Zero-Shot Semantic Neural Style Transfer for Images

Hrishikesh Pawar [1] **Tejas Rane** [1] **Harshal Bhat** [1] **Shounak Naik** [1]

Abstract

*In this paper, we address the computational problem of zero-shot semantic neural style transfer for images. Our approach is to implement the **Ada**ptive **Att**ention **N**ormalization (**AdaAttN**) mechanism, designed for zero-shot arbitrary neural style transfer. We explore the adaptability of AdaAttN in preserving semantic content while applying diverse styles to images. Our analyses, involving a range of experiments and user-study which identify the redundancy of the original paper improving the training time by 13%. Additionally, we not only achieve high-quality style transfer but also excel in maintaining the semantic integrity of the images. Our approach focuses on implementing the AdaAttN from scratch for semantic stylization of images and performing experiments and ablation studies.*

1. Introduction

Recent advancements in Zero-Shot Arbitrary Style Transfer have revolutionized image editing, enabling real-time application of diverse styles onto photos. This technology has moved from academic research to practical applications in image editing software[1], simplifying the creation of visually appealing images. However, while semantic style transfer is well-explored, existing methods often perform global transfer styles, neglecting local image features and causing unpleasing distortions. The need for localized attention in style transfer is critical, particularly in semantic applications. Adaptive Attention Normalization (AdaAttN) (Liu et al., 2021), with its focus on local features, presents a promising solution, and our paper explores its implementation in zero-shot semantic style transfer.

AdaAttN, a novel module for fast arbitrary neural style transfer, has garnered attention for its flexibility and broad applicability. Contrasting with conventional methods, which primarily focus on blending deep style features with content or aligning content features with overall style metrics, AdaAttN introduces a novel technique. It adjusts the normalization process, taking into account individual points. This per-point attentive normalization ensures a more refined and detailed alignment of the content features with the specific characteristics of the style features, which significantly reduces unnatural outputs and local distortions, making it an ideal candidate for our study.

Our project aims to implement the AdaAttN method, with a focus on training and development of corresponding utility functions around it. This paper also aims to integrate arbitrary image segmentation with style transfer. Traditional image segmentation methods are limited by their fixed set of object classes, requiring extensive retraining for new classes. Therefore, as an extension, we propose to integrate a text-based image segmentation method that can generate segmentation based on arbitrary prompts. This approach could lead to more personalized and scalable applications, enhancing the utility of style transfer across diverse use cases.

1.1. Research contributions

We aim to develop AdaAttN training and utility modules from scratch and investigate the effects of module reduction in AdaAttN, impact of local versus global loss, and the skewness in loss. These explorations are intended to provide insights into the flexibility, effectiveness and improvement of AdaAttN.

2. Related Work

The entire idea of Neural Style Transfer stems from the seminal work of Gatys et al. (Gatys et al., 2015) in 2016 of using neural networks for style-transfer. They used VGG network to extract style and content loss echoing two objectives of content and style similarity. They then use an iterative optimisation to find an output image that simultaneously minimise the two loss functions. This method is quite flexible and can transfer any given styles. However, it is slow due to the expensive iterative optimization and hence categorised as Slow and Arbitrary Style Transfer.

The second type of categorization Fast Arbitrary Style Transfer aims to enhance speed and flexibility. Johnson et al. (Johnson et al., 2016) introduced a method using feed-forward networks and perceptual loss functions for rapid

[1]Worcester Polytechnic Institute. Correspondence to: Tejas Rane <turane@wpi.edu>.

CS/DS 541 Deep Learning Final Project, Worcester Polytechnic Institute.

[1]NightCafe

style transfer. Although their implementation was fast, was limited to single-style models. Ulyanov et al. (Ulyanov et al., 2016) expanded this by introducing a generative feed-forward model, improving both efficiency and quality, and partially addressing the single-style limitation. Huang et al. (Huang & Belongie, 2017), marked a significant advancement by enabling real-time, arbitrary style transfer. It achieved this by aligning the mean and variance of content features with those of style features, yet occasionally at the expense of image quality. Park et al. (Park & Lee, 2018) further refined this approach by integrating local style patterns based on the spatial distribution of content images enabling decoder to preserve content structure while enriching style patterns, but still grappling with local distortions. These methods, however, often relied on deep CNN features, neglecting shallower aspects and leading to content structure distortion. This highlighted a need for a more balanced approach between transferring style patterns and preserving content structures, a key focus of our project.

