Image Styling and Transformation

Amarpreet Kaur

amarpreet.kaur@torontomu.ca

Toronto Metropolitan University

Abstract

Visualizing image styling and transformations can be costly and time-consuming. Applying different arbitrary visual styles before starting the remodeling work helps align with the vision. The preliminary application of diverse visual styles aids in aligning the initial remodeling work with the envisioned outcome. Existing feed-forward based methods offer the advantage of efficient inference but typically fall short in their ability to generalize to unseen styles or to maintain high visual quality. In this paper, I introduce a novel and straightforward method that overcomes these limitations. my approach leverages an autoencoder architecture enhanced by block training, high-frequency residual skip connections, and bottleneck feature aggregation. This framework facilitates photorealistic style transfer across various domains, including interior design, fashion, product design, virtual home staging, art, photography, digital marketing, and virtual reality. Utilizing neural representations, my system uniquely separates and recombines the content and style of arbitrary images, establishing a robust neural algorithm for the creative generation of styled images. I demonstrate the effectiveness of my algorithm through the generation of high-quality stylized images, showcasing superior performance and versatility when compared to several contemporary methods. Furthermore, my approach's adaptability allows for dynamic adjustments, enabling real-time application in virtual environments and interactive design workflows, significantly enhancing user engagement and creative output across multiple industries.

Keywords: photorealistic style transfer, visual remodeling, autoencoder, neural algorithms.

1. Introduction

The increasing demand for digital visualization across various industries underscores a significant challenge that I aim to address in my project: accurately and efficiently visualizing changes before they are implemented. This challenge spans numerous domains such

as interior design, fashion, product design, virtual home staging, art, photography, digital marketing, and virtual reality. The need for an effective solution arises from the high costs and extensive time required by traditional styling methods, which often involve manual iterations and can result in dissatisfaction due to discrepancies between envisioned and actual outcomes.

Historically, Deep Neural Networks (DNNs) have been effective in areas like object and face recognition, setting the stage for their application in image processing. Notably, Gatys, Ecker & Bethge [5] in 2015 demonstrated the potential of neural style transfer by applying the styles of famous artworks to natural images, separating and recombining their content and style representations. This pioneering work opened up new avenues for applying similar techniques in more practical, everyday contexts.

In the evolving landscape of artificial intelligence, Deep Neural Networks (DNNs) have significantly advanced the frontier of image processing capabilities. Historically recognized for their prowess in object and face recognition, Deep Neural Networks (DNNs) have progressively transcended conventional boundaries, extending their utility to more creative domains such as artistic style transfer. The seminal work by Gatys, Ecker & Bethge [5] in 2015 marked a revolutionary turning point, demonstrating that these networks could not only mimic but also creatively repurpose the stylistic elements of renowned artworks onto everyday images. Their approach involved the innovative separation and recombination of content and style representations within images, leveraging the layered complexities of Convolutional Neural Networks (CNNs).

As Deep Neural Networks (DNNs) continue to evolve, their application in non-traditional fields like digital art and multimedia suggests a future where the line between technology and art becomes increasingly blurred, opening up new avenues for creative and practical applications. This integration of style transfer into neural network capabilities exemplifies the potential of Artificial intelligence (AI) to transform my approach to both understanding and creating visual art, making it an essential area of study in the field of computer vision and artificial intelligence.

Building upon foundational techniques in neural style transfer, Li et al. [8] in 2017 introduced a novel autoencoder-based approach that significantly streamlined the process and enhanced the scalability of applying style transfer on a larger scale. Their method utilizes an image reconstruction network that incorporates a pair of feature transforms—whitening and coloring transform (WCT)—to perform style transfer in a feed-forward manner without the need for retraining on new styles.

This advancement addresses previous limitations where models were either restricted to a fixed number of styles or compromised on visual quality. The whitening and coloring transform (WCT) method effectively matches the feature covariance of the content image with any given style image, thereby allowing the preservation of content while seamlessly applying the style attributes. This approach not only democratizes the application of style transfer across various styles but also maintains high fidelity in the stylization output,

making it a versatile tool in the broader context of automated digital art creation.

Moreover, Li et al. [8] work eliminates the dependency on iterative optimization, thus reducing the computational cost and making the process more efficient. By leveraging deep neural networks' feature extraction capabilities and coupling them with simple linear transforms, their method achieves an impressive balance between quality, efficiency, and flexibility in style application. This makes it particularly suitable for real-time applications and facilitates the creative manipulation of images at scale, opening new possibilities for enhancing visual media with artistic effects in various practical scenarios.

I expanded this work to include style transfer for general images. The input to my algorithm consists of an image used as the content and another image used as the style. My final model employs an autoencoder approach, similar to that of Li et al. [8], to output a generated image with the new style applied.

1.1 Motivation

In the rapidly evolving fields of design and digital media, the ability to visualize and apply diverse visual styles efficiently is critically important. Industries ranging from interior design to virtual reality rely heavily on being able to preview and modify visual content quickly to meet evolving aesthetic and functional demands. Traditional style transfer methods, while efficient, often struggle with generalization to unseen styles and typically fail to maintain high visual quality when applied across various domains. This limitation can lead to costly and time-consuming iterations in projects, hindering creativity and practical application. The need for a robust solution that offers both high-quality, photorealistic style transfer and the ability to dynamically apply diverse styles in real-time is more pressing than ever. Such a solution would not only streamline creative workflows but also enhance the practical utility of digital styling technologies across multiple industries.

In my project, I aim to extend the capabilities of this technology by adapting it to a broader range of applications beyond traditional artworks. I employ a sophisticated autoencoder architecture with block training, high-frequency residual skip connections, and bottleneck feature aggregation. This approach not only preserves the essential details of the original images but also ensures that the stylization is photorealistic and faithful to the target style.

My method is specifically designed to address the challenges of diverse image styling and transformation tasks. By enabling more accurate and real-time previews of potential changes across various domains, it aligns closely with user visions and reduces the likelihood of costly post-implementation modifications. Through the development and implementation of this technology, I aim to revolutionize how styling decisions are made, offering a dynamic, cost-effective, and time-efficient approach to transforming visual concepts into reality across all industries.

1.2 Research goals and scope

1.2.1 Scope of project

This project seeks to extend the capabilities of style transfer technology beyond its traditional confines of reproducing artistic impressions. By implementing a sophisticated autoencoder architecture equipped with block training, high-frequency residual skip connections, and bottleneck feature aggregation, the research aims to adapt this technology for a broader spectrum of applications. This approach not only meticulously preserves the essential details of the original images but also ensures that the stylization achieved is photorealistic and remains true to the desired aesthetic of the target style.

1.2.2 Research Goals

- **Enhanced Photorealism and Fidelity:** Develop an autoencoder that not only captures the essence of the desired style but does so in a way that the results are photorealistic, greatly enhancing the usability of the transformed images for professional applications.
- **Real-Time Processing and Preview:** Incorporate capabilities that allow for real-time adjustments and previews. This is vital for interactive applications where users need to see immediate results of their style choices, such as in virtual try-ons or digital interior design modifications.
- **Scalability and Versatility:** Ensure that the autoencoder can handle a diverse array of image styles and resolutions without a loss in performance. This includes scaling the technology to accommodate high-resolution images necessary for detailed and nuanced styling tasks.
- **Reduction of Post-Implementation Revisions:** By providing more accurate and adaptable previews that align closely with user expectations, the technology aims to minimize the need for costly and time-consuming post-implementation changes.

