Neural Canvas
Text-to-Art Image Stylization With Neural Style Transfer

Aarav Kumar, Angel Alvarado Reyes, Brian Song, Sophia Huang.
Brown University

Abstract

In this project, we present an approach to neural style transfer that allows users to specify artistic styles through natural language descriptions. Our system combines the power of CLIP (Contrastive Language-Image Pre-training) for text-to-image matching with traditional neural style transfer techniques based on VGG19 architecture. Unlike conventional style transfer methods that require direct style image selection, our approach enables users to simply describe their desired artistic style in words, making the technology more intuitive and accessible to non-expert users. The system maintains a curated database of 400 popular paintings and uses CLIP embeddings to find the most semantically similar artworks to the user's text description. These selected styles are then applied to the user's content image using a modified neural style transfer algorithm that can combine multiple style influences. Experimental results demonstrate the effectiveness of our approach in translating textual style descriptions into visually coherent artistic transformations.

1. Introduction

Neural style transfer has revolutionized digital art creation, but current implementations often limit users by requiring them to manually select specific style images. This creates a barrier for those who may not have extensive art knowledge or access to suitable reference images. Our project addresses this challenge by introducing a text-guided neural style transfer system that bridges the gap between natural language description and artistic transformation. By combining the powerful image feature extraction capabilities of VGG-19 with OpenAI's CLIP model for text-image similarity, users can simply describe their desired artistic style in plain text, and our system automatically selects and applies a weighted combination of the most relevant artistic styles.

Our approach not only makes artistic style transfer more accessible but also expands creative possibilities by combining multiple style influences based on natural language descriptions. This innovation has significant implications for democratizing artistic tools, enabling both casual users and artists to achieve their creative vision through simple text descriptions rather than manual style image selection.

2. Related Work

Our work builds primarily upon the seminal work by Gatys et al. [1], which introduced a groundbreaking approach to separating and recombining the content and style of arbitrary images using Convolutional Neural Networks (CNNs). Their key insight was that the feature responses in different layers of a CNN trained on object recognition could be used to independently manipulate content and style. Following their findings, we utilize their suggested layer selections from the VGG-19 network, specifically using 'conv4_2' for content representation and 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1', 'conv5_1' for style representations.

The Gatys paper demonstrated that higher layers in the network capture the high-level content in terms of objects and their arrangement, while style can be captured through correlations between different filter responses within each layer. We adopt their approach of using Gram matrices to represent these style correlations and their loss function architecture that balances content and style reconstruction.

While Gatys et al. [1] focused on single style transfer between two images, we extend their approach by incorporating multiple style images weighted by their relevance to user-provided text descriptions. This extension is enabled by integrating OpenAI's CLIP model for text-to-image matching, allowing us to automatically select appropriate style images based on natural language input.

3. Method

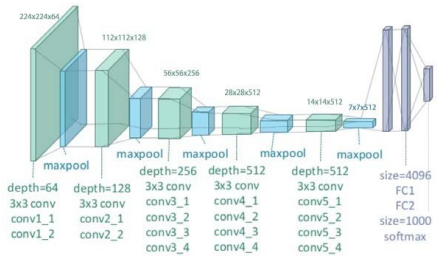
First, we compiled a dataset of style images. Using the MET Collection API, which gives access to MET Museum artwork without restrictions, we downloaded 379 "highlighted" paintings featuring different art styles and time

periods. It was important to us to find a dataset that was not limited to just traditional Western artworks to ensure that our model could account for a range of inputs and styles. The MET collection offered artworks from diverse cultures, time periods, and artists (not just White European men), which was ideal for our project.

Following this, for each of the style images in the collection, we created image embeddings using OpenAI CLIP's image encoder. CLIP embeddings are vector representations that align text and image data in a shared "semantic" space that allows us to compare their meanings. The image encoder generates vectors for images, and the text encoder generates vectors for text descriptions. In this shared space, we can use similarity metrics (we used cosine similarity) to measure how similar the meaning of a style image is to a text description.

Then, we implemented a simple neural style transfer model with CLIP embeddings. From our literature review, we found that many previous NST papers have found success using the VGG-19 model. Therefore, we chose to use the VGG-19 model, which was pretrained on the ImageNet database with over a million images of 1000 object categories.

The architecture of the VGG-19 model is shown below.



