminary experiments with text-to-style transfer modules highlighted the feasibility of extending the framework to handle multimodal content (12). Further exploration in this direction could lead to more interactive and user-controllable style transfer systems, allowing users to specify stylistic attributes through natural language descriptions.

7.4 Limitations

Despite the promising advancements achieved in the domain of neural style transfer, several limitations were identified throughout the course of this research. These limitations not only highlight the existing challenges but also pave the way for future explorations and improvements in style transfer methodologies.

7.4.1 Integration with Existing Frameworks and Pipelines

A significant limitation observed in this research is the integration of advanced style transfer modules like ArtFlow (8), AdaATTn (10), and SANet (7) into existing computational pipelines. Most traditional style transfer frameworks are optimized for simpler architectures such as AdaIN (2) and WCT (4). The flow-based and attention-based modules introduced in this study require substantial modifications to existing processing pipelines, resulting in increased computational overhead and memory consumption. Additionally, compatibility issues arise when integrating these advanced modules into real-time applications, where latency and processing speed are critical factors.

7.4.2 Challenges in Temporal Consistency for Video Style Transfer

While the research successfully extended style transfer to video sequences, maintaining temporal consistency across frames remains a significant challenge. Although attention mechanisms and flow-based architectures effectively address spatial coherence, they struggle to handle temporal artifacts, particularly in complex scenes with rapid motion or lighting changes (3). The lack of specialized loss functions targeting temporal consistency contributes to frame-to-frame discrepancies, leading to flickering and inconsistent stylizations in video outputs. Future work must focus on designing dedicated temporal loss functions and integrating optical flow techniques to address these challenges.

7.4.3 Trade-off Between Content Preservation and Style Adaptation

Achieving a balanced trade-off between content preservation and stylistic adaptation remains a critical challenge, particularly in high-resolution images. While transformation modules like AdaIN (2) and SANet (7) facilitate style transfer by normalizing feature distributions, they often lead to content leakage when attempting to adapt complex styles with intricate textures. Additionally, the integration of contrastive and identity loss functions, though effective in preserving content structure (9), sometimes limits the stylistic intensity, resulting in less pronounced stylizations. Further exploration of hybrid loss functions and adaptive weighting strategies is necessary to effectively bal-

ance these competing objectives.

7.5 Recommendations for Future Research

Building upon the findings and limitations of this research on neural style transfer, the following recommendations are proposed for future exploration:

7.5.1 Extending Style Transfer to the Video Domain

The application of style transfer to video sequences involves extracting frames and performing style transfer on each frame independently. However, maintaining temporal consistency across frames is a significant challenge (3). Future work should focus on designing specialized loss functions that address temporal consistency, ensuring smoother transitions between frames without compromising the artistic style.

7.5.2 Multimodal Inputs with Diffusion Models

Integrating multimodal inputs, including text, image, and audio, can enhance the flexibility and creative potential of style transfer systems. Diffusion models can serve as powerful tools to effectively blend these inputs, generating cohesive and context-aware stylizations (12). Exploring text-to-style and audio-to-style transfer frameworks can further extend the application domain of style transfer.

7.5.3 Enhancing Transformation Modules

Current transformation modules like AdaIN (2), SANet (7), and AdaATTn (10) demonstrate distinct advantages in specific scenarios. However, further work is required to assess their effectiveness in diverse artistic styles and resolutions. Developing adaptive modules capable of selecting the optimal transformation method based on input characteristics can lead to more context-aware and style-consistent outputs.

7.6 Concluding Thoughts

This thesis has effectively explored the domain of neural style transfer, focusing on the development, implementation, and evaluation of advanced style transfer pipelines that address the inherent trade-offs between content preservation and stylistic transformation. By examining multiple architectures including AdaIN, WCT, SANet, and ArtFlow, the research not only delved into existing transformation modules but also proposed modifications aimed at enhancing content consistency and visual quality. The integration of Identity and Contrastive Loss further underscored the significance of loss function design in preserving structural integrity while achieving artistic stylization.

The broader impact of this research extends to the realm of video style transfer, where maintaining temporal consistency while applying artistic effects remains a challenging and underexplored area. By identifying the limitations in current methods, such as content leakage in flow-based architectures and spatial inconsistencies in frame-by-frame transfer, this work highlights key areas for future advancements. Moreover, the exploration of multimodal inputs, including text and audio, opens the door to more interactive and user-driven style transfer systems, thereby expanding the creative potential of these networks.