By addressing these goals, the project will not only advance the technical aspects of style transfer but also revolutionize its application across industries. This research is poised to transform how styling decisions are approached, providing a dynamic, cost-effective, and efficient method for translating visual concepts into tangible realities. Through the exploration of these advanced techniques and their applications, the project will contribute significantly to the field of digital image processing and transformation.

2. Literature review

In their 2020 research paper[2], "Ultrafast Photorealistic Style Transfer via Neural Architecture Search," Jie An, Haoyi Xiong, Jun Huan, and Jiebo Luo[2]address the longstanding

challenges in photorealistic style transfer by developing a novel two-step approach that leverages the power of neural architecture search to both create and optimize a network called PhotoNet. This method significantly enhances the speed and quality of style transfer without the need for intensive pre- or post-processing steps, featuring a construction step (C-step) to build the PhotoNet which incorporates advanced techniques such as Bottleneck Feature Aggregation (BFA) and Instance Normalized Skip Links (INSL) for improved detail preservation and photorealism. This is followed by a pruning step (P-step), utilizing the StyleNAS for network optimization, which results in PhotoNAS—a streamlined version of PhotoNet that maintains high-quality stylization effects while achieving speeds 20-30 times faster than existing methods. The success of PhotoNet and PhotoNAS in achieving superior detail preservation, as demonstrated through higher structural similarity indices and reduced Gram Loss, indicates not only an advancement in the field of style transfer but also showcases the potential of neural architecture search techniques in enhancing and accelerating photorealistic style transfers. Their work paves the way for further exploration of NAS in style transfer and potentially other areas of generative and low-level vision tasks, proposing a scalable, efficient solution that could revolutionize current practices in digital media and beyond.

In their 2021 paper [4]"PhotoWCT2: Compact Autoencoder for Photorealistic Style Transfer Resulting from Blockwise Training and Skip Connections of High-Frequency Residuals," Tai-Yin Chiu and Danna Gurari [4] tackle the inefficiencies of traditional photorealistic style transfer methods that struggle with large image resolutions and slow run-times due to their parameter-heavy designs. They introduce PhotoWCT2, a streamlined autoencoder that significantly enhances efficiency and supports higher resolutions like 4K UHD through innovative blockwise training and skip connections designed for high-frequency residuals. This approach not only preserves intricate image details during style transfer but also reduces the model's parameter count, resulting in faster stylization speeds without compromising the quality or realism of the transferred style. The successful implementation of PhotoWCT2 demonstrates a significant advancement in style transfer technology, enabling more practical applications in ultra-high-definition media [4].

In their groundbreaking 2015 study [5], "A Neural Algorithm of Artistic Style," Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge pioneer the application of Convolutional Neural Networks (CNNs) to the field of art, presenting a novel method for the creation of artistic images by separating and recombining the content and style of arbitrary images. This research significantly bridges the gap between human artistic capabilities and artificial systems by introducing a system that manipulates image content and style independently, offering a profound algorithmic insight into how humans perceive and create art. Utilizing CNNs to process visual information hierarchically, the methodology emphasizes separating content, captured in deeper network layers, from style, represented by the correlations across layers' filter responses. Through iterative gradient descent and

specialized loss functions for content and style, their approach synthesizes new images that maintain the structural essence of one image while adopting the stylistic elements of another. This process not only demonstrates the potential of neural networks to create high-quality artistic images but also enhances my understanding of the neural basis for art perception and style, marking a significant advancement in both computer vision and the application of artificial intelligence in creative domains.

In the transformative study [8] "Universal Style Transfer via Feature Transforms" by Li et al. [8] and colleagues, the team tackles the challenge of style transfer in image editing by introducing an innovative feed-forward approach that avoids the typical necessity for style-specific training. This method utilizes a pair of feature transforms known as Whitening and Coloring Transforms (WCTs), which are integrated into an image reconstruction network to effectively match the feature covariance of the style image to the content image. The approach aims to balance efficiency, generalization, and high-quality output without the reliance on predefined styles or extensive training for each new style. By employing a novel methodology that includes a reconstruction decoder trained to invert features from the VGG-19 network, multi-level stylization for enhanced detail, and user controls for customizing the degree of stylization, the study provides a robust solution to the limitations of existing optimization-based and feed-forward style transfer methods. The results demonstrate that this technique not only achieves high-quality artistic outputs but also supports arbitrary visual styles with greater flexibility and speed than traditional methods, positioning it as a significant advancement in the field of neural style transfer.

In the literature review of neural style transfer techniques, the paper "Understanding Generalized Whitening and Coloring Transform for Universal Style Transfer" Chiu [3] presents a comprehensive study of advanced style transfer methods. It begins with a background on the evolution of style transfer, emphasizing the shift from basic image quilting methods to sophisticated neural style transfer techniques that utilize deep convolutional networks. The aim of the paper is to explore the efficacy of Generalized Whitening and Coloring Transform (GWCT) in enhancing the universal applicability and computational efficiency of style transfer. The methodology employs advanced statistical techniques, particularly Zero-phase Component Analysis (ZCA), to adjust image feature covariances, enabling the high-fidelity transfer of artistic styles. Results from the study demonstrate that GWCT significantly improves style fidelity and content preservation across various styles and contents. The paper concludes that ZCA not only effectively captures and replicates style elements but also supports real-time, high-quality applications, making it a viable solution for universal style transfer.

In the context of photorealistic style transfer, the study outlined in "Photorealistic Style Transfer via Wavelet Transforms" Yoo et al. [13] introduces a groundbreaking approach to enhancing photorealism in style transfer applications. The research emphasizes the limitations of traditional style transfer techniques, which often result in spatial distortions and unrealistic artifacts. To address these challenges, the study proposes a

novel method employing wavelet transforms within an end-to-end style transfer model, which significantly improves the preservation of fine structural details without the need for post-processing. This method, known as Wavelet Corrected Transfer (WCT2), integrates whitening and coloring transforms to maintain the content's structural integrity during the style transfer process. The effectiveness of WCT2 is demonstrated through superior qualitative and quantitative results, outperforming existing methods in achieving photorealistic stylizations

3. Methods

3.1 Dataset and Features

To train the autoencoder, I utilized the *MSCOCO Dataset* [10], which is prominently featured in several WCT papers. This dataset includes 118,288 images in its training set, 5,000 images in the validation set, and 40,670 images in the test set. Additionally, I employed the *ADE20K Dataset* [1] for my initial semantic segmentation approach, which I will describe in more detail in the experiments section Section 4 below. The *ADE20K Dataset* [1] comprises 25,574 training images and 2,000 images in the validation set. This dataset is particularly valuable because it includes semantic segmentations of each scene, covering not just indoor rooms but also outdoor spaces, cities, factories, and more.