[4]

Following Gatys et al. [1], we use their core loss functions while extending them to handle multiple style images. The total loss for our multi-style transfer is given by:

$$L_{total}(\vec{p}, \{\vec{a}_i\}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}) + \beta \sum_{i=1}^{N} w_i L_{style}(\vec{a}_i, \vec{x}) \tag{1}$$

where the content and style losses are computed as:

$$L_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} (F_{ij}^l - P_{ij}^l)^2 \tag{2}$$

$$L_{style}(\vec{a}, \vec{x}) = \sum_l \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2 \tag{3}$$

Our key modification was the introduction of CLIP-derived weights w_i in the total loss function, allowing us to weight multiple style images based on their semantic similarity to the user's text description. The content loss measures feature reconstruction quality at layer l, while the style loss compares Gram matrices of feature correlations across multiple layers, with N_l and M_l representing the dimensions of the feature maps in layer l.

We also wanted to experiment with different methods to actually apply the style transfer to see how they would differ (both qualitatively and quantitatively, i.e. in terms of time taken). In particular, we decided to experiment with three different methods—individual, weighted, and chained.

The individual style transfer method applies one style at a time to a given content image. That is, for each text description provided, the model will find the closest style image from our database, and apply them one by one to the content image. In this method, the styles are merged uniformly with equal weight (i.e each style contributes 1.0 weight) to produce a single optimized image. We decided to experiment with this approach as we thought it would be ideal for isolating the impact of a specific style or comparing the results of different styles applied separately.

The weighted approach, on the other hand, blends multiple styles into a single output based on weights determined by their similarity to the user-provided text description(s). In this method, for each description, the model retrieves the top matching style images and their text-alignment scores (normalized into weights representative of the relative influence of each style), and applies the styles to the content image based on these weights. Thus, during optimization, the style features and Gram matrices from all the selected styles are combined proportionally to these weights, creating a fusion of the styles. We thought this method would be useful for generating images that capture the essence of multiple artistic styles to create more nuanced results. Additionally, we thought this method would also be useful when a particular text description is extremely obscure, or describes something that is unlikely to be in our database — in this case, this method will give higher weight to the styles that have better text-alignment scores, creating more reasonable results.

Finally, we implemented the chained approach, which applies styles sequentially to progressively transform the content image (like a recursive neural style transfer). In this method, the top style selected (based on the inputted text-descriptions) is applied to the content image, and the output of this step is used as the content image for the next iteration, where the 2nd top style is applied, and so on (till the desired number of styles are applied). We thought that this iterative approach would layer different stylistic influences, with each step building upon the transformations of the previous one, capturing the cumulative effect of multiple styles applied in sequence.

For implementing the neural style transfer algorithm, we

carefully tuned several hyperparameters to achieve optimal results. The optimization process runs for 300 iterations using the Adam optimizer with a learning rate of 0.3. We found that setting the content weight to 1e-10 and style weight to 1e8 produced the most visually appealing balance between preserving the content structure and applying artistic style.

To improve computational efficiency, we also implemented several model optimization techniques, primarily experimenting with model quantization and pruning. The quantization approach reduces the precision of model weights and activations from 32-bit to 8-bit, attempting to decrease memory usage and make our computation faster without significantly affecting model performance. Pruning, on the other hand, removes less important filters in the model based on their L1 norm— that is, the filters contributing the least weights are excluded from the CNN, simplifying the network to reduce the time taken. We also explored combining both techniques to maximize performance gains.

Beyond this, we also experimented with alternative neural network architectures specifically designed for efficiency: EfficientNet and MobileNet. Unlike VGG, these architectures are optimized for lightweight computation. MobileNet uses depthwise separable convolutions to reduce the number of parameters [2], and EfficientNet scales the depth, width, and resolution of the network using a compound scaling method to balance accuracy and efficiency [3]. We decided to use these networks to explore whether a more computationally efficient backbone could perform style transfers faster without reducing the quality of the output image.

The text-to-style matching process utilizes CLIP embeddings in a two-stage approach. First, we pre-compute embeddings for all 379 artworks in our database using CLIP's image encoder. When a user provides a text description, we encode it using CLIP's text encoder and compute cosine similarities between the text embedding and all image embeddings. The top-k most similar artworks are selected as style images, where k varies based on the chosen method (individual, weighted, or chained). For the weighted approach, the similarity scores are normalized to create weights that determine each style's contribution to the final result.

Image preprocessing plays a crucial role in our pipeline. Input images are resized to 512x512 pixels and normalized using ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]). The output images undergo inverse normalization and are clamped to valid pixel values before being converted back to viewable format.

