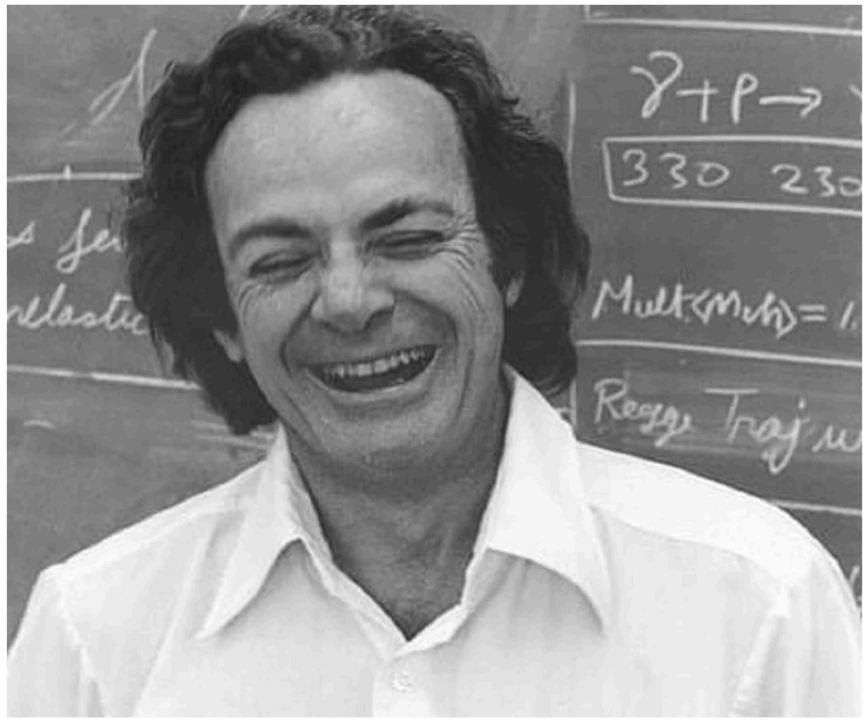
Dual Encoder Transformers for Efficient and Versatile Image Style Transfer

Kareema Batool, Nikhil Nayak, Nishtha Sardana, Saket Joshi

When Richard Feynman met Van Gogh

What if this image of Richard Feynman is painted in the style of Van Gogh's self-portrait?



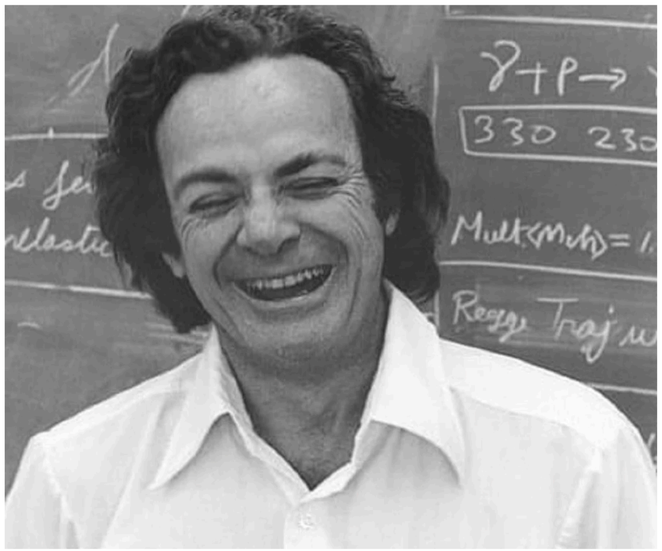
Content Image

Style Image

When Richard Feynman met Van Gogh

What if this image of Richard Feynman is painted in the style of Van Gogh's self-portrait?