Building on the advancements in Neural Style Transfer (NST), recent research has extended NST to localized regions within images. Castillo et al. (Castillo et al., 2017) introduced targeted style transfer using instance-aware semantic segmentation, combining Gatys et al.'s (Gatys et al., 2015) algorithm with instance segmentation. Their method, however, was slow. To address this, Lironne Kurzman et al. (Kurzman et al., 2019) developed the Class-Based Styling (CBS) method for real-time localized style transfer. CBS achieves real-time localized style transfer by simultaneously performing semantic segmentation and global style transfer and fusing the masks obtained from the segmentation network with that of the entire stylised image to achieve localised stylization. But it was limited to single styles. On similar lines, Alex Wells et al.[2] achieved semantic stylization using a blend of style transfer and MRFs for more natural integration. Our paper aims to enhance semantic arbitrary style transfer, leveraging AdaAttN (Liu et al., 2021) for more versatile and visually pleasing results.

3. Methodology

3.1. Overall Framework

The style transfer network takes a style image I_s and a content image I_c to synthesize a stylized image I_{cs}. A pre-trained VGG-19 network (Simonyan & Zisserman, 2014) is used as an encoder to extract multi-scale feature maps. The decoder has a symmetric structure of VGG-19. A multi-level strategy is employed to take full advantage of features in both shallow and deep layers, by integrating three AdaAttN modules on ReLU-3_1 , ReLU-4_1 and ReLU-5_1 layers of VGG, respectively. The extracted feature of layer ReLU-x_1 in VGG is denoted as F^x_{cs}, when it takes an

[2]Localized Style Transfer

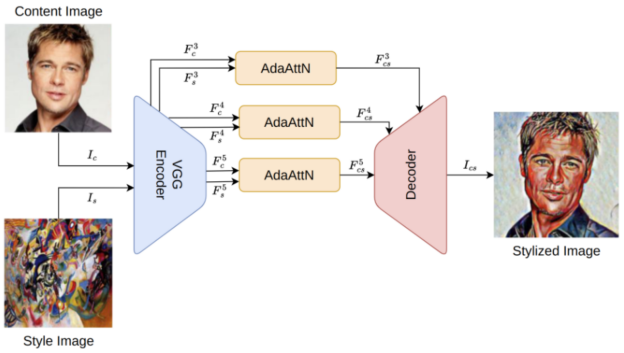

Figure 1. Overall framework of the style transfer network. The content image I_c and the style image I_s are taken as input to synthesize a stylized image I_{cs}.

image I_* as input. $*$ can be c or s, representing content and style features respectively. To fully exploit low-level patterns, the features of current layer are further concatenated with the down-sampled features of its previous layers as:

$$F^{1:x}_* = D_x(F^1_*) \oplus D_x(F^2_*) \oplus \cdots \oplus F^x_* \tag{1}$$

where D_x stands for the bilinear interpolation layer which downsamples the input feature to the same shape of F^x_*, and $\oplus$ here means concatenation operation along channel dimension. Then, the embedded feature of the AdaAttN module at layer l can be denoted as:

$$F^x_{cs} = AdaAttN(F^x_c, F^x_s, F^{1:x}_c, F^{1:x}_s) \tag{2}$$

where F_c, F_s and F_{cs} are content, style, and embedded feature, respectively. With multi-level embedded features, the stylized image I_{cs} is synthesized with decoder as:

$$I_{cs} = Dec(F^3_{cs}, F^4_{cs}, F^5_{cs}) \tag{3}$$

3.2. Adaptive Attention Normalization

The pivotal element in arbitrary style transfer models is the feature transformation module. The Adaptive Attention Normalization (AdaAttN) module plays a crucial role in dynamically adjusting the feature distribution on a per-point basis by incorporating both low-level and high-level features through an attention mechanism. Illustrated in Figure fig, AdaAttN operates in three stages: **(1)** generating an attention map using content and style features across shallow to deep layers; **(2)** computing weighted mean and standard variance maps for the style feature; **(3)** adaptively normalizing the content feature to achieve per-point feature distribution alignment.