To make our system accessible to users, we developed a web interface that accepts content images and text descriptions as input. We used JavaScript and React to create the user interface. When a user submits a content image and a style prompt (can be multiple keywords), this gets sent through Flask as input to our NST Model in a Jupyter Notebook, which then runs on Google Colab. Once the style transfer is complete, the target image is then sent back and displayed on the UI. Because of our CLIP-embedding approach to choose styles, this allows users to easily experiment with different, abstract style descriptions without needing to understand the underlying technical implementation. Here is a link to our web demo: demo link.

4. Results

The results for our different neural networks are presented in Table 1 and Figure 1.

Model Used	Time Taken (s)
VGG19	48.3
VGG19 + Quantization	47.2
VGG19 + Pruning	44.2
VGG19 + Quantization + Pruning	43.7
EfficientNet	23.5
MobileNet	10.7

Table 1. The time taken for each model in applying the styles, given two text inputs: "Van Gogh style trees swirly", and "Rainbow".

Figure 1. The content image, and the output for the VGG model (top left), VGG + Quantization (top middle), VGG + Pruning (top right), VGG + Quantization + Pruning (bottom left), MobileNet (bottom middle), and EfficientNet (bottom right) given the text input in Table 1.

From our results, we can see that both quantization and pruning improved the efficiency of our VGG model, reducing the time taken by a couple of seconds individually, and by a total of 4.6 seconds when applied together. In terms of efficiency, it seems that pruning had more of an effect than quantization did, as quantization seemed to only reduce the time taken by 1.1 seconds.

Qualitatively, we found though pruning improved efficiency, the results were not as visually accurate, as seen in Figure 1. In both the cases where pruning was used, the patterns on the resultant image appeared pixelated, and while the image appeared stylized, it didn't really match the text description (the "Van Gogh trees" did not really appear, neither did the "Rainbow" colors, and it seems the image maintained much of the colors and structure from the content image). With quantization, however, the text image seemed more aligned with the text description, as the output appeared more colorful and lots of swirls and greens were added to the image (likely resemblant of "Van Gogh trees").

Figure 2. The content image (top left), and the output for the individual (top right), weighted (bottom left), and chained methods (bottom right), given the text input in Table 2.

NST Method	Time Taken (s)
Individual	49.3
Weighted	48.7
Chained	97.4 (48.8 + 48.6)

Table 2. The time taken for each approach to applying the styles, given two text inputs: "Van Gogh swirls", and "Impressionistic colors". The chained case has a sum of two times, as 2 styles were applied recursively (rather than together). The VGG-19 model was used, without quantization or pruning.

For the comparisons between the individual, weighted, and chained approaches of style transfer we implemented, the results were a little surprising. A table of the quantitative (time) results (Table 2.) and output images (Figure 2) for each of the methods is included below.

Evidently, the time taken in each approach was not much different from the original VGG approach (expect for the chained case, which was expected, as the style transfer happens over multiple iterations), however, it was the qualitative results that surprised us. It seems that the individual and weighted style transfer don't have much of a difference in terms of the output produced like we originally expected. While it does seem like the weighted approach was more sensitive to the text description (the "Impressionistic colors" seemed to be more present in the weighted output), the overall images seemed very similar. Additionally, for the chained approach, the output image appeared quite blurry and became almost unrecognizable from the content image, with none of the styles appearing distinctly either.

Finally, we also tested comparing the weighted style transfer of multiple styles to the single application of each style, to see how effective the weighting of styles was. the results are shown in Figure 3.

Figure 3. Weighted model output for text descriptions ["Ocean Blue", ","] (left), and individual outputs for description "Ocean Blue" (center), and "," (right)

From this, we can see that the weighted approach does have an impact on the final output, particularly in selecting styles that are most relevant and closely aligned with the text input. For instance, the output generated from a single, vague text input (a comma-separated description, as shown in the right image) appears unclear and lacking in distinctiveness. In contrast, the output generated from the text input "Ocean Blue" is much more representative of an artistic style with blue, wavy (ocean-like) qualities. When combining both text descriptions using the weighted approach, the output demonstrates a clear shift towards the style corresponding to the "Ocean Blue" input, indicating the it places emphasis on styles that have stronger similarity scores (weights) to the text input.