Content Image



Photograph of Feynman

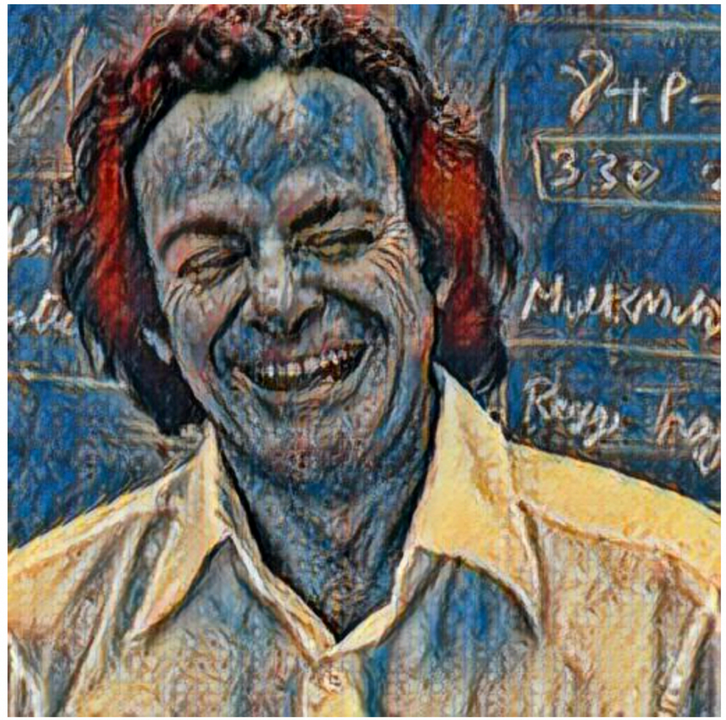
+

Style Image

In the style of

=

Output Image



Download Output Image

Problem Statement

Can you “extract” the style of one image and apply it to another image?

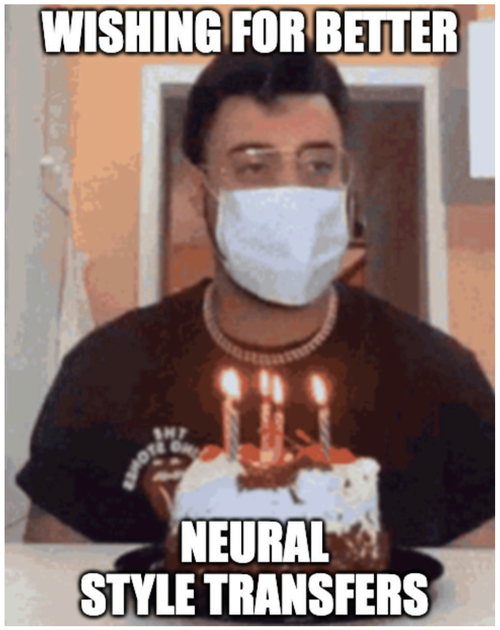
Problem Statement

- 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.

Wishlist:

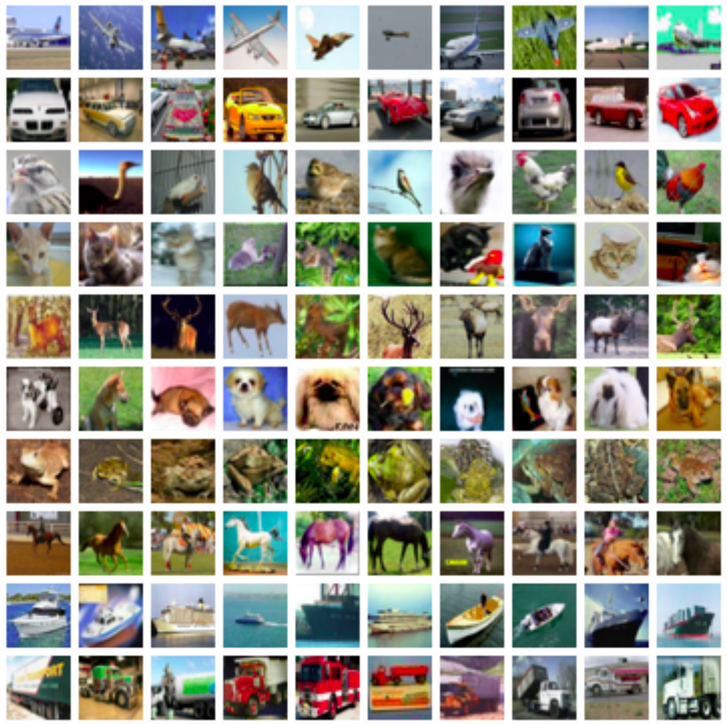
- Efficient and time-saving methods for neural style transfer
- Ability to train a single neural network for multiple styles
- Enhanced flexibility and convenience in the style transfer process



Dataset

Content Images Dataset: COCO (Common Objects in Context)

- Over 330,000 diverse and complex images
- Rich object annotations (categories, bounding boxes, key-points)
- Addresses need for large-scale, diverse dataset



Coco Dataset sample images
Source: https://cocodataset.org/.

