HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

Dual Encoder Transformers for Efficient and Versatile Image Style Transfer

Saket Joshi SAKET_JOSHI@G.HARVARD.EDU

Nishtha Sardana NISHTHASARDANA@G.HARVARD.EDU

Kareema Batool KAREEMABATOOL@G.HARVARD.EDU

Nikhil Nayak NNAYAK@G.HARVARD.EDU

Abstract

Image style transfer, the process of applying a specific artistic style to an input image while preserving its content, has been an area of interest in recent years. Traditional methods that train a separate neural network for each style are inefficient and time-consuming, motivating the need for a faster and more versatile approach. In this study, we investigate the application of state-of-the-art (SOTA) dual encoder transformer-based architectures for efficient and versatile image style transfer, focusing on incremental improvements and refinements.

Building upon existing transformer-based architectures, our approach utilizes two separate transformer encoders to generate domain-specific sequences for both content and style images. The content encoder focuses on preserving content information, while the style encoder captures the unique features of the artistic style. A multi-layer transformer decoder then stylizes the content sequence based on the style sequence, allowing for seamless integration of style features while maintaining the input image's content. We experiment with various refinements in positional encoding techniques, including content-aware positional encoding (CAPE), to evaluate their impact on style transfer tasks.

We conducted comprehensive qualitative and quantitative experiments to assess the effectiveness of our refined dual encoder transformer-based approach. The results demonstrated that the incremental improvements we introduced led to enhanced performance in terms of style transfer quality, versatility, and computational efficiency, compared to the original SOTA methods. To showcase the capabilities of our approach, we further developed and published a web application that demonstrates the image style transfer technique in real-time, providing an accessible and interactive platform for users to explore and experiment with various artistic styles [1]. This work contributes to the ongoing research in image style transfer and highlights the potential of incremental refinements in improving the efficiency, flexibility, and convenience of applying various artistic styles to any input image.

1. The web application can be accessed at https://saketrule-harvard-cs-final-projectdemostyle-transfer-app-wkdeqk.streamlit.app/

1. Introduction

The advent of deep learning has revolutionized the field of computer vision, enabling the development of advanced techniques for various image processing tasks. One such task, image style transfer, has attracted significant attention due to its ability to combine the content of an input image with the artistic style of a reference image, resulting in visually appealing and artistically inspired outputs. This process has found applications in diverse areas, including content creation, entertainment, advertising, and digital art.

Traditional approaches to image style transfer involve training a separate neural network for each artistic style, leading to increased computational overhead and reduced versatility. Recent advances in deep learning, particularly the introduction of transformer-based architectures, have paved the way for more efficient and flexible methods for image style transfer. These state-of-the-art (SOTA) techniques employ dual encoder transformers that generate domain-specific sequences for both content and style images, followed by a multi-layer transformer decoder that fuses the two sequences to produce the stylized output.

In this study, we aim to explore and investigate the effectiveness of SOTA dual encoder transformer-based architectures for image style transfer, focusing on making incremental improvements and refinements to enhance their performance. We experiment with various positional encoding techniques, including content-aware positional encoding (CAPE), to assess their impact on style transfer tasks. Our goal is to demonstrate that through careful experimentation and refinement, it is possible to improve the efficiency, flexibility, and convenience of existing methods for applying various artistic styles to any input image.

The remainder of this paper is organized as follows: Section 2 provides a brief overview of related work in image style transfer, highlighting the evolution of techniques and methods. Section 3 details our refined dual encoder transformer-based approach, including the architectural design and the refinements introduced. Section 4 presents the experimental setup, evaluation metrics, and results, showcasing the effectiveness of our approach. Finally, Section 5 concludes the paper and discusses potential directions for future research.

2. Motivation

The motivation behind this study stems from several key factors that highlight the importance and potential benefits of refining and improving image style transfer techniques. These factors can be categorized into three primary aspects: practical applications, computational efficiency, and research interest.

- Practical Applications: Image style transfer has a wide range of practical applications across various industries. In content creation, it enables artists and designers to generate unique and visually appealing images by applying different artistic styles to source images, streamlining the creative process. In the entertainment industry, style transfer can be used to create distinctive visual effects, enhance film scenes, or generate promotional material for movies and video games. Moreover, in advertising, it allows for the development of eye-catching and engaging marketing materials that resonate with target audiences. By refining and improving image style transfer techniques, we contribute to the advancement of these practical applications, unlocking new possibilities for creativity and innovation.

- Computational Efficiency: Traditional image style transfer methods that rely on training separate neural networks for each style suffer from high computational costs and limited flexibility. In contrast, dual encoder transformer-based approaches have shown promise in reducing these limitations. By focusing on incremental improvements and refinements to these state-of-the-art techniques, we aim to further enhance their computational efficiency, making them more accessible to researchers, practitioners, and end-users. Improved efficiency also enables real-time applications, such as live video stylization or interactive artistic tools, broadening the scope of style transfer applications.

- Research Interest: Image style transfer has been an active area of research in recent years, with ongoing efforts to develop novel techniques and methods that offer better performance and versatility. By investigating the effectiveness of existing state-of-the-art dual encoder transformer-based architectures and refining them through experimentation, we contribute to this body of research, providing valuable insights and findings that can guide future work in the field. Our study may inspire other researchers to explore similar refinements or investigate alternative approaches to further advance the capabilities of image style transfer techniques.

In summary, the motivation for this study lies in the potential benefits and advancements that can be achieved by refining and improving existing image style transfer methods, leading to enhanced practical applications, increased computational efficiency, and continued research interest in the field.