4.1. Technical Discussion

Our exploration of neural style transfer methods revealed several interesting technical trade-offs and challenges. Most notably, our experiments with model optimization techniques demonstrated a clear inverse relationship between compu-

tational efficiency and output quality. While quantization provided modest speed improvements with minimal quality loss (reducing processing time from 48.3s to 47.2s), pruning showed more dramatic effects. Although pruning achieved greater efficiency gains (reducing to 44.2s), it significantly compromised the visual fidelity of the outputs, producing noticeably pixelated patterns and failing to properly capture the intended styles. This suggests that many of the filters identified as less significant by L1 norm pruning may actually be essential for preserving subtle artistic features.

The substantial performance differences between network architectures raised fundamental questions about feature extraction in style transfer. Despite MobileNet's impressive speed improvements (10.7s compared to VGG19's 48.3s), and EfficientNet's balanced performance (23.5s), the quality variations in their outputs suggest that architectural efficiency might come at the cost of style representation capability. This raises an important question about the minimal network complexity required for effective style transfer, and whether certain architectural components of VGG19 might be essential for capturing artistic features.

Our implementation of multiple style combination methods produced unexpected results that challenge conventional assumptions. The similar outputs between individual and weighted approaches suggests that the network's ability to blend styles might be more limited than theoretically predicted. More surprisingly, the chained approach, despite its intuitive appeal, resulted in significant degradation of both content and style elements. This degradation increased with each successive style application, indicating that the optimization process might not be well-suited for sequential style transformations.

The integration of CLIP embeddings for style selection introduced its own set of technical considerations. While the system successfully matched text descriptions to relevant styles, its performance varied considerably based on the specificity and artistic vocabulary used in the text prompts. This highlights the challenge of bridging the semantic gap between natural language descriptions and visual style features, particularly for abstract or complex artistic concepts.

These findings suggest several avenues for future research. There's a clear need to develop more sophisticated pruning strategies that can preserve artistic feature extraction capabilities while reducing computational overhead. Additionally, investigating hybrid architectures that combine the efficiency of lightweight networks with the style representation capabilities of deeper networks could provide a more optimal solution. Finally, developing more nuanced methods for style combination could help realize the full potential of multi-style transfer.

5. Conclusion

Neural style transfer is an interesting area of intersection of computer vision and artistic creativity, with extension in digital art generation. In our final project, we aimed to develop a flexible, accessible neural style transfer system that bridges natural language description and style transformations. We were also interested in exploring the preservation of style quality while aiming to improve computational efficiency.

Our core contribution based on previous work is integrating the OpenAI CLIP into our style transfer pipeline with the VGG19 model and traditional neural style techniques. Through text-image embedding comparisons, the most similar style images can be matched with user-generated prompts. This enables users to specify multiple artistic styles of their choice and provides an approachable introduction to Neural Style Transfer to those without specific technical exposure.

We explored three multi-style integration methods - individual, uniform weighting, and sequential chain transfer - with a content image. Contrary to our initial hypotheses, the weighted and individual models generated similar outputs. This suggests that models may be inherently constrained in the variance of potential style combinations. We anticipated the chaining approach to improve style layering and quality, however we found that the chained images were blurry.

We were also interested in methods to improve computational efficiency. The exploration of model optimization techniques - quantization, pruning, and lightweight models like MobileNet and EfficientNet - exposed the trade-off between speed and image quality. While pruning and quantization slightly reduced time, they simultaneously lowered the style integration of the images. The balance between computational efficiency and style-content integration is a challenge that will likely motivate of future research on Neural Style Transfer, especially for more computationally-expensive applications.

Finally, we created a web application that would allow users to transform images based on their own style input. This relates to the broader social implications for digital creativity and computer vision. The addition of CLIP introduced an aspect of personalization from the user's end, which is more interactive. This can be provided as an interesting and approachable tool for individuals with limited background to interact with technical concepts like Neural Style Transfer.

These findings provide insight into the complexity of creating Neural Style Transfer models to be both fast and effective. This is critical as limits in computational resources are factors that constrain the current applications of Neural Style Transfer. While traditionally Neural Style Transfer has been applicable to images and videos, more efficient models could broaden application to more demanding tasks like real-time video style transfer. Another future area of investigation is the development of further advanced neural

network architectures that can intelligently interpret and combine artistic styles. By developing more sophisticated and efficient neural style transfer techniques, this introduces new possibilities and challenges associated with creativity and artistic generation.

References

[1] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 1, 2

[2] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 3

[3] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019. 3

[4] Yufeng Zheng, Clifford Yang, and Alex Merkulov. Breast cancer screening using convoluional neural network and follow-up digital mammography. *Proceedings of Computational Imaging III*, 2018. 2