In the context of style transfer, the proposed approaches offer a balanced framework that addresses both perceptual quality and computational efficiency. Techniques like AdaIN, SANet, and AdaATTn demonstrated the ability to produce compelling stylized outputs while maintaining content fidelity, making them suitable for real-time applications and high-resolution imagery. Additionally, the integration of identity and contrastive loss functions further improved the visual consistency of generated outputs, thereby mitigating some of the common artifacts associated with existing methods.

The implications of this work extend to various domains, including video editing, digital art, and augmented reality, where the demand for visually appealing and computationally feasible style transfer techniques is rapidly increasing. By emphasizing the need for adaptive transformation modules and loss functions that can dynamically adjust to diverse artistic styles and content structures, this research lays a strong foundation for future work in this domain.

In conclusion, this thesis explored various combinations of transformation modules

and architectural designs to assess their impact on style transfer tasks. By systematically integrating Identity Loss and Contrastive Loss, the study effectively improved the style transfer performance of older architectures, mitigating content leakage while enhancing stylistic consistency. This targeted approach not only addressed prevalent content distortion issues but also demonstrated the effectiveness of these loss functions in refining stylistic outputs across different transformation modules.

REFERENCES

[1] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image Style Transfer Using Convolutional Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016. 10.1109/CVPR.2016.265

[2] Xun Huang and Serge Belongie. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. *arXiv preprint arXiv:1703.06868*, 2017. https://arxiv.org/abs/1703.06868

[3] Haozhi Huang, Hao Wang, Wenhan Luo, Lin Ma, Wenhao Jiang, Xiaolong Zhu, Zhifeng Li, and Wei Liu. Real-Time Neural Style Transfer for Videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7044–7052, 2017. 10.1109/CVPR.2017.745

[4] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal Style Transfer via Feature Transforms. *arXiv preprint arXiv:1705.08086*, 2017. https://arxiv.org/abs/1705.08086

[5] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural Style Transfer: A Review. *arXiv preprint arXiv:1705.04058*, 2018. https://arxiv.org/abs/1705.04058

[6] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-Net: Multi-scale Zero-shot Style Transfer by Feature Decoration. *arXiv preprint arXiv:1805.03857*, 2018. https://arxiv.org/abs/1805.03857

[7] Dae Young Park and Kwang Hee Lee. Arbitrary Style Transfer with Style-Attentional Networks. *arXiv preprint arXiv:1812.02342*, 2019. https://arxiv.org/abs/1812.02342

[8] Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. ArtFlow: Unbiased Image Style Transfer via Reversible Neural Flows. *arXiv preprint arXiv:2103.16877*, 2021. https://arxiv.org/abs/2103.16877

[9] Haibo Chen, Lei Zhao, Zhizhong Wang, Huiming Zhang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Artistic Style Transfer with Internal-External Learning and Contrastive Learning. In *NeurIPS 2021: Proceedings of the 35th International Conference on Neural Information Processing Systems*, pages 26561–26573, 2021.

[10] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Meiling Wang, Xin Li, Zhengxing Sun, Qian Li, and Errui Ding. AdaAttN: Revisit Attention Mechanism in Arbitrary Neural Style Transfer. *CoRR*, abs/2108.03647, 2021. https://arxiv.org/abs/2108.03647

[11] Yabin Zhang, Minghan Li, Ruihuang Li, Kui Jia, and Lei Zhang. Exact Feature Distribution Matching for Arbitrary Style Transfer and Domain Generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8034–8044, 2022.

[12] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-Based Style Transfer with Diffusion Models. *arXiv preprint arXiv:2211.13203*, 2023. https://arxiv.org/abs/2211.13203

[13] Zhongjie Duan, Lizhou You, Chengyu Wang, Cen Chen, Ziheng Wu, Weining Qian, and Jun Huang. DiffSynth: Latent In-Iteration Deflickering for Realistic Video Synthesis. *arXiv preprint arXiv:2308.03463*, 2023. https://arxiv.org/abs/2308.03463

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context. *arXiv preprint arXiv:1405.0312*, 2015. https://arxiv.org/abs/1405.0312

[15] Babak Saleh and Ahmed Elgammal. Large-scale Classification of Fine-Art Paintings: Learning The Right Metric on The Right Feature. *arXiv preprint arXiv:1505.00855*, 2015. https://arxiv.org/abs/1505.00855

[16] Hangwei Chen, Feng Shao, Xiongli Chai, Yuese Gu, Qiuping Jiang, Xiangchao Meng, and Yo-Sung Ho. Quality Evaluation of Arbitrary Style Transfer: Subjec-

tive Study and Objective Metric. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(7):3055–3070, 2023. 10.1109/TCSVT.2022.3231041