3. Data

3.1 Dataset for Content Images

The COCO (Common Objects in Context) dataset is a widely used resource for computer vision tasks such as image recognition, segmentation, and captioning. It comprises over 330,000 diverse and complex images, which showcase a broad spectrum of object categories, including animals, people, vehicles, household objects, and natural scenes. The dataset is a project of Microsoft Research, and it was created to address the need for a large-scale dataset with diverse and complex images that could be used to develop and evaluate computer vision models. The dataset provides rich and detailed object annotations such as object categories, object bounding boxes, and object key-points, in addition to object instance segmentation masks. It can be accessed through the COCO website. The website also provides information about how to access the dataset using APIs for Python. We have leveraged the same for the purposes of this project. We will be using the same dataset for both content and style images, instead of using separate datasets for each, has several advantages:

- **Better consistency**: Using the same dataset for both content and style images ensures greater consistency between them, as they will share the same visual characteristics, such as color palette, texture, and shape.

- **Reduced dataset requirements**: Using a single dataset reduces the amount of data needed for training, as you don't need to acquire and pre-process two separate datasets. This can save time and resources.
- **More meaningful representations**: When using separate datasets for content and style, the model may learn to focus more on the style features, as they may be more distinctive and easily distinguishable. However, using the same dataset ensures that the model must learn to separate content and style based on more subtle differences, resulting in more meaningful representations.
- **Easy to train**: Since you are only using one dataset, you can train your model more easily and efficiently. You don't need to worry about adjusting two different datasets or making sure they have similar properties.
- **Greater flexibility**: Using the same dataset for both content and style images provides greater flexibility when it comes to creating new combinations of content and style. You can mix and match images from the same dataset in any way you like, without having to worry about whether the styles and content are compatible.

3.2 Dataset for Style Images

The WikiArt dataset contains over 200,000 images of various styles and genres, including painting, sculpture, and digital art. The images are sourced from the WikiArt.org website, which is an online art encyclopedia that features information and images of artworks from various cultures and periods. Each image in the dataset is labeled with metadata including the title of the artwork, the name of the artist, and the year it was created. This metadata can be useful for researchers who are interested in studying art history or tracking trends in artistic styles over time.

The images in the WikiArt dataset are available in high-resolution format, which makes it suitable for training deep learning models that require large amounts of training data. The dataset has been preprocessed to ensure that all images are of uniform size and aspect ratio, which simplifies the training process.

3.3 Dataset Collection Methodolody

To gather the datasets for this project, we utilized the COCO dataset as the content dataset, which was downloaded from the COCO 2017 Dataset available at Kaggle - Coco Datset. For the style dataset, we downloaded images from Wikiart using a Python script. The script used to obtain the style dataset can be found on the project's GitHub page at the following file path: "StyleTransfer/wikiart.py".

4. Exploratory Data Analysis

4.1 Summary

We started our analysis by examining several characteristics of the images in the dataset including their data type, number of channels, as well as the minimum and maximum values for both their width and height.

- Number of Images: 5000
- Data Type of Images: torch.uint8
- Shape of first Image: torch.Size([426, 640, 3])
- Number of channels in each image: 3
- Minimum Width: 200
- Maximum Width: 640
- Minimum Height: 145
- Maximum Height: 640

4.2 Image Statistics for All Color Channels

As part of our exploratory data analysis (EDA), we examined the pixel intensity statistics and histograms for the red, green, and blue channels in a sample of images from the COCO dataset. Our goal was to gain insights into the contrast distribution and color characteristics of the images in the dataset. We started by plotting the mean, minimum, maximum and range of the images for all the three channels. The box plots for mean, max, min, and range of pixel values for the three channels show that the data is distributed across a wide range of values. The median value for the mean pixel values is around 100 [highest for red channel], while the median value for the min pixel value is around 0. The range of pixel values is also quite high, with some images having values as high as 255 and as low as 0. This suggests that the dataset contains a diverse set of images with different levels of brightness and contrast. The box plots indicate the presence of outliers in the dataset, where certain images exhibit very high or very low pixel intensities that are far from the typical range of values. Such outliers may stem from a variety of reasons, including over or underexposure, atypical lighting conditions, or image artifacts.