Style Images Dataset: WikiArt

- Over 200,000 images of various styles and genres
- High-resolution format, preprocessed for uniform size and aspect ratio
- Metadata for art history or tracking trends

Dataset Collection methodology

- COCO dataset obtained from Kaggle
- WikiArt images downloaded using Python script (GitHub: StyleTransfer/wikiart.py)

EDA

Image Statistics

- Examined pixel intensity statistics and histograms for RGB channels
- Goal was to gain insights into contrast distribution and color characteristics of images in dataset
- Some images having high and low pixel values, indicating diverse brightness and contrast levels.
- Box plots indicate presence of outliers in dataset, stemming from various reasons such as over or underexposure, atypical lighting conditions, or image artifacts.

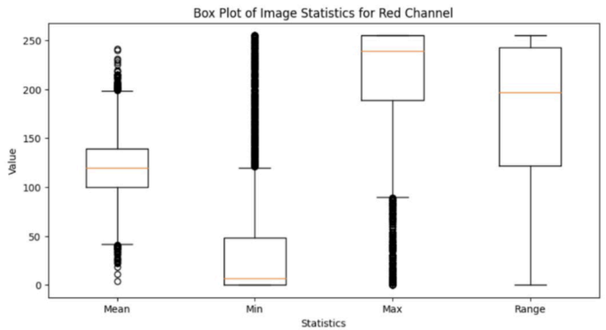


Figure 1: Image Statistics - Red Channel

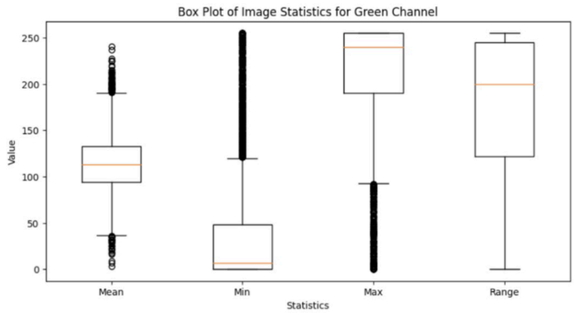


Figure 2: Image Statistics - Green Channel

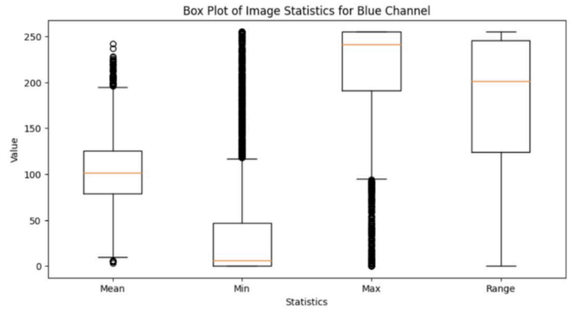
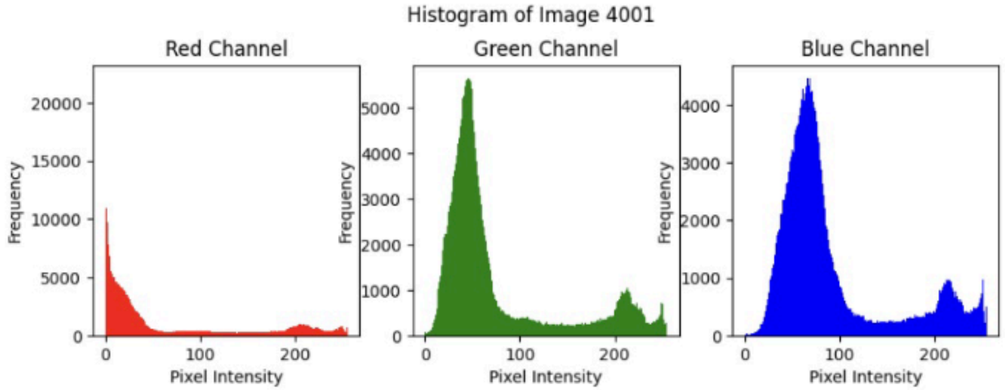