Attention Map Generation. AdaAttN uses an attention mechanism to measure the similarity between the low-level and high-level features of the content and the style images

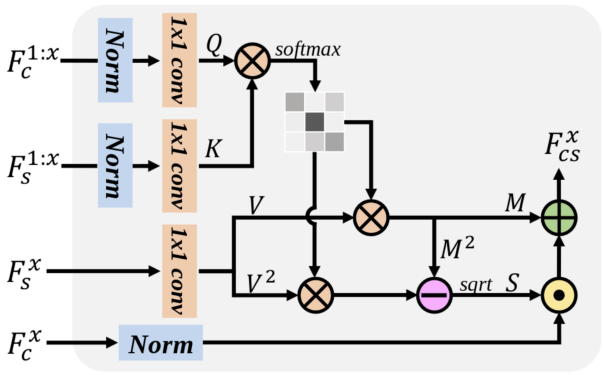

Figure 2. The structure of the AdaAttN module. The module takes four inputs: concatenated content features $F_c^{1:x}$, concatenated style features $F_s^{1:x}$, style features F_s^x, and the content features F_c^x. The output is the embedded feature F_{cs}^x.

simultaneously. To compute the attention map A of the layer x, the Queries Q, Keys K, and Values V are formulated as:

$$\begin{aligned} Q &= f(Norm(F_c^{1:x})), \\ K &= g(Norm(F_s^{1:x})), \\ V &= h(F_s^x), \end{aligned} \tag{4}$$

where, f, g, and h are 1×1 learnable convolution layers, $Norm$ here denotes channel-wise mean-variance normalization, as used in instance normalization. The attention map A can be calculated as:

$$A = softmax(Q^T \otimes K) \tag{5}$$

where, $\otimes$ denotes matrix multiplication.

Weighted Mean and Standard Variance Map. AdaAttN applies attention score matrix A to the style features F_s^x by calculating the attention-weighted mean M and attention-weighted standard variance S as:

$$\begin{aligned} M &= V \otimes A^T, \\ S &= \sqrt{(V \cdot V) \otimes A^T - M \cdot M} \end{aligned} \tag{6}$$

where, $\otimes$ denotes matrix multiplication and $\cdot$ denotes element-wise multiplication.

Adaptive Normalization. AdaAttN transfers feature statistics by generating attention-weighted mean and standardnvariance maps. For each position and each channel of normalized content feature map, corresponding scale in S and shift in M are used to generate transformed feature map:

$$F_{cs}^x = S \cdot Norm(F_c^x) + M \tag{7}$$

3.3. Loss Functions

The overall loss function is defined as the weighted summation of the global style loss ($\mathcal{L}_{gs}$) and the local feature loss ($\mathcal{L}_{lf}$):

$$\mathcal{L} = \lambda_g \mathcal{L}_{gs} + \lambda_l \mathcal{L}_{lf} \tag{8}$$

where, λ_g and λ_l are hyper-parameters controlling weights of their corresponding loss terms.

The global style loss ($\mathcal{L}_{gs}$) is calculated as the distance between the mean μ and standard deviation σ between the generated stylized image and the style image in the VGG feature space:

$$\begin{aligned} \mathcal{L}_{gs} = \sum_{x=1}^{5} (&||\mu(E^*(I_{cs})) - \mu(F_s^x)||_2 \\ &+ ||\sigma(E^*(I_{cs})) - \sigma(F_s^x)||_2) \end{aligned} \tag{9}$$

where, $E()$ denotes the feature of the encoder and the superscript ^{x} denotes the layer index.

The novel local feature loss, proposed by AdaAttN, constraints that the feature map of the stylized image is consistent with the transformation result by the $AdaAttN$ module:

$$\mathcal{L}_{lf} = \sum_{x=1}^{5} ||E^*(I_{cs}) - AdaAttN^*(F_c^x, F_s^x, F_c^{1:x}, F_s^{1:x})||_2 \tag{10}$$

where, $AdaAttN^*$ serves as the supervision signal. Local feature loss makes the model generate better stylized outputs for local areas compared with the conventional content loss term.

4. Experiments

4.1. Implementation Details

We train our arbitrary style transfer network using the PhraseCut dataset (Wu et al., 2020) as our content image set and WikiArt dataset (Phillips & Mackintosh, 2011) as our style image set. The loss hyper-parameters λ_g and λ_l were set to 1. Adam (Kingma & Ba, 2017) optimizer with its default parameters was used as the solver. During the training phase, all the images where resized to 512×512 resolution for training. The network was trained for around 55k iterations on a single NVIDIA A100 GPU (Turing Cluster), with a batch size of 4 images. The total training time required was around 11 hours.