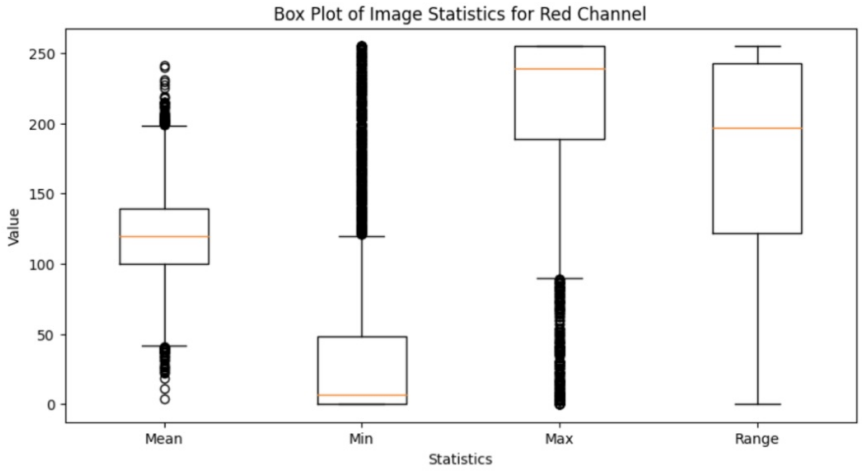

Figure 1: Image Statistics - Red Channel

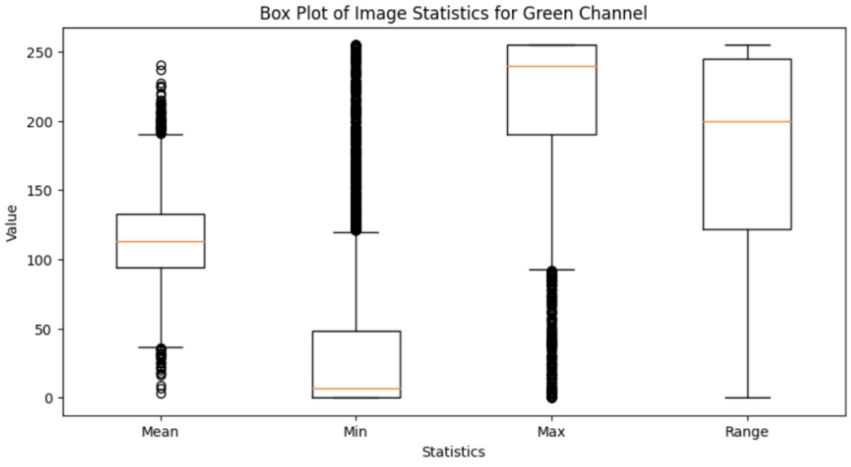

Figure 2: Image Statistics - Green Channel

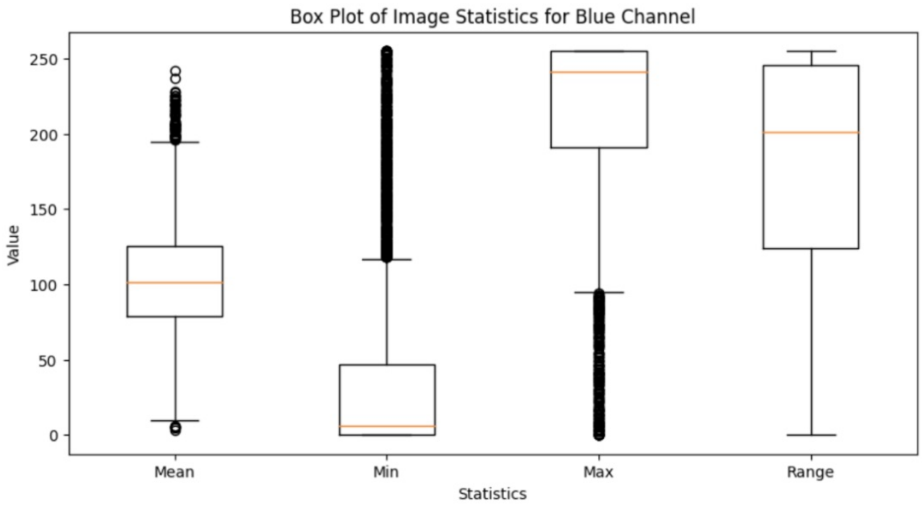

Figure 3: Image Statistics - Blue Channel

4.3 Histogram

By plotting and analyzing the histogram of several random images, we observed that most images had pixel values spread out between 0 and 255. Furthermore, we noticed a diverse range of images, some with a concentration of pixel values in the bright region and others in the dark region. This variety indicates that the images are suitable for neural style transfer since they provide a good range of content and style features. Additionally, the well-spread histogram of pixel values suggests that the contrast of the images is adequate. Overall, these observations lead us to conclude that these sets of images are well-suited for our neural style transfer task.

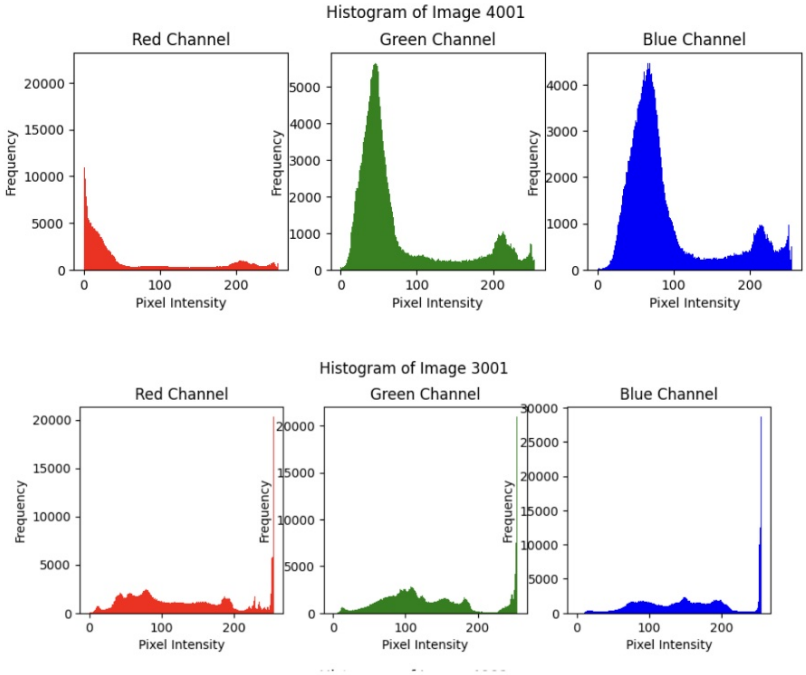

Figure 4: Histogram for Images

4.4 Image Pyramid

As part of our data preprocessing for neural style transfer, we used an image pyramid to enhance the quality of the output. The image pyramid is a multi-scale representation of the image, where each level of the pyramid represents a different scale of the original image. We will use this technique to improve the efficiency and accuracy of our style transfer algorithm. We utilized the image pyramid to break down the content and style images into multiple scales, from low resolution to high resolution. This allowed us to process the images at different scales, applying the style transfer algorithm at each level. By doing this, we will be able to preserve the fine details in the content image while still incorporating the style features of the style image. The image pyramid also allowed us to perform style transfer on larger images, which would have otherwise been computationally expensive or impossible. We could work on smaller images first and then gradually build up to the full resolution of the input images, resulting in a faster and more accurate style transfer.

