Text Based Image Style Transfer

Project report submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Computer Science and Communication Engineering
Computer Science and Engineering

by

Varad Bane - Roll No. 21ucc111
Anushka Jain - Roll No. 21ucc022
Harsha Rani - Roll No. 21ucs088
Elishben Baraiya - Roll No. 21ucs077

Under Guidance of
Dr. Preety Singh

Department of Computer Science and Engineering
The LNM Institute of Information Technology, Jaipur

November 2024

The LNM Institute of Information Technology
Jaipur, India

CERTIFICATE

This is to certify that the project entitled "Text Based Image Style Transfer" , submitted by Varad Bane (Roll no 21ucc111), Anushka Jain (Roll no 21ucc022), Harsha Rani (Roll no 21ucs088) and Elishben Baraiya (Roll no 21ucs077) in partial fulfillment of the requirement of degree in Bachelor of Technology (B. Tech), is a bonafide record of work carried out by them at the Department of Computer Science and Engineering, The LNM Institute of Information Technology, Jaipur, (Rajasthan) India, during the academic session 2024-2025 under my supervision and guidance and the same has not been submitted elsewhere for award of any other degree. In my/our opinion, this report is of standard required for the award of the degree of Bachelor of Technology (B. Tech).

Date

Adviser: Name of BTP Supervisor

Acknowledgments

We would like to express our gratitude to the Almighty God for providing us with the strength, perseverance and clarity in completing the project successfully.

We extend our sincere gratitude to Prof. Preety Singh for her invaluable guidance and support, brainstorming and helping us devise potential solutions to challenges throughout the project. Her emphasis on understanding concepts thoroughly and the importance of visualizations and simplications for clarity of thoughts deeply enhanced our learning.

We are immensely thankful to Prof. Indra Deep Mastan for giving us the opportunity and inspiration to work on this project. His guidance during the initial phases in the 6th semester, shaped our understanding and approach towards the topic. We would like to acknowledge and appreciate the insights from Chanda Grover Ma'am, a PhD scholar at Ashoka University for the concept of contextual loss.

Finally, we would like to thank each group member for their dedication, collaboration and hard work, making the project a rewarding experience.

Abstract

Style transfer is a transformative technique that combines the content of one image with the artistic style of another, enabling the creation of captivating visuals. It has real-world applications in making art accessible to everyone, creating eye-catching visuals for marketing campaigns to boost brand engagement, and enriching entertainment and gaming experiences through immersive storytelling by applying creative styles to graphics.

The evolution of style transfer began with Neural Style Transfer (NST) which introduced a deep neural network-based approach to separate and recombine the content and style of arbitrary images, achieving artistic images of high perceptual quality. Over time, significant advancements were made in image style transfer, leading to innovative methods like ClipStyler which introduced text-based image style transfer (TIST). Unlike traditional methods requiring reference style images, TIST enabled users to define styles through textual descriptions, broadening its applicability. Further progress in this domain led to models such as MMIST which refined TIST techniques to provide greater flexibility and usability in scenarios where reference images are unavailable.

Despite these advancements, a notable limitation in current TIST methods is the lack of distinction between the foreground and background in stylized outputs. This issue arises from suppressed edges of the content image, resulting in less refined stylized images. To address this challenge, we implemented Contextual Loss, a loss function designed to compare regions with similar semantic meaning while considering the context of the entire image. By leveraging this loss function, we developed an optimal approach to integrate the right amount of content image features into stylized images, ensuring refined edges and enhanced perceptual quality.

Our work offers an effective solution for improving the clarity and visual appeal of stylized images, contributing to the ongoing advancements in style transfer technologies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Area of Work

The area of work focuses on style transfer, a transformative technique in computer vision and deep learning which enables the modification of an image's appearance by applying one image's artistic characteristics (referred as "style") to another image ("content image"). It preserves the structure of content image while seamlessly integrating stylistic features to it. Traditional style transfer techniques have proven effective but are often constrained by the requirement of a suitable reference style image.
Text-based image style transfer (TIST) extends the capabilities of style transfer by allowing users to define styles through natural language descriptions, such as "a watercolor effect" or "an abstract cubist painting." By leveraging the expressive power of language, TIST removes the dependency on reference images, providing unparalleled flexibility and accessibility. This innovation not only expands creative possibilities but also democratizes style transfer technology, making it accessible to users without technical expertise. The core technologies and methodologies underlying text-based image style transfer include the synergistic use of deep learning and computer vision. These fields work together to enable the system to process both visual and textual inputs effectively and achieve high-quality stylistic transformations.

Deep Learning

Deep learning, a subset of machine learning (ML), is pivotal in text-based image style transfer. It uses artificial neural networks to learn hierarchical representations from data, enabling the system to:

- Interpret Textual Descriptions: Deep learning models can be used to establish relationships between images and texts by learning to represent them in a common space based on their learned attributes, thus enabling the interpretation of user given texts to corresponding images and styles.

- Generate Stylized Outputs: Generate Stylized Outputs: Deep learning models synthesize these attributes into transformations that modify the input image.

By leveraging hierarchical learning, deep learning models identify patterns and features at multiple levels of abstraction, from basic edges and textures to complex stylistic elements, ensuring the output aligns with the textual style description.

Computer Vision

Computer vision complements deep learning by providing the analytical foundation for understanding and manipulating images. The integration of computer vision with deep learning is critical for style transfer. Computer vision provides the essential image analysis and manipulation capabilities, while deep learning enhances these processes by enabling sophisticated generative transformations. Together, they create a cohesive system that interprets textual descriptions and applies them to images with precision and creativity.

This process has become popular in creative fields such as digital art, photography, and media, allowing people to create unique visual effects and artistic interpretations of everyday photos and scenes.

1.2 Problem Addressed

Text-based image style transfer (TIST) addresses the critical limitations of traditional style transfer methods by enabling users to describe any imaginary style through textual input eliminating the need for a reference style image [2]. Thus, TIST offered greater flexibility than Image-based image style transfer (IIST).