I used these images to produce the final stylized results. However, it's important to note that the segmentation of these images is not perfect, which presents certain challenges that I will discuss in the experiments section Section 4 of my report.

3.2 Methodology

I will discuss the method that my model performs style transfer through the previous papers it builds upon in this section. Like Li et al. [8] I used an autoencoder architecture to perform style transfer. My final model is an autoencoder architecture using the baseline of WCT (Whitening and coloring transforms). The VGG-19 network is used as the feature extractor (encoder) and a symmetric decoder is trained on images from the *MSCOCO Dataset* [10] to invert the VGG-19 features and reconstruct the content image.

3.2.1 Reconstruction decoder

I constructed an auto-encoder network for general image reconstruction. I utilized the VGG-19 network as the encoder and established a fixed decoder specifically for converting VGG features back to the original image, as depicted in Figure 1.

The structure of the decoder mirrors that of the VGG-19 up to the Relu_X_1 layer, employing nearest neighbor up sampling to expand feature maps. To conduct evaluations with features from different depths, I selected feature maps from five levels of the VGG-19, specifically Relu_X_1 (X=1,2,3,4,5), and developed five corresponding decoders. For

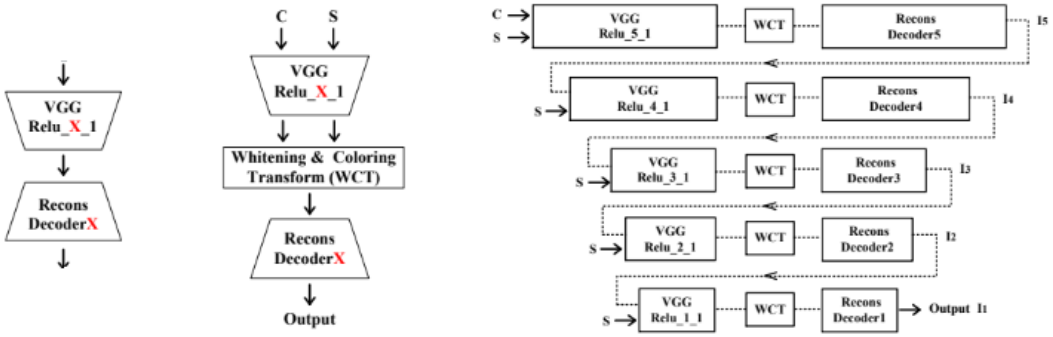

Figure 1: (a) Reconstruction (b) Single-level styling with content and style images. (c) Multi-level styling, enhancing detail through successive matches(Li et al. [8])

image reconstruction, I used both pixel reconstruction loss and feature loss as metrics. The formula for calculating losses is:

$$L = \|I_o - I_i\|_2^2 + \lambda \|\Phi(I_o) - \Phi(I_i)\|_2^2, \tag{1}$$

where I_i and I_o represent the input and reconstructed images, respectively, and Φ denotes the VGG encoder that extracts the Relu_X_1 features. The parameter λ serves to balance these two types of losses. Once trained, the decoder remains unchanged and serves as a feature inverter without further fine-tuning.

3.2.2 Whitening and coloring transforms

Given a pair of content image I_c and style image I_s, the vectorized VGG feature maps $f_c \in \mathbb{R}^{C \times H_c W_c}$ and $f_s \in \mathbb{R}^{C \times H_s W s}$ are extracted at a specific layer (e.g., Relu_4_1). Here, H_c, W_c (and H_s, Ws) are the height and width of the content (and style) feature maps, and C represents the number of channels. If f_c is fed directly into the decoder, the original image I_c is reconstructed. To adapt the content features to the style of I_s, a whitening and coloring transform (WCT) is applied to f_c to match the covariance matrix of f_s. This transformation includes two steps: a whitening and a coloring transform, modifying f_c to incorporate the stylistic elements of f_s.

Whitening transform: Before initiating the whitening process, the feature map f_c is centered by subtracting its mean vector m_c. This prepares f_c for the subsequent transformation according to equation (2), which linearly modifies f_c to produce a whitened feature map $\hat{f}_c$ with uncorrelated features ($\hat{f}_c \hat{f}_c^T = I$):

$$\hat{f}_c = E_c D_c^{-\frac{1}{2}} E_c^T f_c, \tag{2}$$

Here, D_c is a diagonal matrix containing the eigenvalues from the covariance matrix $f_c f_c^T \in \mathbb{R}^{C \times C}$, and E_c is the orthogonal matrix of eigenvectors, ensuring that $f_c f_c^T = E_c D_c E_c^T$.

To assess the effects of the whitening, $\hat{f}_c$ is converted back to RGB space using a

previously trained decoder. The visual analysis indicates that the process of whitening features in images effectively maintains the essential structure of the content while removing specific stylistic details. This demonstrates the whitening step's ability to isolate and eliminate style-related elements, preserving the basic structure of the image. Such a technique prepares the content representation, denoted as $\hat{f}_c$, for further adaptation, where it can be modified to adopt the artistic characteristics of a different target style.

Coloring transform: To begin the coloring transform [12] , the style feature map f_s is first centered by subtracting its mean vector m_s. This step sets the stage for the subsequent transformation of the whitened feature map $\hat{f}_c$ according to equation (3). The coloring transform is effectively the inverse of the whitening process, aiming to adjust $\hat{f}_c$ so that its feature map correlations align with those of f_s:

$$\hat{f}_{cs} = E_s D_s^{\frac{1}{2}} E_s^T \hat{f}_c, \tag{3}$$

where D_s is a diagonal matrix containing the eigenvalues of the covariance matrix $f_s f_s^T \in \mathbb{R}^{C\times C}$, and E_s is the orthogonal matrix of eigenvectors corresponding to those eigenvalues. This transformation ensures that the correlations between the feature maps of $\hat{f}_{cs}$ match those of f_s.

After applying the coloring transformation, the newly transformed feature map $\hat{f}_{cs}$ is re-centered by adding back the mean vector m_s of the style. This final adjustment integrates the mean characteristics of the style into $\hat{f}_{cs}$, completing the process to infuse the target style's characteristics into the content image's structural framework.

For my autoencoder looked at related research in the field to guide my design process. Two other related models sought to improve upon the vanilla autoencoder by modifying the down sampling and up sampling layer.In PhotoWCT, it was replaced with pooling and unpooling to preserve spatial information, reduce artifacts and make the image more photorealistic. An additional post smoothing step was also introduced to remove artifacts using the original content image [7]. In WCT^2[7] , the pooling and unpooling layer in the VGG encoder and decoder was replaced with a wave corrected transfer that would perfectly reconstruct a signal without post processing steps. WCT^2[7] introduced skip connections as well as usage of segmented images to produce better results [13]. An et al. [2] then tackled the problem of the lack of style introduced in WCT^2[7] 's skip connections and the need for segmented images to produce photorealistic results. An autoencoder called PhotoNetAn et al. [2] was used with the pretrained VGG-19 as the encoder and a decoder that would reconstruct the image, similar to the variations of WCT from before. However, PhotoNet also introduces a bottleneck feature aggregation (BFA) module at the bottleneck which concatenates multi-scale features produced by different levels of the network.