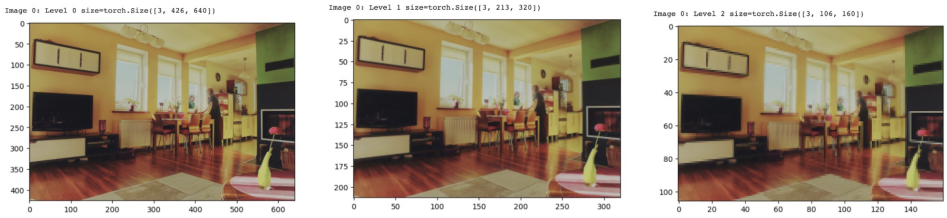

Figure 5: Image Pyramid

4.5 Pre-processing

We pre-processed the dataset for training deep learning models. We specifically applied three main pre-processing techniques, namely: mean imputation, normalization, Gaussian noise, and resizing, mean imputation. First, we performed mean imputation of NaN values in a 3-channel RGB image using neighboring pixels (3x3 neighborhood). Second, we normalized the pixel values of the images to rescale them between 0 and 1. We defined transformations to normalize/standardize the images across the 3 channels. It is a common practice to use mean and standard deviation of ImageNet. The computer vision literature recommends using the mean and standard deviation of ImageNet. However, we're using the default pixel normalization in images, which scales pixel values between 0 and 1. We've found that this configuration works well in practice. Second, we added Gaussian noise to the images to introduce some randomness. Finally, we re-sized the images to a fixed size of 64x64 pixels to standardize the input size of the images. These pre-processing steps can help enhance the performance of the models by making the model more robust to real world variations.

4.6 Class Imbalance

To address the issue of class imbalance, we ensured that all kinds of styles and contents were considered in our implementation of neural style transfer. We carefully curated the dataset to ensure it contained a wide variety of images with different styles and contents. This included selecting images from different artistic movements, as well as including images from different genres such as landscapes, portraits and stills.

Figure 6: Set of Pre-Processed Images

5. Baseline Model

5.1 Method

The baseline model used in our neural style transfer task is a 19 layer VGG network, which is similar to the one used in the original paper. We used PyTorch's implementation of VGG, which is divided into two child Sequential modules: features and classifier. We used the features module because we needed the output of the individual convolution layers to measure content and style loss. To train and validate the model, we used the COCO dataset for content images and the Flickr dataset for style images.

For evaluating the performance of our model, we used two loss metrics: content loss and style loss. The content loss was implemented as a torch module that represented a weighted version of the content distance for an individual layer. We added this content loss module directly after the convolution layer(s) that were being used to compute the content distance. This way, each time the network was fed an input image, the content losses were computed at the desired layers, and all the gradients were computed via auto grad. We used the mean square error between the two sets of feature maps to calculate the content distance, and nn.MSELoss to compute the distance.

The style loss module was implemented similarly to the content loss module. To calculate the style loss, we needed to compute the gram matrix, which was the result of multiplying a given matrix by its transpose matrix. In our application, the given matrix was a reshaped version of the feature maps. Finally, the gram matrix was normalized by dividing each element by the total number of elements in the matrix, to counteract the fact that the N dimension yielded larger values in the Gram matrix. This normalization was crucial since the style features tend to be in the deeper layers of the network, and this step ensured that the first layers (before pooling layers) did not have a larger impact during the gradient descent. Overall, these evaluation metrics helped us assess the performance of our model accurately.

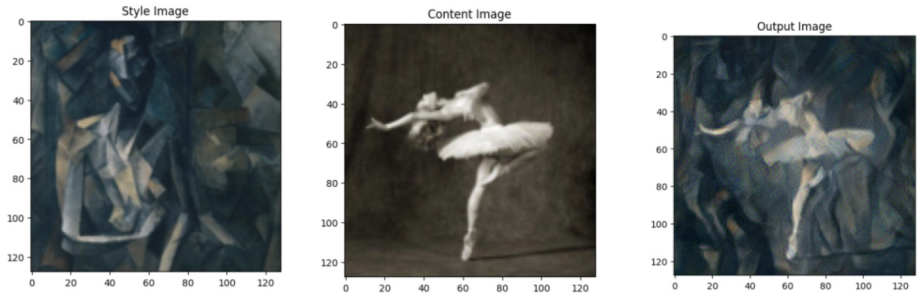

Figure 7: Results of Baseline Model

5.2 Findings

In neural style transfer, we create a correlation plot using two sets of randomly selected content and style images to assess their similarity. The plot helps us understand that content and style images can be vastly different from each other, making it difficult to use gradient descent to obtain a blended image for every content-style image pair. As a result, we conclude that the baseline model is designed to handle each content/style image separately. However, to create a more flexible approach, we require a model that can accommodate any content image with a specific style image or a network that can work with any content-style image combination. Our project's future milestone will involve building this.

The neural style transfer approach involves finding the optimal combination of content and style features to generate a new image that combines the content of one image with the style of another. This process involves initializing the generated image with the content

image, and then iteratively adjusting the pixel values of the generated image to match the style of the style image while preserving the content information.

Although the baseline model is effective in generating stylized images, it suffers from the problem of slow processing time and computational resources due to the need to reset the generated image pixels and perform the pixel search process for each new content image. As a result, the traditional approach is not suitable for real-time production environments and can be computationally expensive.

Additionally, the traditional approach does not guarantee good results, as the generated image may not accurately reflect the desired style or may introduce unwanted artifacts. This is because the optimal combination of content and style features is highly dependent on the specific content and style images used, making it difficult to generalize the approach to new images.