Artifacts Generation:

- The existing TIST techniques mainly used the large vision-language models like CLIP [3][2]. However, directly utilizing CLIP for style guidance often results in undesirable artifacts, such as the inclusion of unrelated visual entities or text fragments in generated images. This was mainly due to the image text entanglement in the latent space of these vision-language models.
- Many models came up addressing this issue of artifacts [4][5]. One such state-of-the-art method is MultiModality-guided Image Style Transfer (MMIST) [4]. MMIST allows style inputs from multiple sources and modalities and generating multiple style representations using GAN inversion technique. This not only resolves the issue of artifacts but also provides the added advantage of accommodating diverse stylistic inputs.

Lack of Foreground-Background seperation:

- Focusing on the TIST component of MMIST, the stylized images it generates often lack proper foreground and background separation. This limitation arises due to the suppression or blurring of edges from the content image, which leads to a loss of structural perception in the final stylized outputs.

By addressing this issue, our method aims to achieve improved structural preservation in the stylized images, ensuring a more coherent separation between the foreground and background while maintaining the creative essence of the text-specified style.

1.3 Existing System

1.3.1 Traditional Image-Based Style Transfer:

Traditional Image-Based Style Transfer (IIST) is a computer vision technique to transform an image such that it captures the style from one image and content from another.

How Image-Based Style Transfer Works:
The main idea of style transfer is to separate the content of an image from its style. This is usually done using Convolutional Neural Networks (CNNs), which can identify and extract different features from an image.

1. Content Features: Represent the structural information in the image, such as objects, shapes, and layout.
2. Style Features: Capture the artistic attributes like texture, color palette, and patterns of the reference style image.

The seminal work on Neural Style Transfer [6] introduced a methodology leveraging a pre-trained VGG network to extract style and content features from specific layers. The stylized image is then generated by optimizing pixel values to minimize the Mean Squared Error (MSE) between the features of the generated image, content image, and style image.
This pioneering approach utilized pixel optimization guided by a loss function balancing content and style reconstruction.

Evolution of IIST: While this method laid the foundation for style transfer, numerous subsequent advancements have been made, introducing:

- CNN-based improvements for more efficient feature extraction and style application.
- Transformer-based models enabling better generalization and scalability.

- Arbitrary style transfer techniques, allowing the application of unseen styles without retraining.

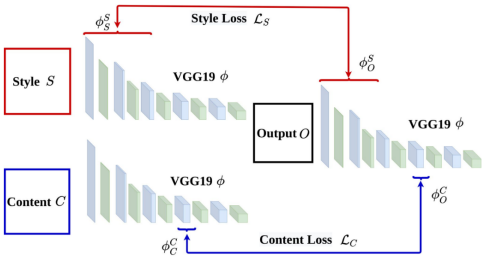

FIGURE 1.1: Neural style transfer by Gatys. et. al

1.3.2 Text-Based Image Style Transfer (TIST):

Text-Based Image Style Transfer (TIST) allows users to provide styles using textual descriptions rather than reference images. This eliminates the need for a style reference thus enabling greater flexibility and creativity in style application.

How Text-Based Image Style Transfer Works:
TIST leverages vision-language models to generate stylized images based on text descriptions. These models encode both textual and visual representations/embeddings into a common space, facilitating the association of texts with relevant images.
Vision-Language Models: Vision-language models like CLIP (Contrastive Language–Image Pretraining) play a pivotal role in TIST [3]. CLIP comprises two encoders:

1. Text Encoder: Converts textual descriptions into embeddings.
2. Image Encoder: Maps images into the same embedding space.

By maximizing the similarity between aligned image-text pairs and minimizing it for mismatched pairs during training, CLIP learns a shared latent space for textual and visual representations.

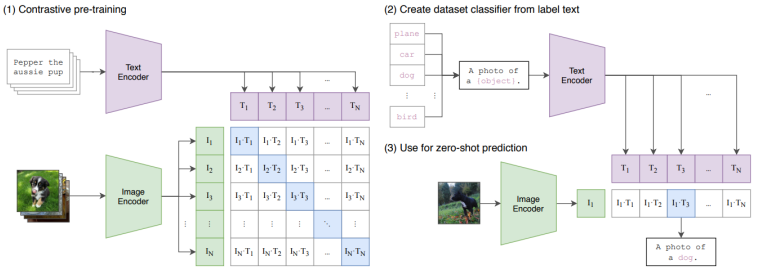

FIGURE 1.2: CLIP's Approach

One such TIST method which used CLIP's embedding space to achieve style transfer is CLIP-Styler [2]. It basically aligned the difference between textual description of style with content text "A Photo" and Stylized image and content image using Patch wise directional loss. This was successful in transferring global as well as finer style details to the content images.

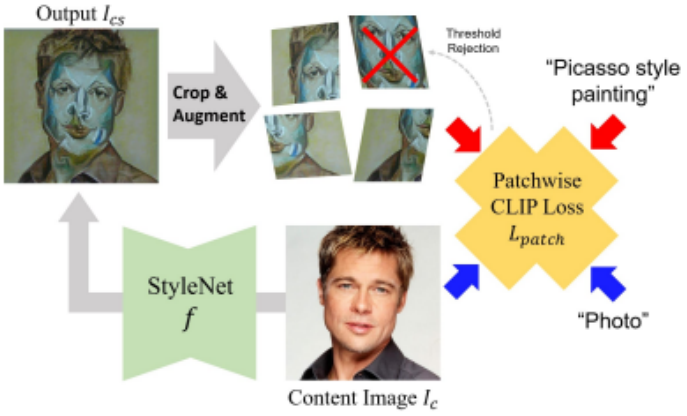

FIGURE 1.3: Overall schematics of patch-wise CLIP loss in CLIPStyler

Many other methods were also developed to enhance the results of TIST and even that of CLIPStyler, mainly with the aim of eliminating its limitations of visual and textual artifacts due to entanglement in the embedding space of the vision language models.

1.3.3 Multimodality-Based Image Style Transfer (MMIST)

Multimodality-Based image style transfer effectively addresses the limitations of prior method by reducing the undesirable visual and textual artifacts in the stylized images. It also offers a greater flexibility in style transfer by incorporating multiple style inputs from diverse modalities.
MMIST is a novel framework designed for image style transfer that leverages style guidance from multiple modalities [4]. It builds upon traditional Image-guided Image Style Transfer (IIST) and Text-guided Image Style Transfer (TIST) to synthesize stylized images that maintain the integrity of the content while achieving the desired artistic style. This method is particularly suitable when styles are not easily represented through a single reference.