3.2.3 Bottleneck feature aggregation (BFA)

Feature aggregation is a network module that concatenates multi-scale features produced by different layers of deep networks. Feature aggregation enables networks to integrate information from different field-of-views, thus may enhance low-level detail preservation of stylization that happens in high-level features. Based on this,bottleneck feature ag-

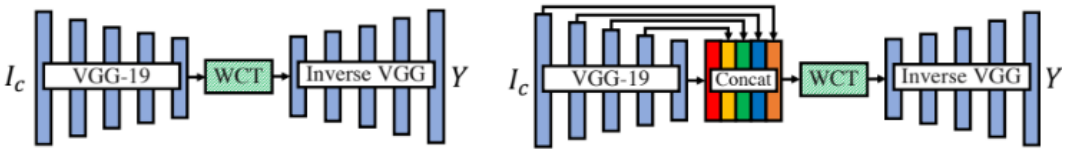

Figure 2: (a) Vanilla auto-encoder (b) Vanilla auto-encoder + BFA (An et al. [2])

gregation (BFA) module was introduced to the auto-encoder. In detail, I first resize features from Relu_1_1 to Relu_4_1 to the size of Relu_5_1 in the VGG encoder, then I concatenate them together at the bottleneck.

Feature aggregation enables networks to integrate information from different fields-of-views, thus may enhance low-level detail preservation of stylization that happens in high-level features and lead to more details in the reconstructed image.

Furthermore, An et al. [2] replaced the skip connections in WCT^2 [13] with Instance Normalized Skip Links (INSL) as WCT^2 [13] generally lost its ability to produce stylized images since the short circuit could block the information stream flow into transfer module work at the bottleneck.

Instance Normalized Skip Links (INSL): An et al. [2] introduced the concept of Skip Connection (SC), first deployed by FCN [9] and U-Net [11] noting its significant enhancement of segmentation outcomes. However, it was observed that auto-encoders equipped with SC often lose their capacity to generate stylized images, as SCs at critical junctures may render the transfer modules at the auto-encoder's bottleneck ineffective. This is referred to as the "short circuit" phenomenon. INSL seems to alleviate the short circuit phenomenon and strengthen the detail preservation and distortion elimination abilities of photorealistic style transfer networks.

Chiu & Gurari [4] introduced block wise training to perform coarse-to-fine feature transformations in a single autoencoder instead of the cascade of fmy autoencoders used in PhotoWCT [6]. PhotoWCT [6] consists of a cascade of four autoencoders AECN 's (N = 1, 2, 3, 4), where each includes an encoder encN and decoder decN . encN is a pretrained network, specifically VGGNet, from the input layer to the Relu_1 layer. decN is structurally symmetric to encN .

To realize the coarse-to-fine feature transformation, the cascade of four autoencoders is in the order from N = 4 to N = 1. Specifically, content and style images are first encoded by enc4 into the Relu_4_1 features. The Relu_4_1 content feature is then transformed with reference to the Relu_4_1 style feature using a ZCA feature transformation [[8], [3]]. The transformed feature is then decoded by dec4 to become an image. The three steps

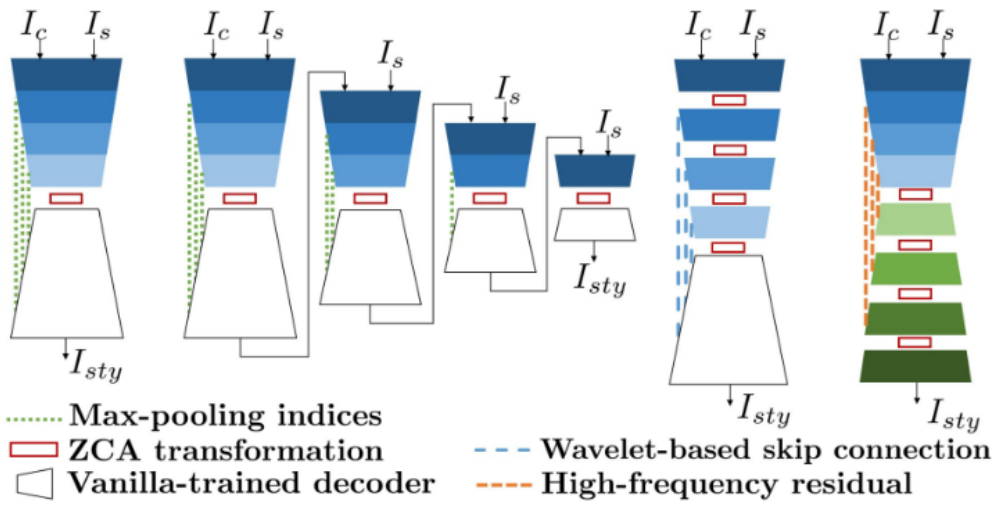

Figure 3: (a) WCT (b) PhotoWCT (c) WCT^2 (d) PhotoWCT^2 (Chiu & Gurari [4])

of encoding, transformation, and decoding repeat in the next three rounds of N = 3, 2, 1, with N+1 as the content image, until the stylized image is decoded by dec1. Finally, image smoothing (using guided filtering) is applied as a post-processing step to remove undesired artifacts in the final stylized image.The course-to-fine feature transformations was also an improvement from WCT^2 [13] that had fine-to-course transformations which had weaker stylization strengths as the fine-tune details may have been overshadowed later by coarser big-picture modifications.

3.2.4 Blockwise training

Further in project two methods are proposed: end-to-end training and blockwise training to achieve four function inversions for the decoder. End-to-end training allows the decoder to simultaneously learn the four function inversions. Although this method suffices for inverting the functions, blockwise training is introduced as an improvement. Blockwise training, as illustrated in Figure 4, organizes the learning of the four function inversions into four stages. In each stage, a specific decoder block, labeled as decbtblkN, learns the inverse function corresponding to enc4blkN.

Mathematically, each decoder block, decbtblkN, is sequentially trained from N = 1 to N = 4 by minimizing the loss function LN:

$$\begin{aligned}\mathcal{L}_N(I) = \mathbb{1}_{N\neq 1}||\phi_{N-1}(I) - dec_{bt}blk_N(\phi_N(I))||_2^2 \\ + ||I - \psi_N(\phi_N(I))||_2^2 \\ + \lambda||\phi_N(I) - \phi_N(\psi_N(\phi_N(I)))||_2^2,\end{aligned} \tag{4}$$

where φ_N and ψ_N represent the series of encoder and decoder functions respectively, from $\{\text{enc4blk1}, \ldots, \text{enc4blk}N\}$ to $\{\text{decbtblk}N, \ldots, \text{decbtblk1}\}$. The indicator function $1_{N\neq 1}$ equals one when $N \neq 1$ and zero when $N = 1$. The coefficient λ is set to one for $N \neq 1$ and zero for $N = 1$. The three components of the equation represent function inversion,