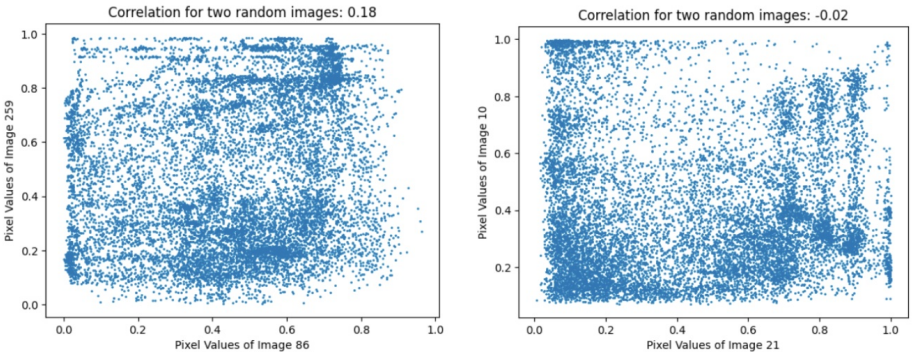

Figure 8: Findings

6. Updated Problem Statetment

To overcome these limitations, we plan to employ a generalized approach by training a neural network to apply a specific type of style on any input image. A single model can be trained for each style which would eliminate the need for repeated training. Although this approach may have some limitations, it may offer a faster solution than the traditional approach. Hence, we plan to further work on fast neural style transfer. It is an optimized version of neural style transfer that significantly reduces the processing time required to generate stylized images. This approach utilizes a pre-trained convolutional neural network (CNN) to extract style features from a style image, and then applies those features to a content image using a feedforward neural network.

Moving forward, our plan is to develop two approaches: the first approach involves building a dedicated model for each style image, which can then be used with any content image to generate the stylized output. The second approach involves building a single, versatile model that can be used with any combination of style and content images, allowing for greater flexibility and convenience in the style transfer process.

7. Related Work

7.1 Fast Neural Style Transfer

The original style transfer method proposed by Gatys involves an iterative optimization process that generates a styled output image by minimizing a loss function that combines the content loss and style loss between the input image and the style image. Due to the iterative nature of the optimization, this method can be slow and impractical for real-time applications, such as generating stylized videos. To address this issue, the "Perceptual Losses for Real-Time Style Transfer and Super-Resolution" paper introduced a fast neural style transfer method that uses a pre-trained feedforward neural network to directly map an input image to an output image with the desired style. The network is trained on a dataset of image pairs consisting of a content image and a corresponding style image, where the loss function used during training is the same perceptual loss function used in the original style transfer method. By learning the mapping between the input image and styled output image with a neural network, the fast style transfer method allows for generating infinite images with the same style in real-time without the need for iterative optimization. This makes it suitable for real-time applications such as styling videos.

While the fast neural style transfer method offers several advantages over the original iterative optimization-based style transfer method, it also has some limitations that should be considered.

- One of the main limitations of the fast neural style transfer method is that it requires a large amount of training data to generate high-quality stylized images. This is because the method relies on a pre-trained feedforward neural network to learn the mapping between the input image and the output stylized image. If the network is not trained on a diverse and representative set of image pairs, it may fail to capture the full range of style variations and produce suboptimal results.

- Another limitation of the fast neural style transfer method is that it is not as flexible as the original style transfer method in terms of controlling the degree and style of the stylization. The method relies on the pre-trained neural network to generate the stylized output, and it may not be possible to fine-tune the stylization parameters in real-time without retraining the entire network.

- Additionally, the fast neural style transfer method may produce some artifacts or distortions in the stylized output due to the non-iterative nature of the mapping process. These artifacts may be more apparent in high-contrast or complex images, and they may require post-processing to correct.

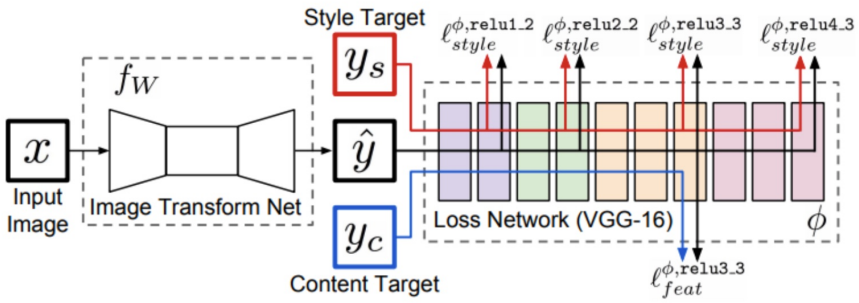

Figure 9: Architecture for Fast Neural Style Transfer

7.2 CNN Based Encoder Decoder

Unfortunately, the fast neural style transfer approach comes at the cost of the network being tied to a fixed set of styles and the inability to adapt to arbitrary new styles. The "Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization" paper proposes an adaptive instance normalization (AdaIN) method for arbitrary style transfer that builds on the fast neural style transfer method by introducing a learnable affine transformation to normalize the feature maps.Compared to the fast neural style transfer algorithm, which uses fixed normalization parameters for all images, the AdaIN method introduces more flexibility and fine-grained control over the stylization process, while still maintaining the real-time speed and efficiency of the fast algorithm.