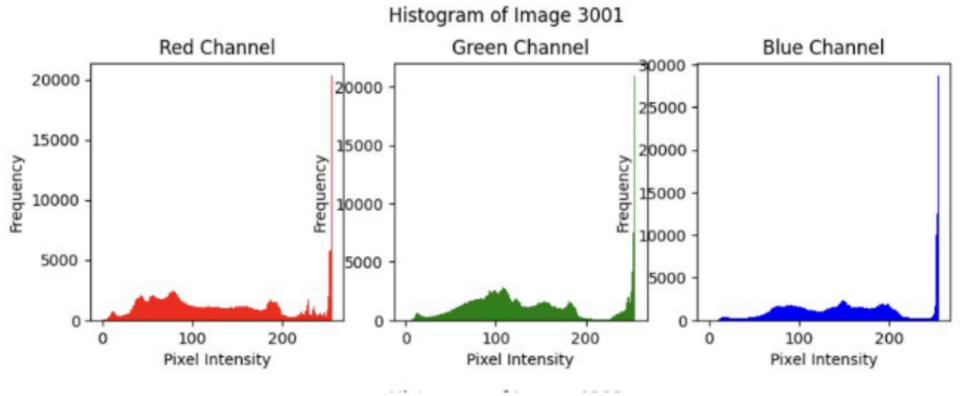
Figure 3: Image Statistics - Blue Channel

EDA

Image Histograms

- Analyzed histograms for several images
- Diverse range of images, some with a concentration of pixel values in the bright region and others in the dark region.
- The well-spread histogram of pixel values also suggests that the contrast of the images is adequate, leading us to conclude they are well suited for the task at hand.





Pre-Processing

Mean Imputation → Image Pyramids → Data Normalization /Standardization → Data Augmentation


MODELS
MODELS EVERYWHERE

Baseline Model

- Uses VGG network to extract style and content from image
- Optimizer back-propagates and updates pixel values of generated image using gradient descent

Shortcoming:

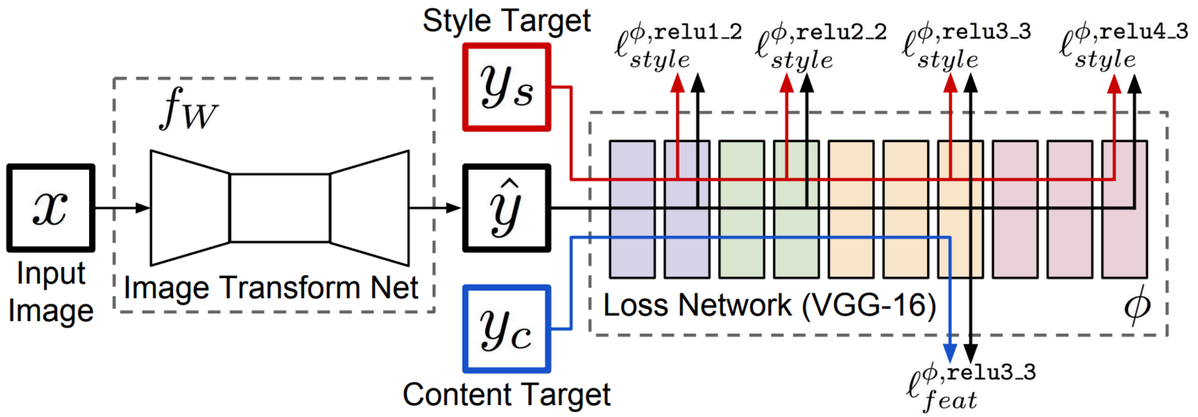
- Slow process not practical for styling multiple images (have to run the optimization method for each unique pair of content and style image)

Fast Neural Style Transfer Model

- Uses an auto-encoder network to learn how to apply a specific style to any content image
- Uses VGG to calculate style and content loss

Shortcoming:

- Can only be used for a single style image. To apply another style have to retrain the network