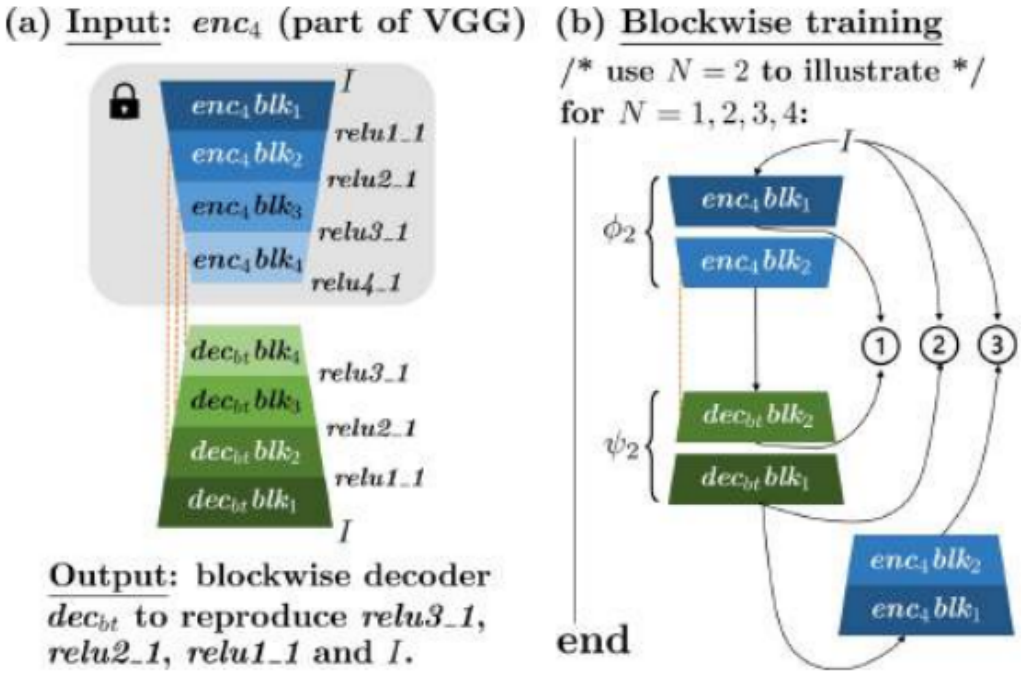

Figure 4: (a) Overview of PhotoWCT^2 (b) Model with blockwise training needed to effectively support coarse-to-fine feature trans-formation in a single autoencoder (Chiu & Gurari [4])

image reconstruction, and perceptual losses, respectively. During training, previously trained blocks and the encoder remain fixed.

3.2.5 Skip connections of high-frequency residuals

Skip connections of high-frequency residuals were introduced to preserve image quality when applying the sequential coarse-to-fine feature transformations. This preserved the advantages of WCT^2 [13]'s better image reconstruction with less parameters.

$$\begin{aligned} \mathbf{F}'_{LL} &= (\mathcal{K}_{LL} * \mathrm{AEC}_{part}((\mathcal{K}_{LL} * \mathbf{F})_{\downarrow 2}))_{\uparrow 2} \\ &\approx (\mathcal{K}_{LL} * (\mathcal{K}_{LL} * \mathbf{F})_{\downarrow 2}))_{\uparrow 2} = \tilde{\mathbf{F}}_{LL} \end{aligned} \tag{5}$$

The skip connection of a high-frequency residual, as illustrated in Figure 5, supports the goals of both end-to-end and blockwise trainings for feature/image reconstruction. This approach simplifies wavelet-based skip connections (introduced in Yoo et al. [13]) into a less computationally demanding representation by substituting Haar convolution-sYoo et al. [13] with average pooling, upsampling, and subtraction, and Haar deconvolutions with upsampling and addition. The wavelet pooling layer, when processing a feature F, outputs a low- frequency component $\tilde{F}_{LL}$ and three high-frequency components: $\tilde{F}_{LH}$, $\tilde{F}_{HL}$, and $\tilde{F}_{HH}$. Structurally, $\tilde{F}_{LL}$ propagates through the middle layers of the network, denoted by encpart-decpart AEC_{part}. This method transforms the concatenation of $\tilde{F}_{LL}, \tilde{F}_{LH}, \tilde{F}_{HL}$, and $\tilde{F}_{HH}$ for wavelet-based skip connections into an addition for this approach, allowing the channel length of the resulting $\tilde{F}_{sum}$ to become one fourth of that

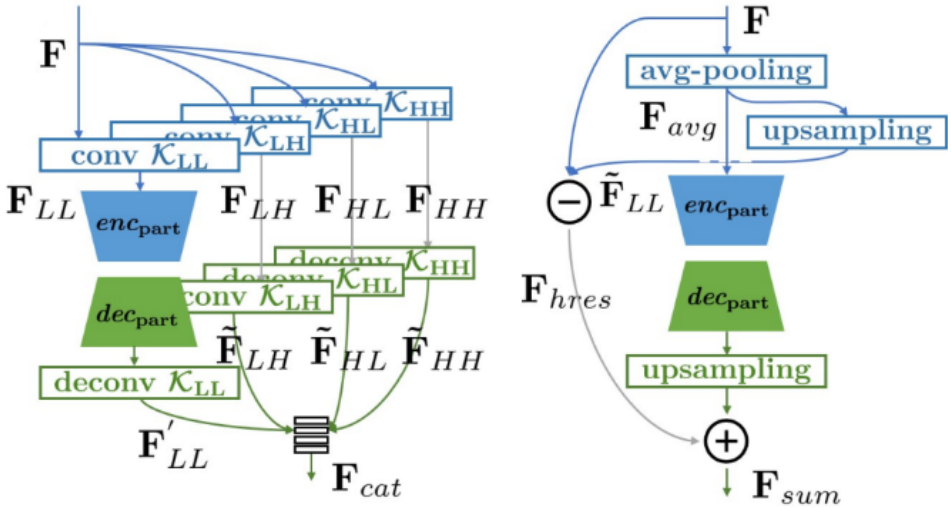

Figure 5: (a) WCT^2 (b) Model with skip connection of the high-frequency residual(Chiu & Gurari [4])

of $\tilde{F}_{cat}$ from the wavelet-based skip connection. The motivation for this addition-based approach stems from an observation: low-frequency parts of an image are often better reconstructed by an autoencoder than the high-frequency edges. Utilizing this observation, it is assumed that a low-frequency feature f can be approximately reconstructed by $\text{AEC}_{\text{part}}(f) \approx f$.