The CNN based methods for learning style and content representations have limitations in capturing long-range dependencies due to the limited receptive field of convolutional operations. While increasing the depth of the network can help capture more complex dependencies, it can also lead to loss of feature resolution and fine details. This missing information can negatively affect the stylization results in terms of preserving the content structure and displaying the style. Furthermore, typical CNN-based style transfer methods have a bias toward content representation, as the stylization process can cause the extracted structures of the input content to change significantly after multiple rounds of stylization operations. The content leak issue usually occurs in the stylization process because CNN-based feature representation may not sufficiently capture details in the image content. This content leak can result in a loss of style information and a suboptimal stylization outcome.

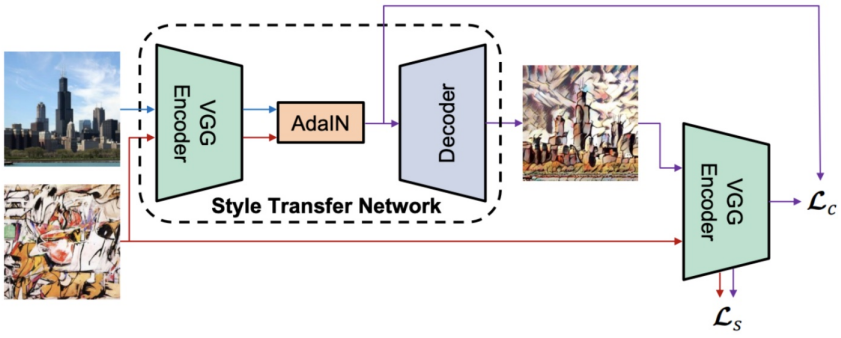

Figure 10: Architecture for CNN Based Encoder Decoder

7.3 Tranformer with one Encoder

The transformer-driven style composition module uses multi-head attention, inspired by the transformer architecture, to globally combine the style codes. The entire network is divided into three parts: style bank generation, transformer-driven style composition, and parametric content modulation. Based on these modules, our feed-forward arbitrary style transfer method, referred to as StyleFormer, can produce visually plausible stylization results for various artworks, while ensuring the style diversity with fine-grained style details, and the content coherence with the input content images

The proposed method starts by creating a set of learnable style codes and finding their global composition. Then, it uses this global style representation to modulate the content features. The transformer driven architecture helps to create new style distributions that are in line with the content structures, while also capturing the finer style variations. The end result is a stylized image that belongs to the style manifold of the input style images.

The method involves processing the style and the content images by an encoder to obtain their respective content and style features. The content feature contains information about the content of the image (i.e., the objects and their arrangement), while the style feature contains information about the style of the image (i.e., the colors, textures, and patterns). After obtaining the content and style features from the input images, a new feature transfer module is used to combine these features in a way that produces the desired stylized output. This module includes several sub-modules, such as the style bank generation, transformer-driven style composition, and parametric content modulation. These sub-modules work together to transform the content feature to the stylized feature based on the style feature. Finally, the stylized feature is fed into a decoder to generate the final stylized image.

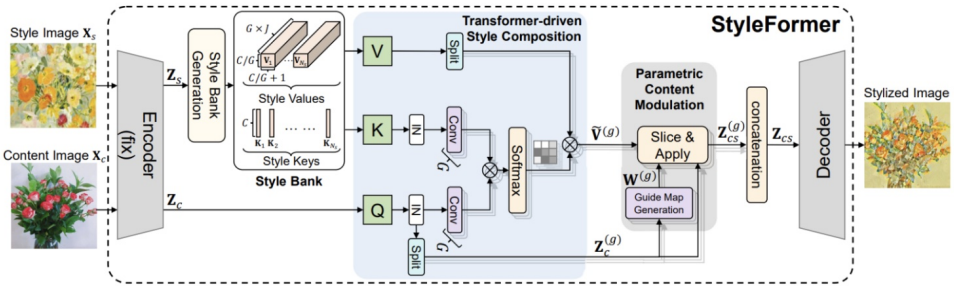

Figure 11: Architecture for Transformer based model with one encoder

8. Method and Approach

As discussed above, traditional neural style transfer methods face biased content representation. This happens because owing to the locality in convolutional neural networks, extracting and maintaining the global information of input images is difficult. To tackle this, a transformer-based approach is utilized to account for long-range dependencies in input images during style transfer.

As opposed to the previous model, two different transformer encoders were employed to generate domain specific sequences for content and style. Following the encoders, a multi-layer transformer decoder is adopted to stylize the content sequence according to the style sequence.

8.1 Style Transfer Transformer

In our exploratory data analysis, we have used the concept of co-integration to find the correlation between different stock pairs. We have

- **Transformer Encoder:** Two transformer encoders are leveraged to encode domain-specific features, which are then used to translate a sequence from one domain to another in the next stage. The input content sequence, which is a collection of embedded features, is passed through the transformer encoder. The encoder consists of multiple layers, with each layer having a multi-head self-attention module (MSA) and a feed-forward network (FFN). During the encoding process, the input sequence is transformed into query (Q), key (K), and value (V) representations. Similarly, the embedding of an input style sequence is encoded into a sequence Ys following the same calculation process.

- **Transformer Decoder:** Transformer decoder is designed to generate stylized images by translating the encoded content sequence according to the encoded style sequence in a reverse order. Unlike in natural language processing tasks where auto-regressive processes are used, in the current method, the sequential patches are all utilized as input at once to predict the output. The transformer decoder consists of multiple layers, each containing two multi-head self-attention (MSA) layers and one feed-forward network (FFN). The input to the transformer decoder includes the encoded content

sequence and the style sequence. It uses the content sequence to generate the query Q, and the style sequence to generate the key K and value V.

- **CNN Encoder:** The transformer in the model generates an output sequence X with a shape of (HW/64) × C. However, instead of directly upsampling this sequence to obtain the final stylized image, it is refined using a three-layer CNN decoder. Each layer of the decoder expands the scale of the output by performing operations such as 3 × 3 convolution, ReLU activation, and 2 × upsampling. The final result is then obtained at a resolution of H × W × 3.