4.2. Ablation Study

4.2.1. Reduction in Modules

In the original paper (Liu et al., 2021), the authors have removed the shallow module of AdaAttN and showed that the

visual results are displeasing when we remove the shallowest AdaAttN module. We wanted to take this thought further and wanted to study the effect of the deeper AdaAttN modules. We removed these modules one at a time and trained our model for 30 epochs.

4.2.2. Local vs Global Loss

The total loss is described in Equation 8. We have studied how the local feature loss and global style loss formulataion affect the visual results. We have varied the λ_g and λ_l from 0 to 1 with an interval of 0.25. We train models for 30 epochs for each of these configurations and compare the visual results against each other.

4.2.3. Skewness in Loss term

The global loss is responsible for "potraitizing" the image, it gives the global style to the content image. It does this by trying to match the distribution of the style image and the stylized image. The original paper assumes that matching μ and σ would be enough. We think of a case where this would not be enough. We carefully construct a Gamma and Normal distribution with the same μ and σ. These distributions are visualized in Figure 6. Thus we introduce the skewness factor into our global loss function. Sknewness is defined as the asymmetry of a distribution. The addition of this term would encourage the model to fit the style distribution more uniquely to the content image.

5. Results

In this section, we discuss the overall results and the conclusions of the training and ablation studies that were performed. We see that our loss curve was converging after around 40 epochs as seen in Figure 5. We saw that the style is infused better as we approach convergence during training. We see that the global style is quick to be adapted onto the content image but as the training progresses, local features get better style infusion.

5.1. Reduction in Modules

We removed the 4_1 attention module as seen in Figure 4. We then removed the 5_1 module. We saw very similar looking results no matter the deletion in attention modules. As a standard practice in Neural Style transfer papers, we floated a user study to ask users if they felt the results from different configurations look the same or not. The user study indicated that the users also felt the reduction in attention module does not make a difference in the visual results. The user study for this part is summarized in **??**. We further observed that the complete model needs 128 seconds to train one epoch and by removing the 4_1 attention module, the model needs 115 seconds to train the same epoch. This means we would achieve a 13% latency improvement by removing the 4_1 attention module. The visual results are reported in the appendix.

5.2. Global Loss , Local Loss

Changing the proportion of the global loss and local loss leads to significant visual changes. Without the global loss, we see the image not capturing the essence of the style. But without the local loss, we found out the local features don't get the style features and the style is imbibed onto the image in a crude way. The user study led us to conclude that most of the users prefer a blend of local and global loss. The user study for this part is summarized in Figure **??**. The visual results are reported in the appendix.

5.3. Skew Factor

Although, we thought the skewness factor would be a good addition theoretically, we got very similar looking results. This might mean the style features are already in a Gaussian distribution and thus the skewness would anyway be 0. The visual results are reported in the appendix.

6. Discussion

Our study on zero-shot semantic neural style transfer using AdaAttN method has yielded significant insights and contributions to the field of image processing and neural style transfer with one key limitation which lies in the computational resources required for training and implementing the AdaAttN model. While our modifications and ablation studies have made strides in reducing training time and resource requirements (as evidenced by a 13% reduction in training time with the removal of certain AdaAttN modules), the approach still demands substantial computational power, particularly when dealing with high-resolution images or complex styles. This can limit the accessibility of our method for users with limited computational resources. Our work also opens up new possibilities for future exploration. The integration of text-based image segmentation methods, such as CLIPSeg, offers a more personalized and scalable approach to semantic style transfer.

7. Conclusions and Future Work

Our research demonstrates that zero-shot semantic neural style transfer can be effectively achieved using the AdaAttN model. This model allows for a dynamic and attentive blending of style and content features, resulting in visually compelling images that were validated via user study. Future work in this area should focus on improving the efficiency and accessibility of the model. Developing a more resource-efficient version of AdaAttN could significantly enhance its applicability. Additionally, integrating advanced segmentation techniques like CLIPSeg promises to add a new dimension to style transfer, allowing for more personalized and context-aware applications.