3.3 Project Method

Considering the approaches discussed in details in Section 3.2 , I chose to make a model that combined the approaches in PhotoNas [2] with that of PhotoWCT^2 [4] . Specifically, I added bottleneck feature aggregation, BFA, from PhotoNas [2] to PhotoWCT^2 [4] to help further preserve details in the content image. I kept the high-frequency residuals from PhotoWCT^2 [4]

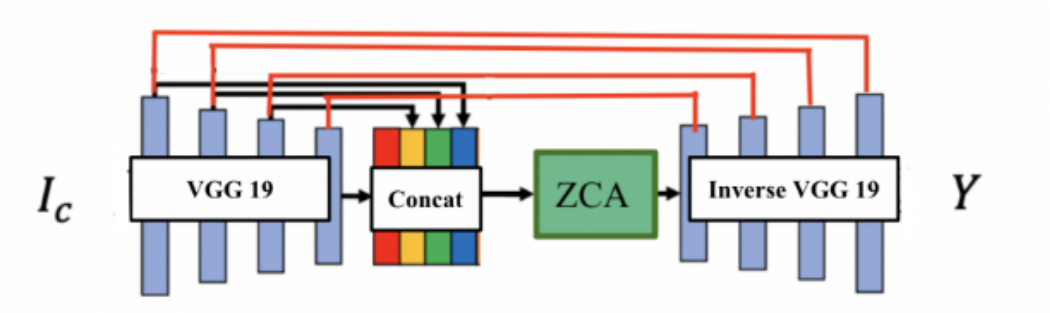

Figure 6: My autoencoder architecture with a BFA layer, ZCA transform (specific type of WCT transform) and high frequency residual skip connections

that allowed for better feature reconstruction and preserved the information with less parameters than WCT^2[7] and kept blockwise training which would also help minimize

the feature reconstruction error.

4. Experimental setup

4.1 NST with Multi-Label Semantic Segmentation Weights

Initially I had a completely different approach and followed Gatys' [5] approach in "A Neural Algorithm of Artistic Style" by using layers of the existing CNN as representations of the image's style and content and then using gradient descent from a white noise image to optimize the loss function.

Let $\tilde{p}$ be the content image,$\tilde{a}$ be the style image and $\tilde{x}$ be the result image (output). The loss function to minimize is given by

$$L_{\text{total}}(\tilde{p}, \tilde{a}, \tilde{x}) = \alpha L_{\text{content}}(\tilde{p}, \tilde{x}) + \beta L_{\text{style}}(\tilde{a}, \tilde{x}) \tag{6}$$

where α and β are the weighting factors for content and style reconstruction, respectively.

Using VGG19 as my baseline model I applied Neural Style transfer to an interior room giving it a content and style image. The results showed distortions and noise in the image as this approach leads to a more "artistic" result and notably applied a very global style onto the image. I pivoted to trying to use pre segmented images however in this case not only were existing noise artifacts not fixed but some new artifacts were introduced as well. In some cases, the segmentation map was not precisely labeled leading to cases where errors in the segementation map would cause parts of the image to be improperly styled. One example of these kinds of artifacts is the loss of definition in the white fence, as seen in the Figure 7, where segmentation inaccuracies have led to visual discrepancies.

Figure 7: (a) Segmented style transfer (b) Segmentation map

Ultimately these downsides motivated me to pivot working on the autoencoder approach which would likely yield more photorealistic results and remove the potential problems of working with and needing segmentation maps.

4.2 Hyperparameters for the autoencoder method

The two main hyperparameters I tuned were the number of VGG layers I concatenated in my BFA, and the learning rate. I had three VGG layers I could concatenate so to test I ran three separate models appending different numbers of VGG layers and evaluated them against the image reconstruction loss and feature reconstruction loss terms defined in the original PhotoWCT^2 [4] paper. I also tried generating images for these different models to compare outputs.

Table 1: Hyperparameter Tuning: VGG Layer Concatenation and Learning Rate

Variation	Image Reconstruction Loss	Feature Reconstruction Loss	Total Loss
Concat Relu1-1, Relu2-1, Relu3-1	0.0013253256	0.17549543	0.1768207556
Concat Relu1-1, Relu2-1	0.0014241216	0.19465684	0.1960809616
Concat Relu1-1	0.0019849546	0.195485645	0.1974705946

Overall, all variations led to similar image outputs with only slight differences in lighting and shadows. I did find however that the image reconstruction loss and the feature reconstruction loss was the lowest for the variation that con-catted all three possible VGG layers. As a result, I chose this variation as the one I trained the final model on. After I determined which concatenation of layers led to the best results, I tuned the learning rate of the model by performing a log-scale random search to see which had the best loss convergence. I found that the learning rate used by the original PhotoWCT^2 [4] produced good results and that other variations did not lead to significantly faster convergence. Ultimately, I settled on a rate of 1e-4 to train my final model.

5. Results and Discussion

5.1 Image Outputs

Compared to WCT (baseline) [8] , my results were significantly more realistic. As expected, the introduction of skip connections to the decoder from PhotoWCT^{2,} s [4] approach and BFA from PhotoNet [4] gave the decoder significantly more information to work with over WCT (baseline) [8]and as a result, my method was able to reconstruct the image better and produce more photorealistic results. However compared to PhotoWCT^2 [4] , the approach that just added blockwise training with high frequency skip connections, i