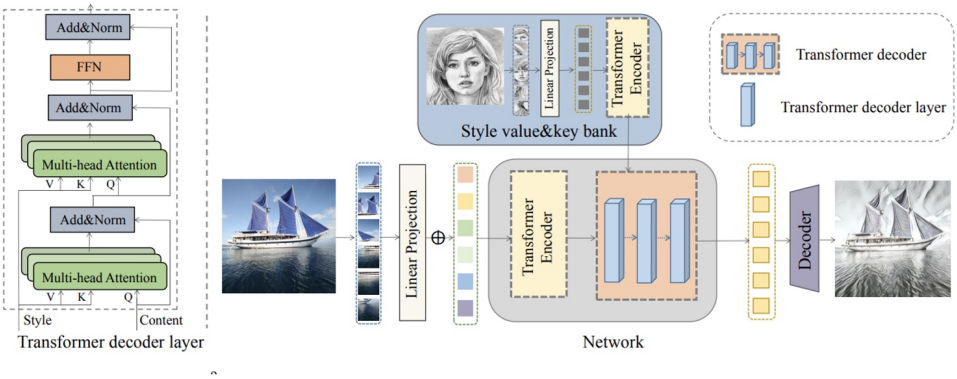

Figure 12: Architecture for Dual Encoder Transformer

8.2 Loss

The objective of the model is to preserve the original content structure and style patterns of the input image. To achieve this, we employ two types of perceptual loss: content loss and style loss. Feature maps extracted from a pre-trained VGG model are used to construct these loss functions. The content perceptual loss (Lc) is computed by comparing the features extracted from the i-th layer of the VGG19 model for the generated image (Io) and the input content image (Ic). The style perceptual loss (Ls) is computed by comparing the mean (μ) and variance () of the features extracted from the VGG19 model for the generated image (Io) and the input style reference (Is).

In addition to the perceptual losses, the model also employs identity loss to learn richer and more accurate representations of content and style. The identity loss ensures that the output image (Icc for content or Iss for style) generated from two identical input images (Ic or Is) is identical to the input image. Two identity loss terms are computed to measure the difference between the generated output (Icc or Iss) and the input image (Ic or Is).

The optimization of the entire network is achieved by minimizing these four loss functions (content perceptual loss, style perceptual loss, content identity loss, and style identity loss).

9. Further Experiments

We conducted two additional experiments to explore alternative methodologies and network architectures:

9.1 Transformer using VGG Based Encoder

Instead of using transformer encoders for the style and content images, which have an attention mechanism, we attempted to use VGG to obtain the content and style feature embeddings and then passed them to the decoder. However, the results were not as good, since we know that attention encoders (transformers) have better contextual representations, even for images.

9.2 Transformer Using CNN Style Encoder

In the second experiment, we replaced the style encoder with a CNN network. Our hypothesis was that a complex model was not necessary to extract the style from style images since we did not use Context-Aware Positional Encoding (CAPE). Therefore, we could replace the style encoder with a CNN to simplify the model, reduce inference/training time, and improve its efficiency. This CNN model was trained using the input style image (x) and the output of the style encoder (y), which extracts the representations/embeddings from the style image. This approach enabled us to learn to map the input style image to style representations using the style encoder, which significantly reduced inference time using the CNN model. However, we observed that this experiment resulted in a greater loss of the style transferred image, despite the reduced inference time. This experiment was conducted in line with the ablation study/distillation, where complex parts of the model are replaced with simpler parts.