How MMIST Works:
The MMIST framework uses a cross-modal GAN inversion approach to map input styles from various modalities (e.g., text descriptions, reference images) into a latent space of a pretrained GAN (such as StyleGAN).This latent space vectors then generate final style representations which are used as styles for IIST based model used for style transfer. Here's an outline of its process:

1. Cross-modal Style Representation: Style inputs from text and images are encoded into a shared latent space, producing intermediate style representations. This ensures that styles are disentangled from content and combined meaningfully.

2. Adapted Style Transfer Model: These style representations are then used with an adapted IIST model, which synthesizes stylized outputs while preserving the original content's structural details.

3. Boosting Stylization Quality: A multi-style boosting mechanism enhances style representations.In this approach multiple style representations are generated for each style combination given by user.This ensures that finer details and patterns are accurately captured.

By separating style generation from the final style transfer process, MMIST ensures higher quality and more consistent results. The architecture of MMIST allows seamless integration of multiple styles, facilitating cross-modal style interpolation and flexible stylization.
The figure below shows the architecture of MMIST method to visualize it's working.

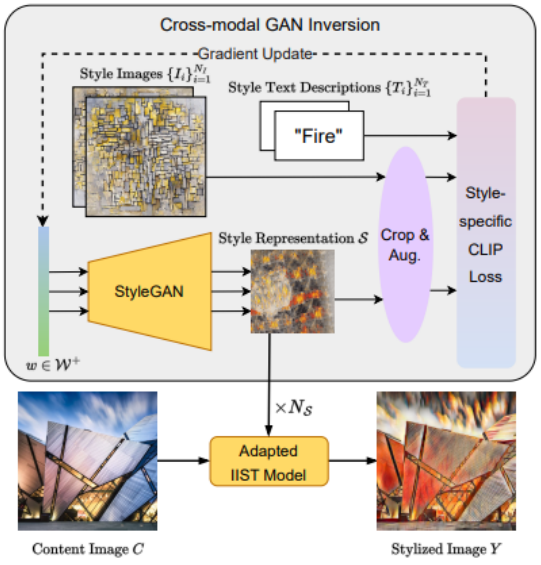

FIGURE 1.4: Overview of MMIST Method

This innovative approach achieves state-of-the-art results in both TIST and MMIST tasks, as demonstrated by extensive experiments and user evaluations in [4].

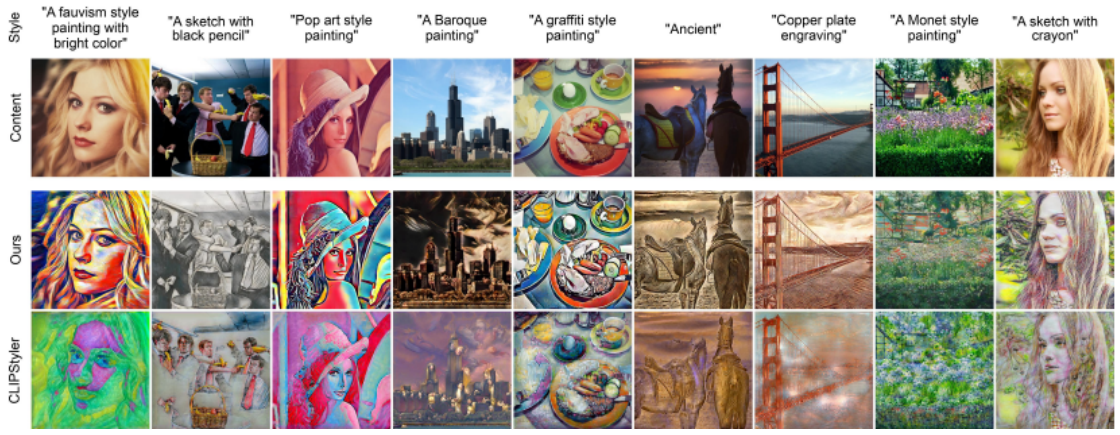

FIGURE 1.5: TIST results by MMIST and CLIPStyler

Chapter 2

Literature Review

2.1 Introduction

Style transfer is an application of computer vision and deep learning which allows us to combine the content of one image with the artistic style of another. It has numerous applications in real world like creation of digital art, enhancing marketing visuals to boost brand engagement and enriching the entertainment experiences by applying artistic styles to graphics and scenes [7][8].

Deep Learning underpins style transfer by enabling separation of content and style features of an image using various types of neural networks. Convolutional Neural Networks (CNNs) and vision-language models are key tools in this process.

Style transfer has evolved and improved in many aspects with the efforts of various researchers over the years.

One such aspect of improvement in traditional style transfer methods were that they required reference style images, limiting flexibility. Text-Based Image style transfer revolutionized the field by using natural language descriptions to define styles, thus eliminating the need for reference images [2]. Evolution of various other models like the vision language models played a pivotal role in enabling TIST by mapping textual descriptions and images to a shared embedding space.

This section reflects our exploration of key advancements in style transfer. It examines the transition from traditional to text-based approaches. Through this exploration we aim to provide a comprehensive understanding of methodologies and innovation shaping this domain.

2.2 Methods Used in Style Transfer

We first collected several papers in the domain of style transfer and classified them into three categories based on the type of style transfer: **Text Style Transfer (TST), Image-to-Image Style Transfer (IIST), and Text-Based Image Style Transfer (TIST)**. We then arranged

these papers in chronological order to study their objectives and the improvements they aimed to achieve. This process helped us understand the limitations present in the field and provided a rough direction for selecting our baseline paper and formulating a solution to address one of its key limitations.