Baseline Model

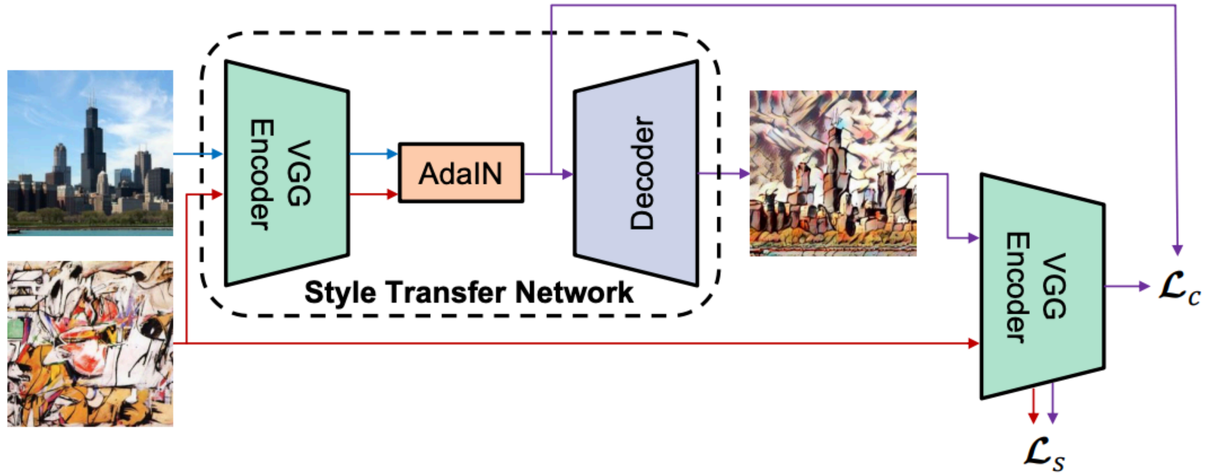
Fast NST

CNN Based Encoder Decoder

- Uses CNN based encoder decoder network to extract content and style from any pair of images to generate the stylized image
- Encoder and decoder use CNN and FCNN layers

Shortcoming:

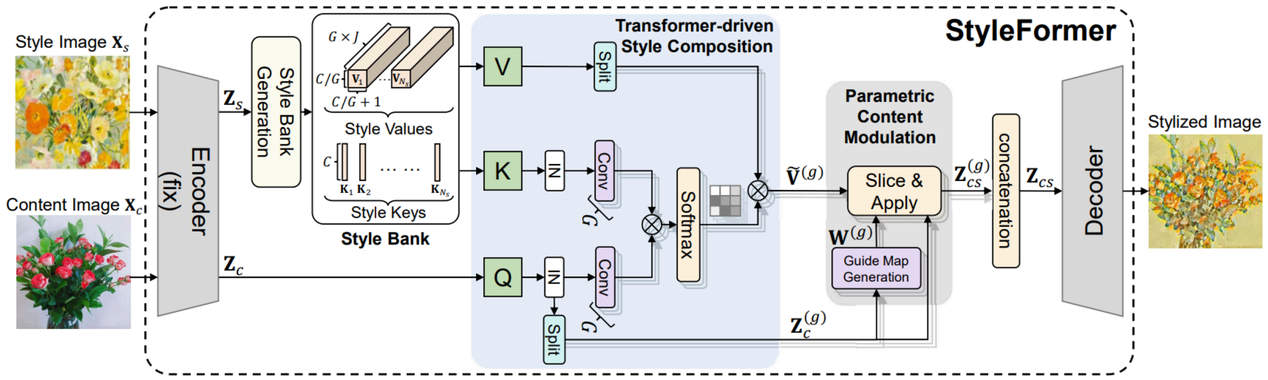
- Content leak in stylized image
- Inability to capture long range dependencies due to the limited receptive field of convolution operation -> motivating the need for a faster and more versatile approach.
- Causes loss of feature resolution and fine details



Attention is All you Need!