References

Castillo, Carlos Domingo, De, Soham, Han, Xintong, Singh, Bharat, Yadav, Abhay Kumar, and Goldstein, Tom. Son of zorn's lemma: Targeted style transfer using instance-aware semantic segmentation. *CoRR*, abs/1701.02357, 2017. URL http://arxiv.org/abs/1701.02357.

Gatys, Leon A., Ecker, Alexander S., and Bethge, Matthias. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. URL http://arxiv.org/abs/1508.06576.

Huang, Xun and Belongie, Serge J. Arbitrary style transfer in real-time with adaptive instance normalization. *CoRR*, abs/1703.06868, 2017. URL http://arxiv.org/abs/1703.06868.

Johnson, Justin, Alahi, Alexandre, and Fei-Fei, Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016. URL http://arxiv.org/abs/1603.08155.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization, 2017.

Kurzman, Lironne, Vázquez, David, and Laradji, Issam H. Class-based styling: Real-time localized style transfer with semantic segmentation. *CoRR*, abs/1908.11525, 2019. URL http://arxiv.org/abs/1908.11525.

Liu, Songhua, Lin, Tianwei, He, Dongliang, Li, Fu, Wang, Meiling, Li, Xin, Sun, Zhengxing, Li, Qian, and Ding, Errui. Adaattn: Revisit attention mechanism in arbitrary neural style transfer, 2021.

Park, Dae Young and Lee, Kwang Hee. Arbitrary style transfer with style-attentional networks. *CoRR*, abs/1812.02342, 2018. URL http://arxiv.org/abs/1812.02342.

Phillips, Fred and Mackintosh, Brandy. Wiki Art Gallery, Inc.: A Case for Critical Thinking. *Issues in Accounting Education*, 26(3):593–608, 08 2011.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

Ulyanov, Dmitry, Vedaldi, Andrea, and Lempitsky, Victor S. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. URL http://arxiv.org/abs/1607.08022.

Wu, Chenyun, Lin, Zhe, Cohen, Scott, Bui, Trung, and Maji, Subhransu. Phrasecut: Language-based image segmentation in the wild, 2020.