Image-to-Image Style Transfer (IIST)
Image-to-Image Style Transfer focuses on transforming one image's style to another while preserving the content. We explored three primary aspects within this domain:

1. **Artistic Style Transfer:**
    - Initiated by Gatys et al. (2019) in their seminal paper "Image Style Transfer Using Convolutional Neural Networks" [6].
    - This work introduced deep learning frameworks, specifically CNNs, to extract and recombine content and style features.
    - Their approach laid the foundation for neural style transfer (NST) by using pixel optimization with a loss function to balance style and content reconstruction.Other methods came up improving this domain [9][10][11]
2. **Photorealistic Style Transfer:**
    - Papers like "Line Search-Based Feature Transformation" (2023) introduced methods to preserve content while ensuring a strong adoption of style, suitable for realistic photographs [12].
    - This approach leveraged encoder-decoder architectures to balance content preservation and style application effectively.
    - Domain-Aware Universal Style Transfer (2021) combined CNNs with Whitening and colouring Transforms (WCT) to transfer not only style but also domain-specific properties [13].
3. **Facial Style Transfer:**
    - Techniques in this area focus on applying styles while maintaining facial features, ensuring recognizable and high-quality transformations for human subjects [14].

Text Style Transfer (TST)
Text Style Transfer modifies the stylistic attributes of text while preserving its semantic meaning. Key contributions include:

- "Text Style Transfer: A Review and Experimental Evaluation" (2022):Provided a comprehensive overview of 19 TST algorithms, focusing on changing textual stylistic properties.

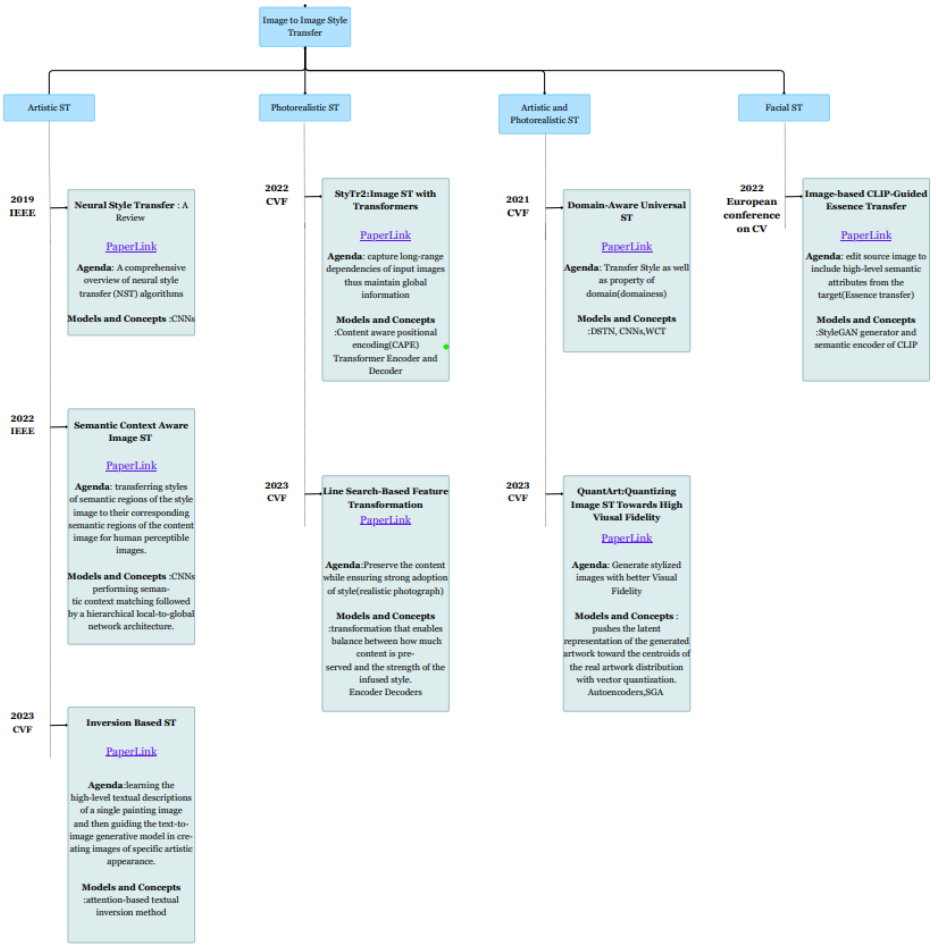

FIGURE 2.1: A synapse of IIST papers classified

[15]

Text-Based Image Style Transfer (TIST)

Text-Based Image Style Transfer enables users to define styles using textual descriptions instead of reference images, offering unparalleled flexibility [16][17][11]. Some key works include:

- **CLIPStyler (2022) [2]:**
  - Introduced the first framework to achieve style transfer without a style image, relying solely on textual descriptions.
  - Utilized the CLIP model, which maps text and images into a shared embedding space, enabling style representations to be extracted from text.
  - CLIPStyler employed patch-wise directional loss to align textual descriptions with image content, achieving global and fine-grained style details.
- **Semantic CLIPStyler (SEM-CS, 2023) [18]:**
  - Addressed issues like style spillover and content mismatch by segmenting the content image into salient and non-salient objects.

  - Applied styles selectively based on text descriptions to enhance semantic coherence.

- **FastCLIPStyler (2024): [19]**

  - Introduced optimization-free methods for faster, lightweight text-based style transfer, achieving efficient stylization in a single forward pass.

Baseline Paper: Multimodal Image Style Transfer (MMIST)
Our chosen baseline, MMIST [4], builds upon both IIST and TIST methodologies. It offers several innovations that address the limitations of prior methods:

- **Cross-Modal Style Representation:**

  - Encodes style inputs from multiple modalities (e.g., text, images) into a shared latent space, ensuring meaningful style and content separation.

- **Adapted Style Transfer Model:**

  - Uses an adapted IIST framework to synthesize stylized outputs while preserving structural details of the content image. However, it also provides the flexibility to choose any IIST model for style transfer once the style representations are achieved.

- **Multi-Style Boosting Mechanism:**

  - The idea of using multiple style representations of the same style enhances the quality of style representations, ensuring accurate and detailed stylistic transformations.

We selected MMIST as our baseline due to its ability to achieve the disentanglement of text and image embeddings in the shared space allowing integration of text and image inputs seamlessly, addressing artifact-related issues, and enabling flexible and high-fidelity style transfer. It provided a strong foundation for addressing some of the limitations of TIST and devising innovative solutions for future work.