Transformer with One Encoder

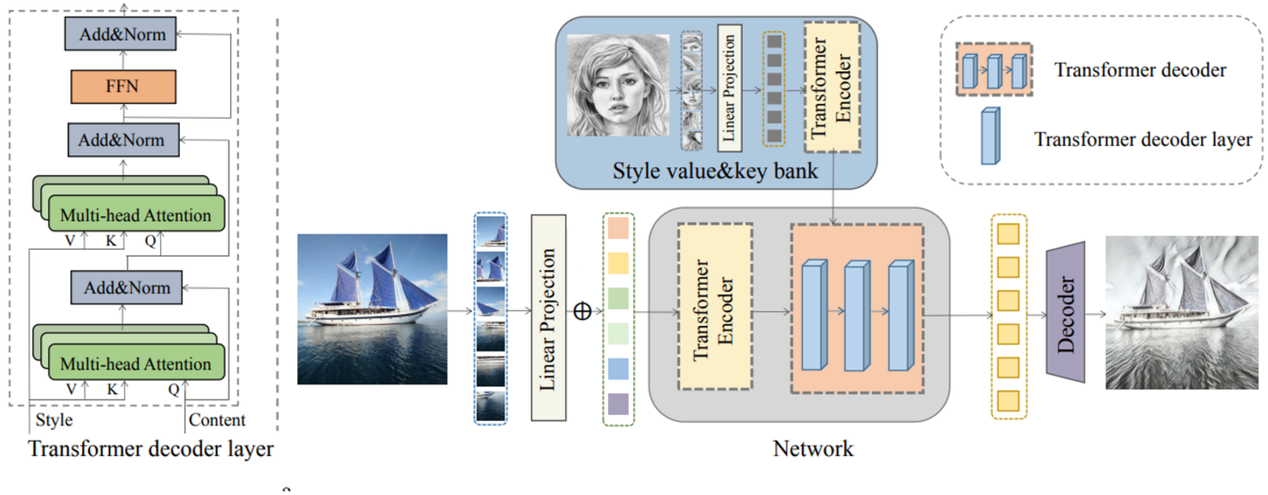
- Uses self attention mechanism to better capture the contextual dependencies.
- Solves the problem of limited receptive field seen in CNNs due to access to unlimited context.



Baseline Model | Fast NST | CNN Based Encoder Decoder | Tranformer w/ One Encoder

Transformer with Dual Encoders

- Dual encoders one for content and second for style.
- Able to extract better representations
- Single decoder that takes as input both encoder outputs and generates a consistent, high quality stylized image
- 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.



Baseline Model

Fast NST

CNN Based Encoder Decoder

Tranformer w/ One Encoder

Transformer w/ Dual Encoder

Training Stats

Given the significant computation requirements, We adopted the following training schedule -

1 - We save models, and metrics through a callback
2 - Train `n` epochs, manually evaluate the loss and output quality
3 - Decide to continue training

- **Time Taken:** Approx. 6 hours on JupyterHub GPU
- **All experimentation training time:** Approx. 24 hours
- **Dataset:** COCO (Content Dataset), WikiArt (Style)
- **Number of Epochs:** 18,750 iterations (~150 epochs)
- **Batch Size:** 8
- **Optimizer & Learning Rate:** Adam with 0.0005

Results - Comparison of architectures

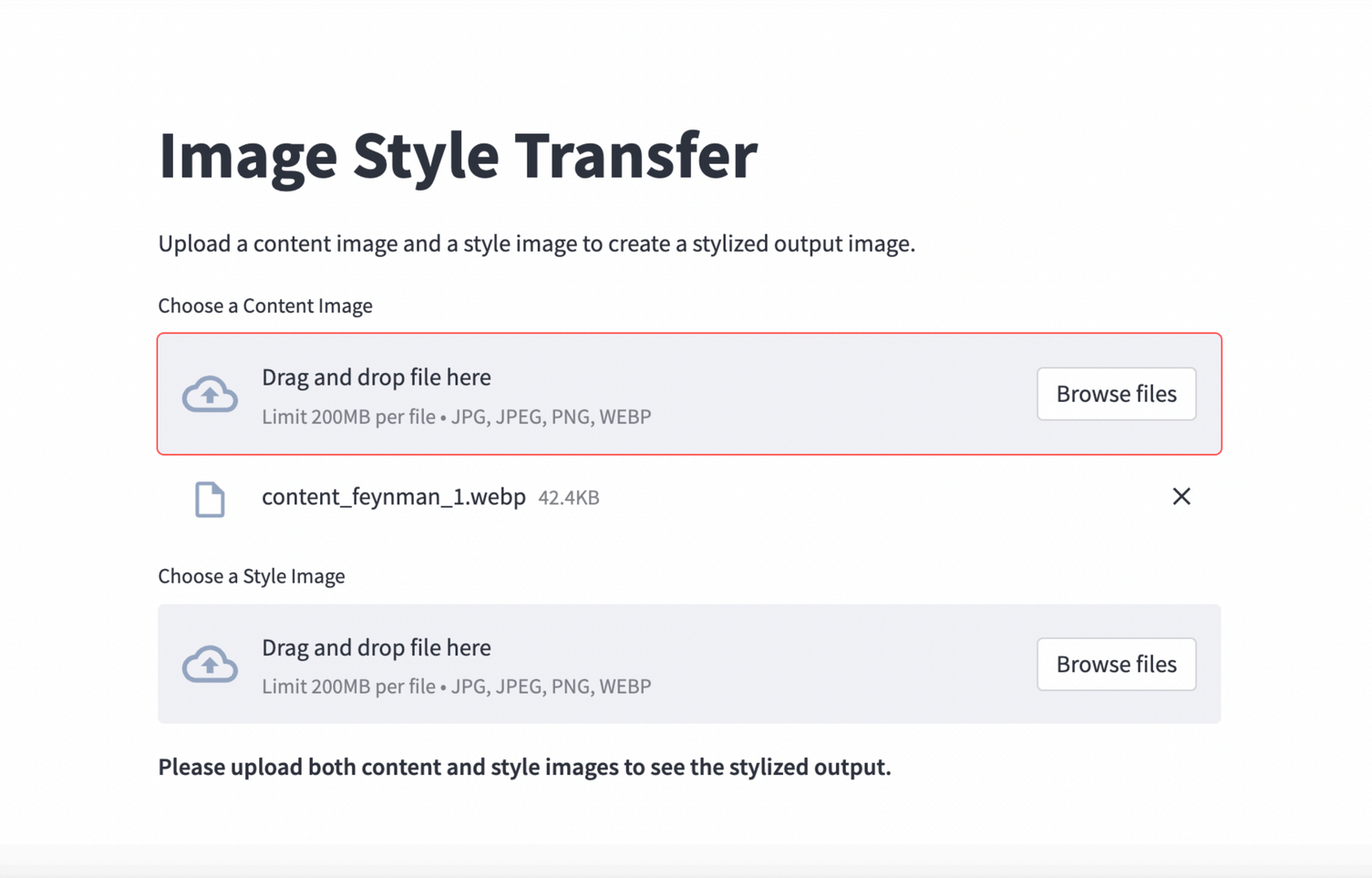
[Arch 1] - Transformer based Content encoder with CNN Style encoder Network