8. Appendix

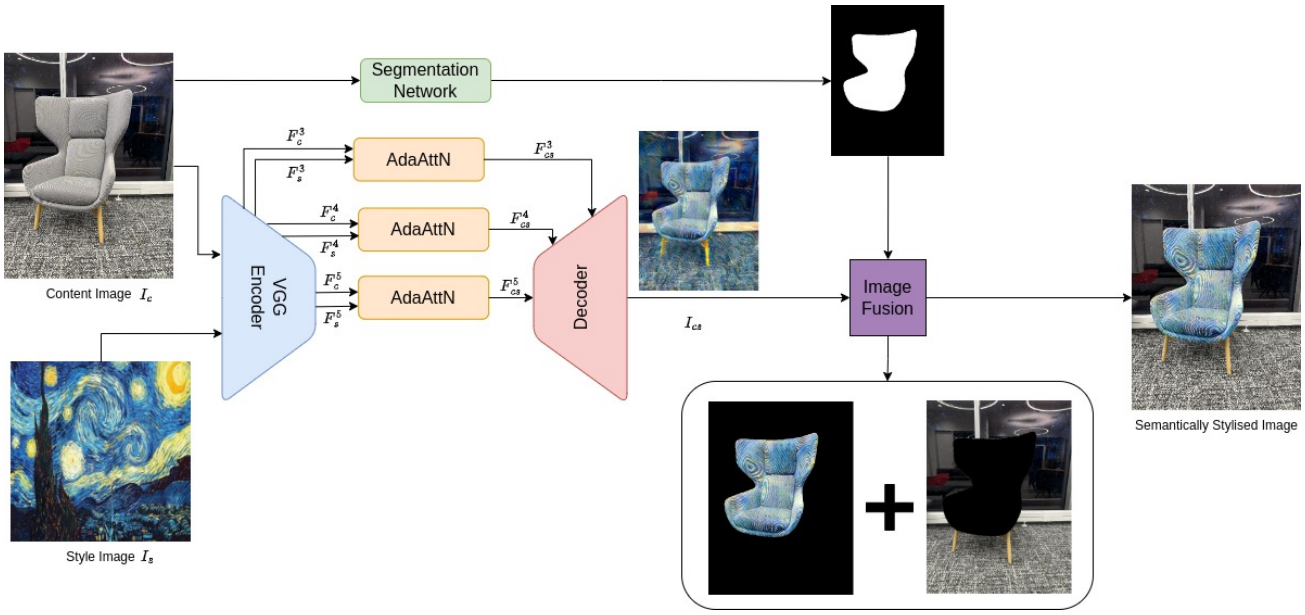

Figure 3. Overview of the Style Transfer Process: Pipeline for semantic style transfer, combining a content image (a chair in a room) with a style image (Van Gogh's 'The Starry Night'). The process begins with the content image being passed through an encoder to extract feature maps at multiple levels, which are then adaptively altered using attention mechanisms (AdaAttN) in correspondence with the style features derived from the style image. A segmentation network provides a mask that guides the style application. A decoder then reconstructs the modified feature maps to produce a stylized image, which is fused with the original content image's spatial structure to yield a semantically stylized image that maintains the integrity of the original content while adopting the artistic style of the style image.

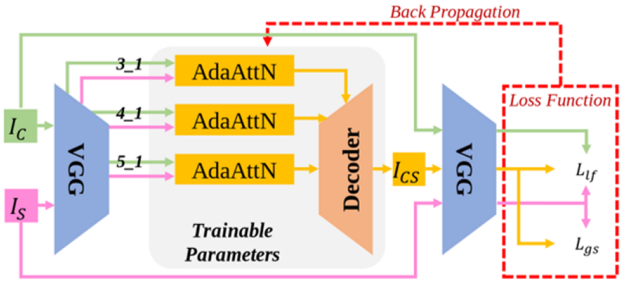

Figure 4. Overview of the full framework. The content image I_c, style image I_s, and the generated stylized image I_{cs} pass again through the VGG encoder, and the global style loss ($\mathcal{L}_{gs}$) and local feature loss ($\mathcal{L}_{lf}$) are calculated in the VGG encoder feature space. The three AdaAttN modules and the decoder are trainable, while the VGG encoder is pre-trained on ImageNet.

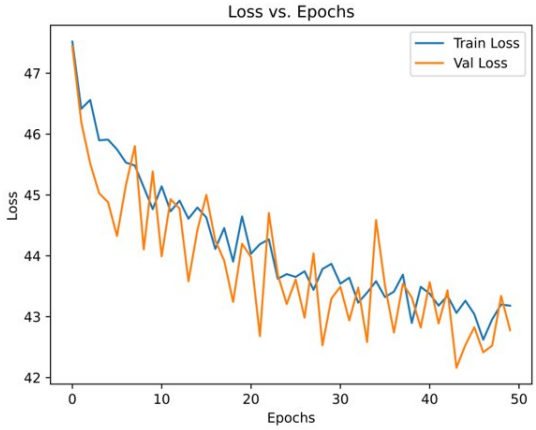

Figure 5. The loss curve on the complete training run.

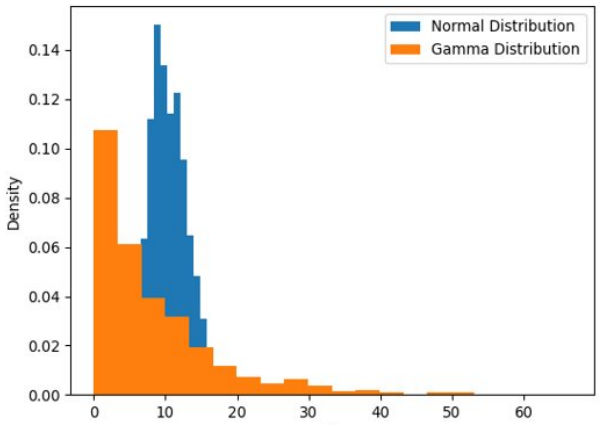

Figure 6. A gamma and normal distribution with the same μ and σ but with a different skew.