Chapter 3

Proposed Work

3.1 Problem in MMIST's TIST and Other TIST Methods

Current techniques in MMIST's TIST and other TIST approaches have a major problem. They blur edges, which makes it hard to separate the foreground from the background. This blurring reduces the sharpness of structural details especially at the boundaries. As a result, features from the content image get lost. The edges should keep their structural integrity from the content image. Instead, they often become too stylized and lose their distinct look. Below, we illustrate this issue with magnified images highlighting blurred edge areas:

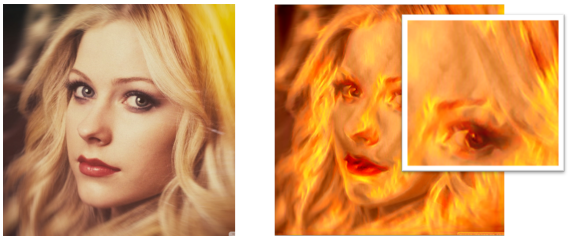

FIGURE 3.1: Blurred edges in fire stylized image by ClipStyler

FIGURE 3.2: Blurred edges in fire stylized image by MMIST's TIST

This drawback shows we need a method to keep a fine balance between content and style for each pixel. Such a method should preserve structural details while adding artistic style.

3.2 Motivation for Our Approach

We propose a method to improve stylization outcomes by introducing an appropriate proportion of content and style for each pixel. This proportion is decided by calculating the contextual loss between the original and stylized images. Contextual loss gauges how similar two images look, with a focus on keeping structural details intact [1].
The contextual loss paper states that contextual loss is a way to measure distance between the corresponding features in the latent space that aims to ignore the spatial positions of the features thus being able to tackle even the non-aligned data. Since this contextual loss will be captured on the high-level features (e.g. edges and structures) of the content and the stylized image obtained from the TIST model, a higher contextual loss shows that the image has moved further from the original meaning it has kept less of its structural details. So, to keep these structural details, the amount of content in the stylized image needs to go up as the contextual loss increases.

3.3 Understanding Contextual Similarity

The concept of contextual similarity is rooted in the idea that two images are similar if their corresponding sets of features are similar [1]. These need for finding corresponding that is the most contextually similar features between two images as shown in figure, enables the comparison of non-aligned data. Thus, the implementation of contextual similarity involves two stages where in first is to find the corresponding features i.e. make feature pairs out of the two sets of features maps. And then based on the sum of cosine distances between each feature pairs the total contextual similarity is calculated.

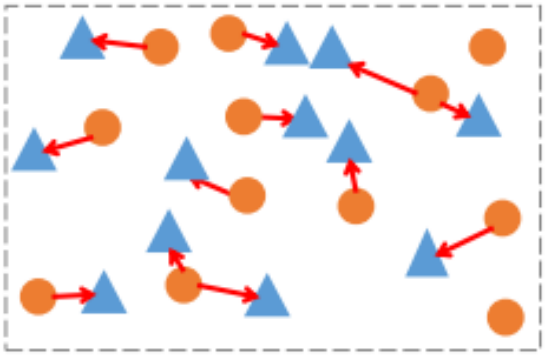

FIGURE 3.3: Orange circles represent the features of an image x while the blue triangles represent the features of a target image y. The red arrows match each feature in y with its most contextually similar feature in x [1]

The feature sets are typically extracted from deep layers of a pre-trained neural network (e.g., VGG19).

In our implementation:

- x = content image
- y = stylized_image_1 (obtained from the MMIST model)
- X = set of feature maps of content image
- Y = set of feature maps of stylized_image_1

Feature maps are obtained from conv4_2 and conv5_2 layers of (VGG19) (since deeper layers capture structural details of the image)

- $Conv4\_2 = 28 \times 28 \times 512 \; feature\; map\; size$
- $Conv5\_2 = 14 \times 14 \times 512 \; feature\; map\; size$

Step 1: Computing Similarity Between Feature Maps

To measure similarity, we compute a contextual similarity score for each pair of feature maps, x_i from the content image and y_j from the stylized_image_1.

1. **Cosine Distance:** The similarity between two feature maps,x_i and y_j, is first computed using the cosine distance:

$$d_{ij} = \left(1 - \frac{(x_i - \mu_y) \cdot (y_j - \mu_y)}{\|x_i - \mu_y\|_2 \|y_j - \mu_y\|_2}\right), \quad \text{where} \quad \mu_y = \frac{1}{N} \sum_j y_j \tag{3.1}$$

   where $\mu_y = \frac{1}{N} \sum_j y_j$ represents the mean of the target image feature maps.

2. **Normalization of Distance:** The distance is normalized to ensure numerical stability:

$$\tilde{d}_{ij} = \frac{d_{ij}}{\min_k d_{ik} + \epsilon} \tag{3.2}$$

   where ϵ is a small constant to avoid division by zero.

3. **Exponentiation to Obtain Similarity:** The normalized distance is then converted to similarity using an exponential function:

$$w_{ij} = \exp\left(\frac{1 - \tilde{d}_{ij}}{h}\right) \tag{3.3}$$

   where $h > 0$ is a bandwidth parameter that controls the sharpness of the similarity decay:

- A higher value of h results in a gradual decay of similarity.
- A lower value of h results in a sharper decay.

For this implementation, h=0.5 as recommended in the original paper.

4. **Final Similarity Between x_i and y_j:** The final similarity is normalized to ensure it lies between 0 and 1:
$$CX_{ij} = \frac{w_{ij}}{\sum_k w_{ik}} \tag{3.4}$$
A higher CX_{ij} indicates greater similarity between x_i and y_j.

Step 2: Overall Contextual Similarity

The overall contextual similarity between the content image x and the target image y is computed as:
$$\mathrm{CX}(x, y) = \mathrm{CX}(X, Y) = \frac{1}{N} \sum_j \max_i \mathrm{CX}_{ij} \tag{3.5}$$
Here, $\max_i \mathrm{CX}_{ij}$ ensures that for each feature map y_j, we find the feature map x_i that is most similar to it. Summing over all y_j gives the total contextual similarity.

3.4 Introducing the Blend Weight Concept

To address the limitations of existing methods, we incorporated the concept of "blend weight." Blend weight dynamically adjusts the contribution of the content and style for each pixel based on the contextual loss.