1. Experiment involved replacing the style encoder with a CNN to simplify the model, reduce inference/training time, and improve its efficiency.
2. Results showed that although inference time was significantly reduced, the loss of the style transferred image was greater, suggesting that the complex transformer encoder was necessary for better results.

[Arch 2] Transformer based Style and Content encoder Network

1. We get a content loss of 1.95 and a style loss of 1.50. This indicates that the model performs well in maintaining the content as well as learning the style of the image.
2. We get better performance with the trade-off of significantly higher computation time
   This model is made available and demonstrated via the web application

Results - Web application



Available at - https://saketrule-harvard-cs-final-projectdemostyle-transfer-app-wkdeqk.streamlit.app/1

Results - Example output images

+

=

Results - Example output images

+

=

Results - Comparison of architectures

[Arch 1] - Transformer based Content encoder with VGG Style encoder Network

VGG is used for content and style feature embeddings instead of transformer encoders.
Results were not satisfactory in output image qualities
We attribute the inferior performance to missing contextual representation compared to attention encoders.

[Arch 2] - Transformer based Content encoder with CNN Style encoder Network

Experiment involved replacing the style encoder with a CNN to simplify the model, reduce inference/training time, and improve its efficiency.
Results showed that although inference time was significantly reduced, the loss of the style transferred image was greater, suggesting that the complex transformer encoder was necessary for better results.
This model demonstrated better performance than VGG style encoder network

[Arch 3] Transformer based Style and Content encoder Network

We get a content loss of 1.95 and a style loss of 1.50. This indicates that the model performs well in both maintaining the content as well as learning the style of the style image.
We get better performance than both models with the trade-off of significantly higher computation time
This model is made available and demonstrated via the web application

Conclusion/ Inferences

- Thorough investigated image style transfer techniques
- Experimented different model architectures for accuracy & efficiency improvements
- New framework using content and style transformer encoders (dual encoders) and tailored transformer decoder
- Identity loss terms combined with Perceptual (VGG-based) loss in transformer-based model to enhance quality of style transfer
- Experimented with VGG and CNN-based models for faster inference

Future Work

- Explore ways to improve test-time efficiency of the proposed method
  - Currently not as fast as some CNN-based methods
  - Incorporate Bayesian framework using CNN-based encoder-decoder architectures to speed up computation without sacrificing performance

- Investigate utilization of VGG feature embeddings/representations to enhance accuracy of style and content loss
  - VGG network already used while computing loss for developed model
  - Potential for better overall performance and improved ability to capture style and content information from input images