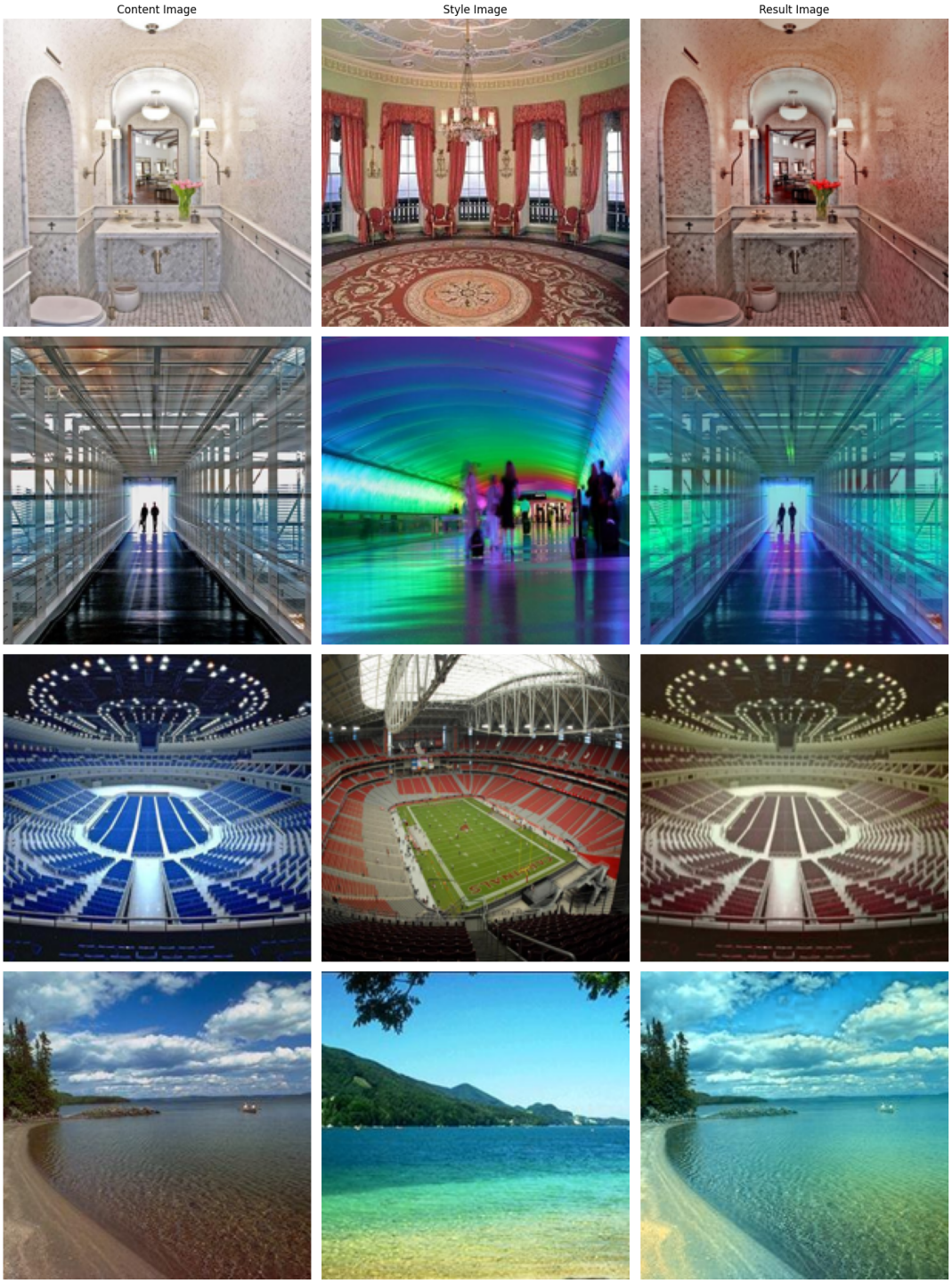

Figure 8: Results with three concatenated ReLu outputs in BFA layer (see appendix for more results and larger images)

found my approach was not significantly better. I suspect part of the reason why this was the case was because PhotoWCT^2 [4] already had skip connections from the VGG encoder to the decoder. While it was theorized that feature aggregation might still be able to provide a unique benefit because it provided the information at the bottleneck rather than sending it directly to specific layers, it seems that in practice adding BFA didn't significantly increase the amount of information that the decoder had when stylizing the image as the skip connections already provided similar information. There were

still some cases where my model was able to perform slightly better than PhotoWCT^2 [4]. Notably as seen in the appendix image of the indoor restaurants and wooden wall structure , my approach is sometimes more conservative in altering parts of the image and actually correctly doesn't choose to stylize parts of the image that PhotoWCT^2 [4] would. These cases are few and far between. my method overall though did not perform any worse than PhotoWCT^2 [4] in image output in the situations tested, showing that BFA is fully compatible with the blockwise training approach.

5.2 Metrics

I also compared my approach to PhotoWCT^2 [4]with two metrics used in the original PhotoWCT^2 [4] paper, image reconstruction loss and feature reconstruction loss. The reconstruction loss is defined as the pixelwise L2 distance between an input image, and the reconstructed image after it is run through the encoder and reconstructed with the decoder, $|||I - I_{\text{rec}}|||_2^2/(HWC)$ where H, W, C are height width and number of channels. The feature reconstruction loss is the L_2 loss between a feature F_n from a given layer $relu_{N-1}$ to the corresponding decoder block feature $F_{N,r}$. This is expressed as $|||F_n - F_{N,r}|||_2^2/|||F_n|||_2^2$. Overall, my losses don't deviate too much from PhotoWCT^2 [4] and at least empirically be looking at the model image outputs, there isn't as much of a difference. One reason that could explain the difference in losses is the training time. The PhotoWCT^2 [4] model didn't need to train additional parameters associated with my BFA layer and likely had more training time then I did for this report. It is fairly likely then that given more training time the difference in these metrics would shrink.

Metric	My model loss values
Image Reconstruction Loss	0.0014265842
Feature Reconstruction Loss	0.1695426

Table 2: Summary of Losses

6. Conclusion and Future Work

This paper demonstrates the compatibility of the BFA approach with blockwise training and the approach is partially successful at stylizing some components of image. Ultimately, I have demonstrated that adding BFA alone to PhotoWCT^2 [4] approach doesn't yield significant benefits, likely because the information given is redundant given the existence of skip connections. A major challenge for the project was the training time for the autoencoder architecture which I would have liked to train for more epochs in my experiments if given more computational resources. I also considered a GANs approach to this problem that I was unable to explore due to time constraints. However, this approach would have been more promising as at the core of it, I would like to be able to generate,

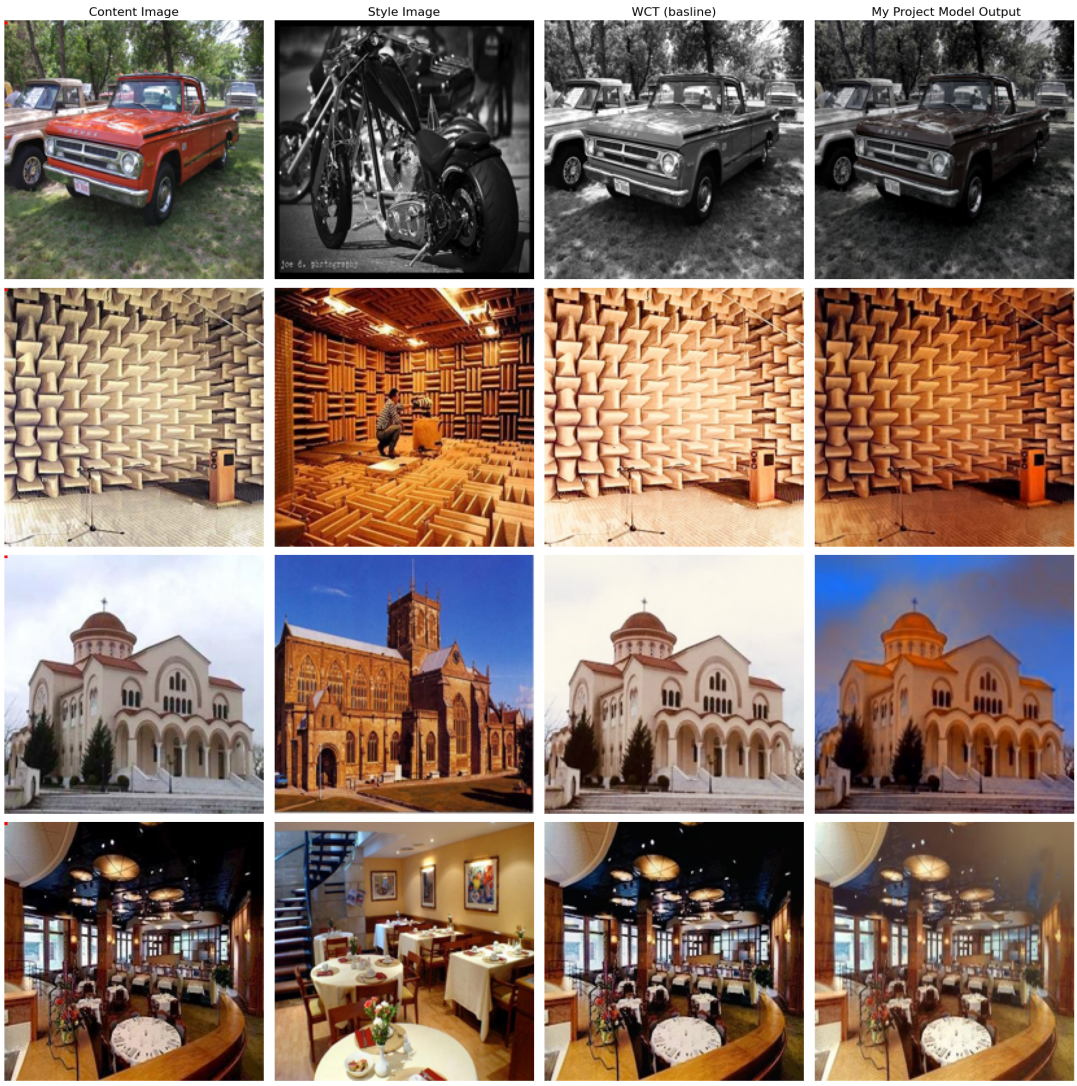

Figure 9: Results with baseline output and my project output (see appendix for larger images)

add, remove, or change existing image into the new style the user proposes. With style transfer this is not possible as the style I introduce cannot actually modify the structure of the underlying image it is being applied to.