Description of Our Method

1. **Extract VGG Features:** The content image and the initial stylized image (referred to as "stylized_image_1") are passed through a pre-trained VGG19 network to extract their feature representations. These layers are particularly Conv4_2 and Conv5_2. These being the deeper layers in the network, capture high-level features like edges and structural details.

2. **Compute Contextual Loss:** Using the extracted features, the contextual loss is computed to measure the perceptual similarity and structural consistency between the content image and "stylized_image_1." The contextual similarity between two feature maps (content) and (stylized) is given as:
$$\mathrm{CX}(x, y) = \mathrm{CX}(X, Y) = \frac{1}{N} \sum_j \max_i \mathrm{CX}_{ij}$$
Here,

- x = content image
- y = stylized_image_1 (obtained from the MMIST model)
- X = set of feature maps of content image
- Y = set of feature maps of stylized_image_1
- CX_{ij} = contextual similarity between a feature map xi of content image and a feature map yj of stylized_image_1.

The detailed explanation of computing this contextual similarity is present in Contextual similarity paper(citation).
Contextual loss (cx_loss) can now be calculated as the negative log of this contextual similarity.

$$\mathcal{L}_{\mathrm{CX}}(x, y) = -\log\left(\mathrm{CX}\left(X, Y\right)\right)$$

3. **Calculate Blend Weight:** Having obtained the cx_loss, A blend weight is derived as:

$$blend\_weight = cx\_weight * clamp(cx\_loss, 0, 1) \tag{3.6}$$

Blend Weight has two terms:

(a) **cx_loss:** This captures the amount of mismatch in edge like features between the content image and the stylized_image_1.
(NOTE: cx_loss is clamped between 0 and 1 to ensure that when multiplied with image pixels doesn't lead to values out of range.)

(b) **cx_weight:** This is a tuneable hyperparameter which gives us a control to maintain the proportion of content and style in our output.

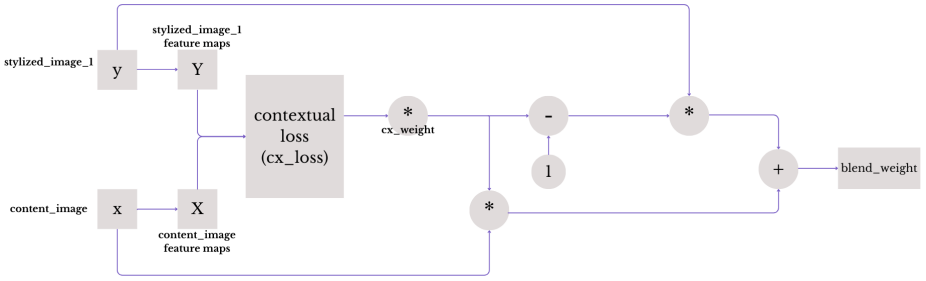

FIGURE 3.4: Flowchart of blend_weight calculation

4. **Blend Images:** Using the blend weight, the final image ("stylized_image_2") is computed as a weighted combination of the content image and "stylized_image_1":

$$stylized\_image\_2 = (1\text{–}blend_weight) * stylized\_image\_1 + blend\_weight * content \tag{3.7}$$

we can infer from this that since blend_weight is multiplied with content , if the cx_loss increases(indicating more difference between content and stylized_image_1) , blend_weight also increases, this pulls target closer to content hence increases the gradience (cause of edge in an image)

Pixel-Level Example

Consider a pixel level example to understand how blend_weight enhances edges i.e high gradience region (large difference between adjacent pixel values)
Consider a specific edge in the content image:

- **Content Image:** An edge exists between a light region (intensity = 200) and a dark region (intensity = 50).

$$At\ pixel(i,j) : content = 200. \tag{3.8}$$

$$At\ pixel(i+1,j) : content[i+1,j] = 50. \tag{3.9}$$

- **Target Image:** The same region is blurry, with pixel values smoothed to intermediate values:

$$At\ pixel(i,j) : target[i,j] = 120. \tag{3.10}$$

$$At\ pixel(i+1,j) : target[i+1,j] = 100. \tag{3.11}$$

- **Blending at Pixels:** Using the blending formula:

$$targetnew[i,j] = (1 - blend\_weight) * target[i,j] \ + \ blend\_weight * content[i,j] \tag{3.12}$$

  If blend_weight = 0.8:
  - At (i,j) :

$$targetnew[i,j] = (1 - 0.8) * 120 \ + \ 0.8 * 200 = 176 \tag{3.13}$$

  - At (i+1,j) :

$$targetnew[i+1,j] = (1 - 0.8) * 100 \ + \ 0.8 * 50 = 60 \tag{3.14}$$

- **Result:**
  - The new target image now has values 176 and 60 at (i,j) and (i+1,j), respectively.
  - This increases the contrast between the two pixels, sharpening the edge and aligning it more closely with the content image.

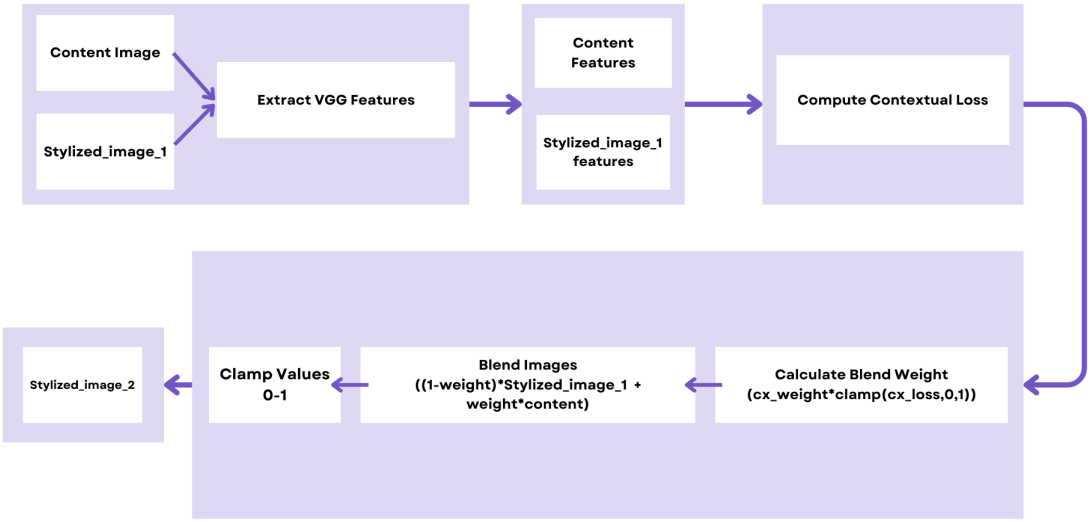

FIGURE 3.5: Flowchart of complete implementation of our method on MMIST generated stylized_image_1

