Multiple Style Transfer with Transformers

Anonymous Authors[1]

Abstract

The objective of style transfer is to render the stylistic features in a style reference image onto a different image while maintaining the original content. Traditionally, style transfer was accomplished by using convolutional neural networks (CNNs). However, using CNNs results in many drawbacks such as biased content representation, not being scale invariable, and suffering from content leak. To fix this, transformers traditionally used in NLP programs have been adapted for style transfer. As an extension, the ability to add multiple style images and vary the weighting when blended together has been added to show other potential uses of style transfer.

1. Introduction

The goal of this project was to extend the idea of style transfer that has been present in the computer vision industry. First, I began with a recent paper on the advancements in style transfer in which the authors developed a method to accomplish style transfer with transformers. Transformers are commonly used in natural language processing problems and are not traditionally used in style transfer, which uses convolutional neural networks. I began looking at other papers to see what I could do with the ideas of style transfer and transformers.

The first paper I review introduces the concept of style transfer with transformers, which is the main project and the base of code which my extension depends on. The second paper is ArtFlow: Unbiased Image Style Transfer which aims to solve the content leak problem found in convolutional neural networks in a different method: with reversible neural flows. This paper exposes me to other approaches to style transfer. Finally, the third paper is Text to Image Generation which uses style transfer and transformers in the natural language processing (NLP) application to generate artwork. This paper shows the logical extent of transformers in which the NLP algorithms can bypass the need for reference images. With this, a prompt is used instead to generate images. These three papers all provide ways in which style transfer can be used for artistic purposes.

For my extension, I wanted to demonstrate other ways style transfer could be used in art. To do this, I implemented a way to have multiple styles be blended together to be used for style transfer. This means that two unrelated style images could be put together to create interesting artworks. Furthermore, I added optional weighting parameters to each style image so that the final image can have different levels of contribution from each style image. This overall allows for interesting new imagery to be generated and could serve as inspiration for artists.

2. Background

Style transfer is a form of computer vision that manipulates digital images or videos to adopt the appearance or "style" of a different image. These algorithms use deep neural networks to take a content image that will be modified and a style image to be used as a reference for what "style" to apply to generate a new, combined image. Mainly, style transfer is used for artistic purposes to create new artwork based on existing images, and can be used for artistic inspiration and creating interesting imagery.

For setup, the style transfer algorithm will take a content image which will be the image that will be modified and a style image which will be the image in which its features are extracted and imprinted onto the content image. During this process, there are two important parameters called content loss and style loss. Content loss is the difference between the generated image and the content image and is typically measured by comparing feature maps of the model. Style loss measures the difference between the style of the generated image and the style image, and is often calculated by comparing the Gram matrices of feature maps to capture texture and visual patterns.

3. Review of paper: $StyTr^2$: Image Style Transfer with Transformers

3.1. Storyline

High-level motivation and problem Ultimately, style transfer is very computationally intensive. Using neural networks requires lots of hardware power through heavy RAM, CPU, and GPU usage. Thus, to generate images in large projects using style transfer will be very expensive.

Furthermore, the images produced are not always high quality. After many iterations of applying style transfer, the style's features will gradually begin to overtake the image, creating a very jumbled and hard-to-comprehend picture. The two metrics you can use to measure image degradation is content loss and style loss. Content loss is difference between the features extracted from the inserted content image and the outputted generated image. To the contrary, style loss is the measurement of the style in the output image's resemblance to the style reference image. The goal of this paper is to minimize content and style loss which as a result will produce clearer images.

Research Gap Current style transfer methods using convolutional neural networks (CNNs) face issues with biased content representation and lack content-aware positional encoding. The first issue can be resolved with a transformer-based approach to generate domain-specific sequences for content and style. Transformers also will be scale invariant and have content aware positional encoding, meaning that it is more applicable to image style transfer tasks compared to CNNs. This is because CNN generated images lack a more holistic understanding of the image and merely look at the neighboring areas of the image due to the nature of convolution. Since style transfer has not been accomplished with the usage of transformer encoders, this paper aims to implement a style transfer method using transformers.

Prior work on this problem Style transfer had a breakthrough in 2015 by using Convolutional Neural Networks (CNNs) which introduced an optimization-based approach to separate and recombine the style of these arbitrary images. It demonstrated that CNNs, in particular the VGG network, could be employed to extract the features that defined the style of the image, creating a method to manipulate and merge images to match another image's style. This was achieved by defining loss functions that measure the difference in content and style between images, which could be minimized using gradient descent.

Style transfer further advanced with the incorporation of Generative Adversarial Networks (GANs). Researchers used GANs to generate stylized images in a more computationally efficient manner compared to optimization. Some GANs such as CycleGAN, enabled unpaired image-to-image translation, meaning that content did not need to be paired together for image-to-image translation, eliminating the need for aligned pairs during training. This opened up new possibilities such as style transfer from summer to winter and night to day.