7. Appendix

References

[1] *ADE20K Dataset*. https://groups.csail.mit.edu/vision/datasets/ADE20K/.

[2] Jie An et al. 'Ultrafast Photorealistic Style Transfer via Neural Architecture Search'. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (Apr. 2020), pp. 10443–10450. DOI: 10.1609/aaai.v34i07.6614.

[3] Tai-Yin Chiu. 'Understanding Generalized Whitening and Coloring Transform for Universal Style Transfer'. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.

[4] Tai-Yin Chiu & Danna Gurari. *PhotoWCT^2: Compact Autoencoder for Photorealistic Style Transfer Resulting from Blockwise Training and Skip Connections of High-Frequency Residuals*. 2021. arXiv: 2110.11995 [eess.IV].

[5] Leon A. Gatys, Alexander S. Ecker & Matthias Bethge. *A Neural Algorithm of Artistic Style*. 2015. arXiv: 1508.06576 [cs.CV].

[6] Yijun Li et al. *A Closed-form Solution to Photorealistic Image Stylization*. 2018. arXiv: 1802.06474 [cs.CV].

[7] Yijun Li et al. 'A Closed-form Solution to Photorealistic Image Stylization'. In: *ArXiv* abs/1802.06474 (2018). URL: https://api.semanticscholar.org/CorpusID:3499796.

[8] Yijun Li et al. *Universal Style Transfer via Feature Transforms*. 2017. arXiv: 1705.08086 [cs.CV].

[9] Jonathan Long, Evan Shelhamer & Trevor Darrell. *Fully Convolutional Networks for Semantic Segmentation*. 2015. arXiv: 1411.4038 [cs.CV].

[10] *MSCOCO Dataset*. https://cocodataset.org/.

[11] Olaf Ronneberger, Philipp Fischer & Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[12] Aliaksandr Siarohin, Enver Sangineto & Nicu Sebe. *Whitening and Coloring transform for GANs*. June 2018.

[13] Jaejun Yoo et al. *Photorealistic Style Transfer via Wavelet Transforms*. 2019. arXiv: 1903.09760 [cs.CV].

Figure 10: Results with WCT baseline output : Wooden wall

Figure 11: Results with my model output : Wooden wall

Figure 12: Results with WCT baseline output : Restaurant

Figure 13: Results with my model output : Restaurant

Figure 14: More results with my approach

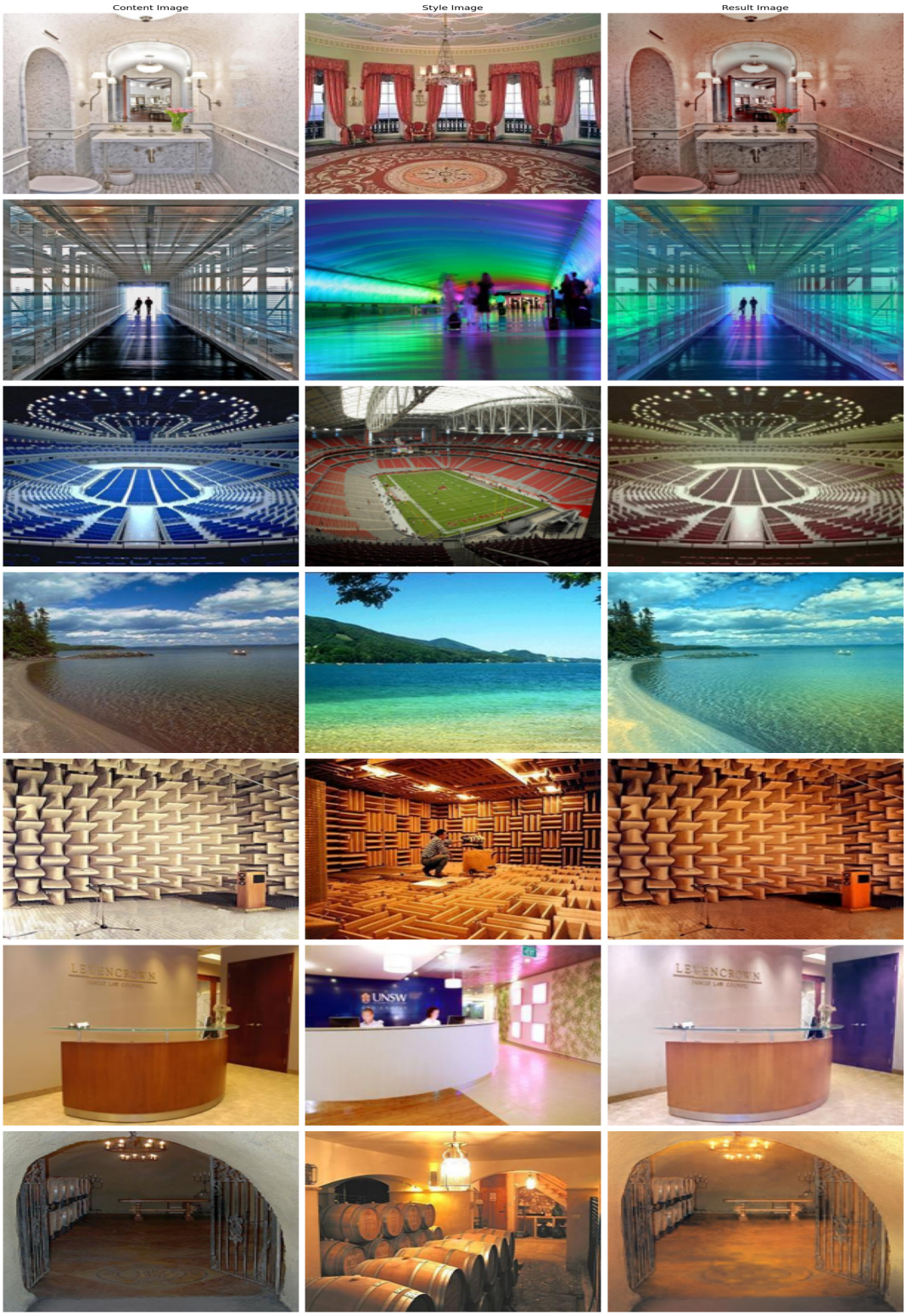

Figure 15: More results with my approach