Chapter 4

Simulation and Results

The whole purpose of our approach was to enhance or restore the content features of the stylized_image_1 obtained from MMIST's TIST method. The proportion of content is controlled by blend_weight which is indeed made up of cx_loss and cx_weight (equation 3.6). Cx_loss is a term which cannot be altered by us since it fully depends on the content and the stylized image, but cx_weight is a parameter which is tuneable. It is evident that from the image blending equation 3.7 that as we increase cx_weight the blend_weight increases, and thus the proportion of content increases in our final output.
Though this is what we looked for, this also leads to loss in style eventually. Therefore cx_weight needs to be set to a point such that we get a perfect balance between content and style.
This would first require us to find appropriate metrices for measuring content and style of our final output image.
We used "Structural similarity index measure (SSIM)" and "Difference in Gram Matrix" for content and style measurement respectively.

4.1 Structural Similarity Index measure (SSIM)

This is a metric which is used to measure similarity between two images, based on the perceptual quality of the images [20]. It compares three aspects of the image, that are luminance, contrast and structure (here it also captures the blurriness of the images). It generates a value between -1 and 1, where 1 indicates perfect similarity, 0 indicates no similarity, and negative values suggest dissimilarity.
Thus, a higher SSIM score would indicate greater similarity between images.
In our case, we would be computing SSIM score between stylized_image_2 (our output) and content image to get their structural similarity.
The SSIM formula as given below, involves comparing local patterns of pixel intensities in the domain of luminance, contrast, and structure.

The SSIM between two images x and y is defined as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

where:

- μ_x and μ_y are the average luminance of the two images being compared.
- σ_x^2 and σ_y^2 are the variances of the pixel intensities in the two images.
- σ_{xy} is the covariance of the pixel intensities between the two images.
- C_1 and C_2 are constants added to prevent instability when the denominators are close to zero.

4.2 Gram Matrix Difference

Gram matrix is a mathematical tool which was used by Gatys et al. in their Neural Style Transfer [6] method to represent the style of an image. By Style we mean the overall texture and pattern distributions of the image which are independent of their exact position. It is based on the idea that CNNs extract hierarchical features, thus each layer detects some patterns like edges, textures or shapes. The style is a relationship between these patterns, for example, whether certain textures appear together in the image. This is done by capturing the correlations between different features(filters) of CNN layers.

Gram Matrix Calculation: For a feature map F of dimensions $n_C \times n_H \times n_W$:

- Flatten F into a matrix of size $n_C \times (n_H \cdot n_W)$, where:
  - n_C: Number of channels (filters).
  - n_H, n_W: Height and width of the feature map.
- The Gram matrix G is obtained by multiplying this matrix with its transpose:

$$G[i, j] = \sum_k F[k, i] \cdot F[k, j]$$

  This measures the correlation between feature maps i and j.

Style Loss: A difference in the gram matrix of generated image with that of the style image would give the style loss. We want this difference to be minimized to ensure good style transfer.
In our case, we would be computing Gram Matrix Difference score between stylized_image_2 (our output) and stylized_image_1 (MMIST's TIST output) to get their style difference.

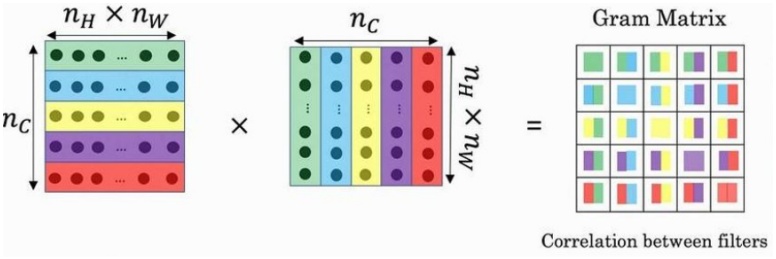

FIGURE 4.1: Gram matrix demonstration

4.3 cx_weight (Hyperparatmeter) tunning

Using these measures we computed best value for cx_weight for 5 different styles. We combined each style with 500 content images from Flickr30k dataset[].Thus having generated 5×500 stylized_images for each experimental value of cx_weight that are 0.2, 0.4, 0.6 and 0.8 respectively, we then calculated the SSIM and Difference of Gram Matrix scores.The tables below provide an average value over all stylized images for each style with each cx_weight.

NOTE: cx_weight with a value of 0 would give the original stylized_image_1 since this makes the blend_weight 0, therefore zero content would be added.

Style: Fire	**cx_weight**	**SSIM**	**Difference of Gram Matrix**
	0.2	0.572	0.271
	0.4	0.710	0.292
	0.6	0.804	0.367
	0.8	0.931	0.456

TABLE 4.1: Table displaying different cx_weight values with corresponding SSIM and Gram Matrix differences for ”Fire” Style.

Style: Pop Art	**cx_weight**	**SSIM**	**Difference of Gram Matrix**
	0.2	0.391	0.095
	0.4	0.581	0.148
	0.6	0.722	0.271
	0.8	0.951	0.342

TABLE 4.2: Table displaying different cx_weight values with corresponding SSIM and Gram Matrix differences for ”Pop Art” Style.

Style: Mosaic	cx_weight	SSIM	Difference of Gram Matrix
	0.2	0.511	0.289
	0.4	0.683	0.302
	0.6	0.732	0.368
	0.8	0.921	0.401

TABLE 4.3: Table displaying different cx_weight values with corresponding SSIM and Gram Matrix differences for "Mosaic" Style.

Style: The great wave off Kanagawa	cx_weight	SSIM	Difference of Gram Matrix
	0.2	0.463	0.412
	0.4	0.520	0.591
	0.6	0.612	0.615
	0.8	0.725	0.741

TABLE 4.4: Table displaying different cx_weight values with corresponding SSIM and Gram Matrix differences for "The great wave off Kanagawa by Katsushika Hokusai" Style.