10. Results

10.1 Quantitative Evaluation

- **Content Loss** - 1.95

- **Style Loss** - 1.50

10.2 Demonstration web application

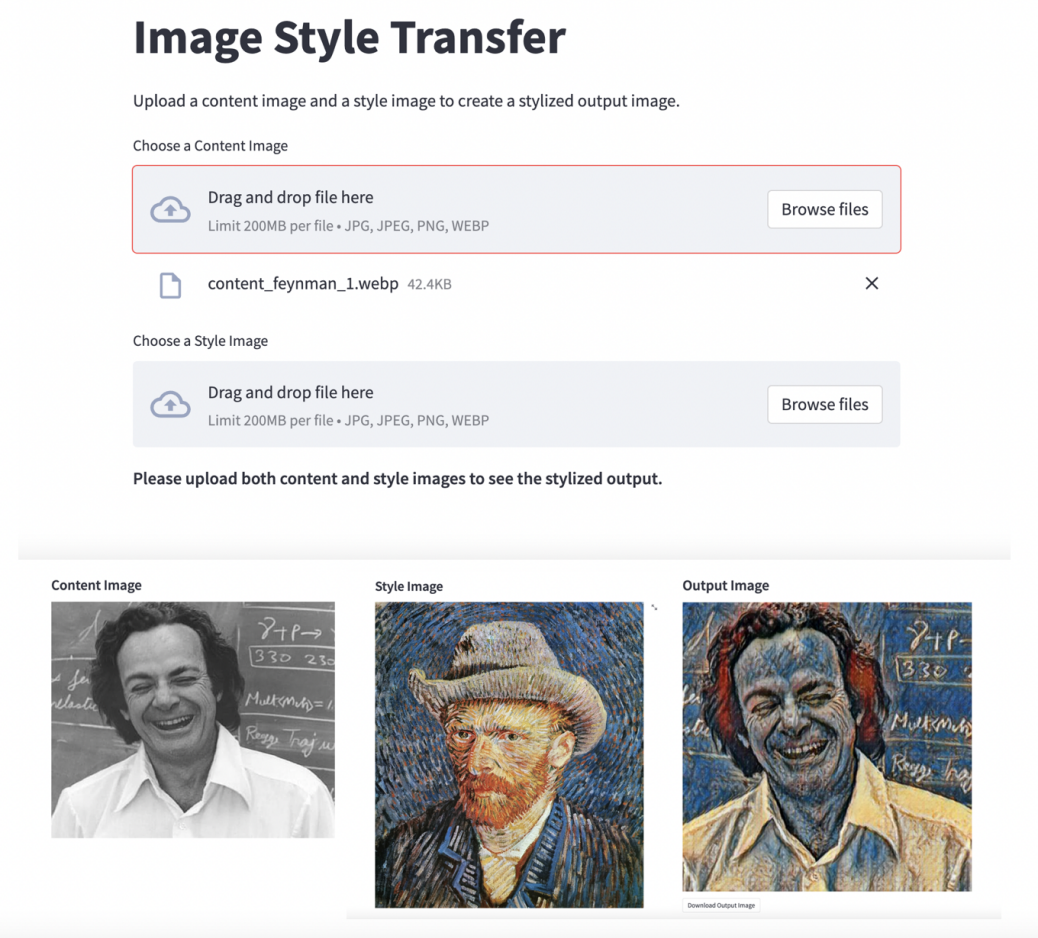

Figure 13: Demo application web interface and example content style and output images

To showcase the effectiveness of our refined dual encoder transformer-based approach for image style transfer and provide an accessible and interactive platform for users to explore and experiment with various artistic styles, we developed a demonstration web application [2]. The application allows users to experience the capabilities of our approach in real-time and evaluate the quality of the generated stylized images.

The web application has a user-friendly interface that enables users to easily select a content image and a style image from their local devices. Upon uploading the content and style images, the application processes the images using our refined dual encoder transformer-based approach for image style transfer. The application then generates an output image for the content image in the style of the selected style image, showcasing the seamless integration of style features while preserving the original content.

2. The application is available at https://saketrule-harvard-cs-final-projectdemostyle-transfer-app-wkdeqk.streamlit.app/1

In addition to the generated output image, the application provides users with the option to download the stylized image in a suitable format, allowing for further exploration or utilization of the output in various applications, such as digital art, content creation, or advertising. The demonstration web application serves as a testament to the efficiency, flexibility, and convenience of our approach in applying various artistic styles to any input image.

By making the results accessible through a web application, we aim to promote further exploration and experimentation within the image style transfer domain, fostering an environment of creativity and innovation. This demonstration web application not only highlights the effectiveness of the refined dual encoder transformer-based approach but also offers an engaging platform for users to experience the power of image style transfer techniques firsthand.

10.3 Result Outputs

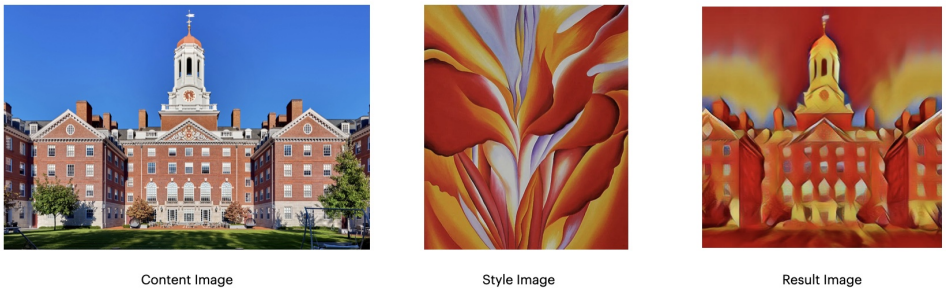

Figure 14: Output for Example Input 1

Figure 15: Output for Example Input 2

11. Conclusion

In conclusion, we have conducted a thorough investigation into various image style transfer techniques and experimented with different model architectures to improve the accuracy

and efficiency of this task. Our research spans from the baseline approach that required optimization for each pair of content and style image to the final model we developed that can seamlessly incorporate any style and content image using just one trained model. We proposed a new framework for image style transfer that consists of a content transformer encoder and a style transformer encoder designed to capture domain-specific long-range information. The framework's transformer decoder is tailored to translate the content sequences based on the reference style sequences, thus representing an improvement over traditional CNN-based models. Our research has demonstrated that the perceptual loss terms and identity loss, combined with a transformer-based model, can effectively enhance the quality of style transfer. We also experimented with various architectures, such as using VGG for feature extraction and replacing the style encoder with a CNN, which helped us determine the most effective methods for achieving the desired outcomes. Overall, our work contributes to the advancement of image style transfer techniques, paving the way for future research in this area. The proposed framework shows promising results in terms of both accuracy and efficiency, opening new avenues for creative and innovative applications of style transfer technology.

12. Future Work

As a future direction, it is worth exploring ways to improve the test-time efficiency of our proposed method, which is currently not as fast as some CNN-based methods. One potential approach could be to incorporate the Bayesian framework using CNN based encoder decoder architectures to speed up the computation without sacrificing performance. Additionally, we intend to investigate how VGG feature embeddings/representations can be utilized to further enhance the accuracy of style and content loss since the VGG network is already used while computing loss for the model we developed. This could potentially lead to better overall performance and improve the model's ability to capture style and content information from input images.

References

Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer, 2017.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1601. URL `https://aclanthology.org/P19-1601`.

Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11326–11336, 2022.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization, 2017.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution, 2016.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. doi: 10.1109/CVPR.2019.00453.

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

Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration, 2018.

Yuan Yao, Jianqiang Ren, Xuansong Xie, Weidong Liu, Yong-Jin Liu, and Jun Wang. Attention-aware multi-stroke style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.