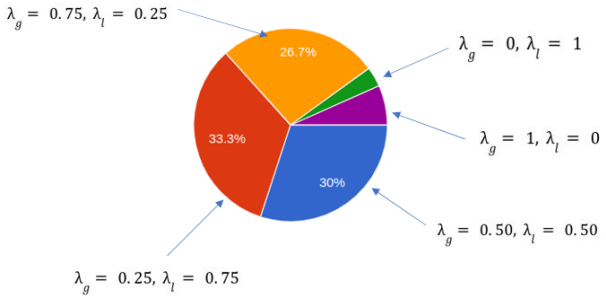

Figure 7. User Survey for preference between global and local loss

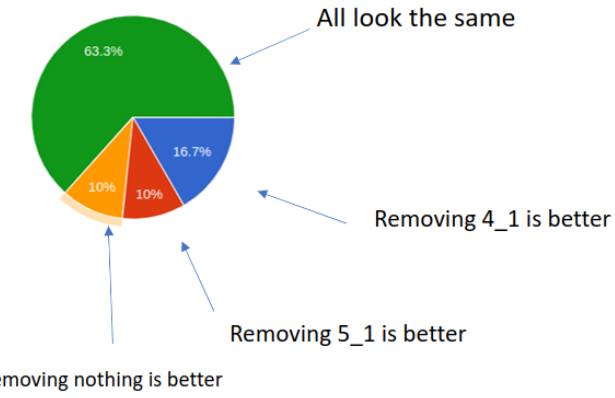

Figure 8. User Survey for removing attention modules

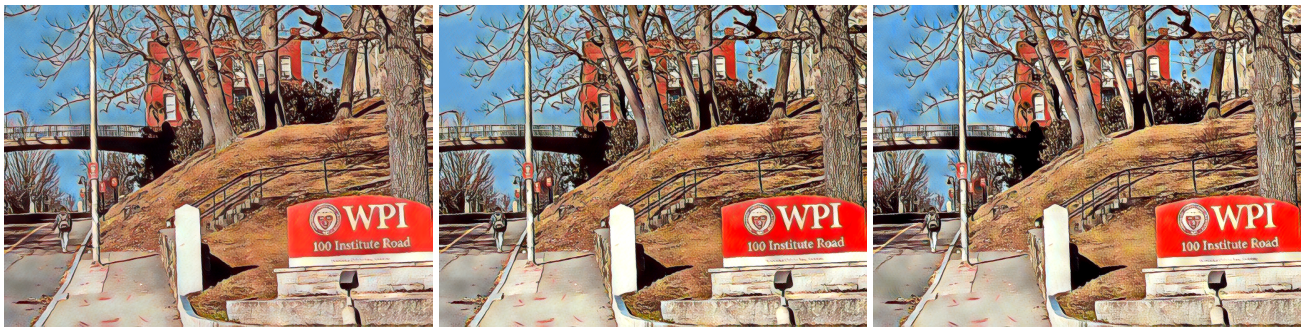

Figure 9. Results for section 8.1. The first figure is when all the attention modules are active. The second one is when attention module 4_1 is turned off and the third one is when 5_1 is turned off.

Figure 10. Results for section 8.2. The first 2 figures to the left are the content and the style image respectively. The rest of the five figures are as follows: 1) $\lambda_g = 1$, $\lambda_s = 0$, 2) $\lambda_g = 0.75$, $\lambda_s = 0.25$, 3) $\lambda_g = 0.5$, $\lambda_s = 0.5$, 4) $\lambda_g = 0.25$, $\lambda_s = 0.75$, 5) $\lambda_g = 0$, $\lambda_s = 1$,