Style: White Wool	cx_weight	SSIM	Difference of Gram Matrix
	0.2	0.449	0.249
	0.4	0.538	0.439
	0.6	0.679	0.678
	0.8	0.919	0.919

TABLE 4.5: Table displaying different cx_weight values with corresponding SSIM and Gram Matrix differences for "White Wool" Style.

Below is the table with the best values of cx_weight for each of the 5 styles along with their SSIM and Gram Matrix difference scores.
NOTE: The best values corresponds to a value out of the four values of cx_weight we considered which is closest to the intersection point of (1-SSIM) and Difference of Gram Matrix score. SSIM increases as more content proportion is taken that is higher value of cx_weight but this increase leads to a increase in Gram matrix difference as well since the style depreciates.Hence the best cx_weight would be the one that maintains a balance between these two.

4.4 Comparison with Baseline method

these best values indicated 0.4 to a possible optimal value for cx_weight (from table 4.6). We used the same measure to compare our method with 0.4 cx_weight with the baseline method,

Style	Best cx_weight value	SSIM	Gram Matrix Difference
Fire	0.4	0.710	0.292
Pop Art	0.6	0.722	0.271
Mosaic	0.4	0.683	0.302
The great wave off Kanagawa	0.4	0.520	0.591
White Wool	0.4	0.538	0.439

TABLE 4.6: Comparison of Best cx_weight Value, SSIM, and Gram Matrix Difference for Different Styles

which is MMIST's TIST by averaging the scores over the 5 styles and 500 content images.

	SSIM	Gram Matrix Difference
Ours	0.606	0.394
MMIST's TIST	0.589	0.288

TABLE 4.7: Comparison of SSIM and Gram Matrix Difference for ours and baseline Method

CX_WEIGHT	FIRE	POP ART	SSIM SCORE	GRAM MATRIX DIFFERENCE
0.0			0.512	0.186
0.2			0.579	0.269
0.4			0.703	0.283
0.6			0.805	0.367
0.8			0.934	0.456

FIGURE 4.2: cx_weight tuning demonstration

Chapter 5

Conclusions and Future Work

In this report, we presented a method to enhance the results of existing style transfer frameworks like MMIST's TIST and address its limitations. Our method focused on adjusting the balance between the content and the style of each pixel using a blend weight from contextual loss. This improved the foreground-background separation and preserved the important structural details while also ensuring artistic style fidelity. Through detailed experimentation and extensive hyperparameter tuning we proposed a best value for the cx_weight parameter and showed visual comparisons. Our approach improved stylization quality, particularly in edge regions where other methods often struggled.

While this method did indeed improve results by adding structural details, it is possible that a more advanced deep learning-based method, such as training autoencoders or other neural network architectures, could allow for even more sophisticated and meaningful blending. This may further enhance the integration of content and style into a seamless and harmonious fusion.

Bibliography

[1] R. Mechrez, I. Talmi, and L. Zelnik-Manor, "The contextual loss for image transformation with non-aligned data," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 768–783, 2018.

[2] G. Kwon and J. C. Ye, "Clipstyler: Image style transfer with a single text condition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18062–18071, 2022.

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*, pp. 8748–8763, PMLR, 2021.

[4] H. Wang, P. Wu, K. D. Rosa, C. Wang, and A. Shrivastava, "Multimodality-guided image style transfer using cross-modal gan inversion," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4976–4985, 2024.

[5] Z. Xu, S. Xing, E. Sangineto, and N. Sebe, "Spectralclip: Preventing artifacts in text-guided style transfer from a spectral perspective," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5121–5130, 2024.

[6] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2414–2423, 2016.

[7] F. AI, "Style transfer," 2024.

[8] V. Labs, "Neural style transfer," 2024.

[9] Z. F. J. Y. Y. Y. M. S. Yongcheng Jing, Yezhou Yang, "Neural style transfer: A review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3365–3385, 2019.

[10] Y.-S. Liao and C.-R. Huang, "Semantic context-aware image style transfer," *IEEE Transactions on Image Processing*, vol. 31, pp. 1911–1923, 2022.

[11] Y. Zhang, N. Huang, F. Tang, H. Huang, C. Ma, W. Dong, and C. Xu, "Inversion-based style transfer with diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10146–10156, 2023.

[12] T.-Y. Chiu and D. Gurari, "Line search-based feature transformation for fast, stable, and tunable content-style control in photorealistic style transfer," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 249–258, 2023.

[13] K. Hong, S. Jeon, H. Yang, J. Fu, and H. Byun, "Domain-aware universal style transfer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14609–14617, 2021.

[14] H. Chefer, S. Benaim, R. Paiss, and L. Wolf, "Image-based clip-guided essence transfer," in *European Conference on Computer Vision*, (Cham), pp. 695–711, Springer Nature Switzerland, 2022.

[15] C. C. A. A. Z. Zhiqiang Hu, Roy Ka-Wei Lee, "Text style transfer: A review and experimental evaluation," *ACM SIGKDD Explorations Newsletter*, vol. 24, no. 1, pp. 14–45, 2022.

[16] Z.-S. Liu, L.-W. Wang, W.-C. Siu, and V. Kalogeiton, "Name your style: Text-guided artistic style transfer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3530–3534, 2023.

[17] N. Huang, Y. Zhang, F. Tang, C. Ma, H. Huang, W. Dong, and C. Xu, "Diffstyler: Controllable dual diffusion for text-driven image stylization," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[18] C. G. Kamra, I. D. Mastan, and D. Gupta, "Sem-cs: Semantic clipstyler for text-based image style transfer," in *2023 IEEE International Conference on Image Processing (ICIP)*, pp. 395–399, IEEE, 2023.

[19] A. P. Suresh, S. Jain, P. Noinongyao, A. Ganguly, U. Watchareeruetai, and A. Samacoits, "Fast-clipstyler: Optimisation-free text-based image style transfer using style representations," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7316–7325, 2024.

[20] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, and L. Vanneschi, "Structural similarity index (ssim) revisited: A data-driven approach," *Expert Systems with Applications*, vol. 189, p. 116087, 2022.