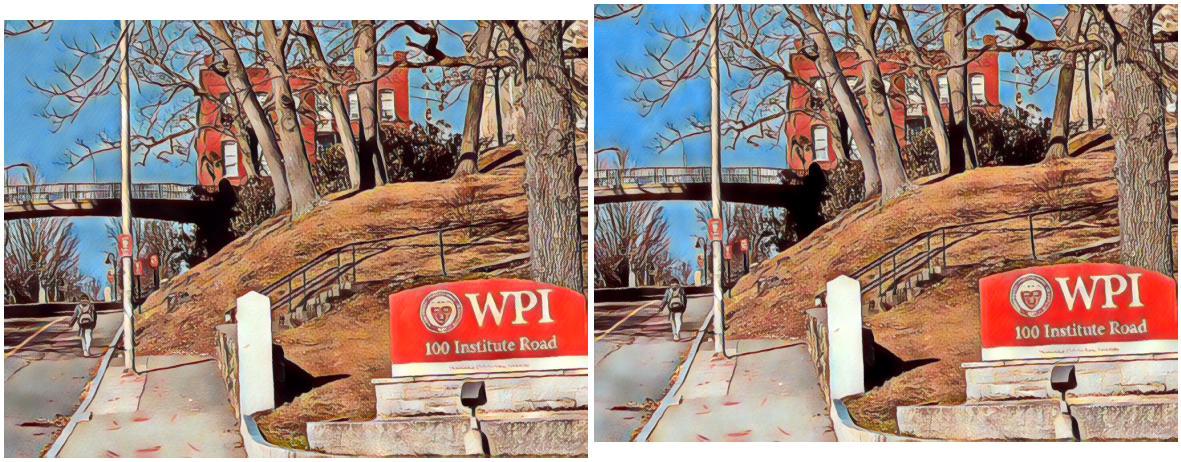

Figure 11. Results for section 8.3. The first figure is when the loss function is without the skew term in it whereas the second has a loss term in it.

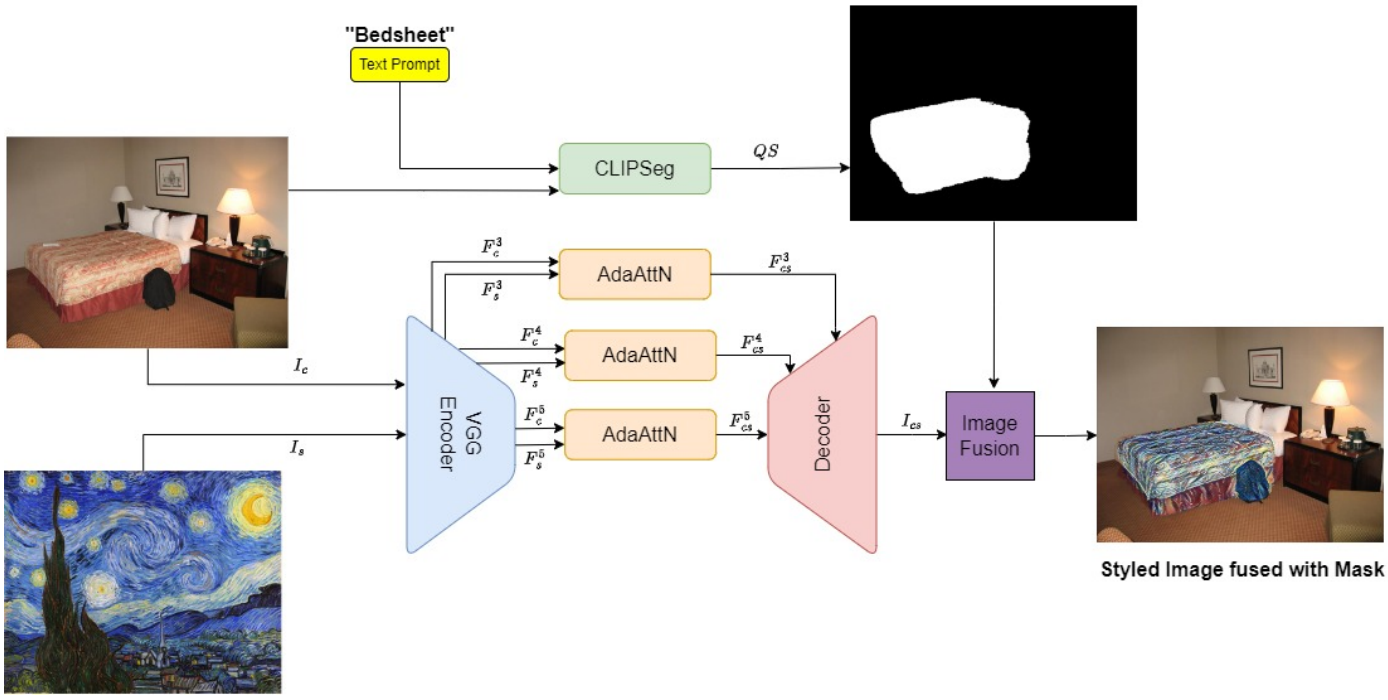

Figure 12. This figure demonstrates the pipeline of a zero-shot semantic neural style transfer algorithm, showcasing the transformation of a bedroom's bedsheet to adopt the style of Van Gogh's 'The Starry Night,' driven by a text prompt and utilizing advanced neural network techniques for image segmentation and style application.