Style transfer also has aspirations towards videos, as this method could be used to assist artists in animation and video graphics. Style transfer for videos is still in its infancy and has a lot less research compared to still images.

Contributions With the recent success of using transformers in natural language processing (NLP), the paper aims to apply transformer-based models to style transfer. Since style transfer is typically done with CNN-based models, this provides a novel method to achieve style transfer. The researchers claim that using transformer-based models can prevent many of the weaknesses of using convolution such as preventing artifacts and lowering the amount of content detail loss and prevent biased representation. The paper provides a novel image framework called Style Transfer Transformer or $StyTR^2$. $StyTR^2$ contains two transformer-based encoders in the framework to obtain domain-specific information. The paper provides three main contributions: (1) image sequence tokens are associated with semantic information of image content instead of sentences ordered by logic. (2) For style transfer, the transformer-based model uses content-aware positional encoding scheme that is scale invariant that allows for images of different resolutions to be modified. (3) Empirical evidence that $StyTR^2$ outperforms other style transfer methods and achieves results with desirable content structures and style patterns. (Deng et al., 2022)

3.2. Proposed solution

This paper will fill the research gap because it presents an alternative method to achieving style transfer. This paper moves away from the traditional method of using CNNs to learn style and instead uses transformers commonly found in LLMs. This allows for learning global information of the input with the help of a transformer's self-attention mechanism which means it can obtain a holistic understanding within each layer. The transformer architecture also models relationships in input shapes, and different layers will extract similar structural information. This means the transformer is very capable of capturing precise content representation and avoids missing the finer details in images. As a result, the integrity of the images can be well-preserved.

The paper presents its transformer-based model as Content-Aware Positional Encoding (CAPE). Given a content image $I_c \in \mathbb{R}^{H \times W \times 3}$ and a style image $I_s \in \mathbb{R}^{H \times W \times 3}$, the images are split into patches and use a linear projection layer to project input patches in a shape of $L \times C$ where $L = \frac{H \times W}{m \times m}$ is the length of $\mathcal{E}$, $m = 8$ is the patch size and C is the dimension of $\mathcal{E}$.

With CAPE, the algorithm will be conditioned on the semantics of image content, meaning it cares about what objects or features are present in different parts of the image and not just their relative positions.

3.3. Claims-Evidence

Claim 1 This paper claims the content of images will be better preserved than other style transfer methods. Because

style transfer is typically done by convolutions, each image will gradually lose their details with each successive iteration of the style transfer algorithm. This results in missing content and style details.

Evidence 1 The researchers compared their style transfer algorithm to ten different algorithms. Next, they calculated the content loss and style loss over 40 different styles applied to 20 different images to generate 800 stylized images. As a result, the researchers found that the transformer-based approach $StyTr^2$ had the lowest content lost and the second lowest style loss, showing that $StyTr^2$ can simultaneously preserve the input content and reference style.

	$L_c \downarrow$	$L_s \downarrow$
OURS	**1.91**	<u>1.47</u>
STYLEFORMER	2.86	2.91
IEST	<u>1.97</u>	3.47
ADAATTN	2.29	2.45
ARTFLOW	2.13	3.08
MCC	2.38	1.56
MAST	2.46	1.55
AAMS	2.44	3.18
SANET	2.44	**1.18**
AVATAR	2.84	2.86
ADAIN	2.34	1.91

Table 1. Numbers in bold are the best value and numbers underlined are the second best value. (Deng et al., 2022)

Claim 2 This paper claims their transformer-based architecture will prevent content leak.

Evidence 2 The content leak issue typically occurs in CNN-based feature representation because CNN-based style transfer potentially may not capture enough details in the image content. This artifact is easily noticed by the human eye after repeating several rounds of the same stylization process. To solve this problem, one would have to implement a reversible network to replace CNN-based models, which may not be suitable for art generation. Reversibility may also impact the robustness and feature representation negatively. Using a transformer-based model will prevent the content leak problem because it does not use convolutional networks. To prove this, the researchers compared their model to 10 CNN-based models, ran the stylization process 20 times and saw that the in the final image, CNN-based methods resulted in a blurry photo whereas the transformer-based method retained a distinct image with clear content structures. This means the transformer-based model captures precise content representation while avoiding content-leak.

Claim 3 This paper claims that its Content-Aware Positional Encoding (CAPE) takes semantic information of content images into account. This means CAPE takes into account the meaning of the image and recognizes the relationships between the objects in each image. As an extension of these claims, CAPE is also scale-invariant for resolution.

Evidence 3 The research paper compared CAPE with a sinusoidal positional encoding algorithm, which is not semantics-aware. This shows two cases where the input content image has repetitive patterns or is a collage via repeating one image four times. From the images, it's shown that sinusoidal PE is inconsistently stylized whereas CAPE is correctly stylized. Results shown in figure 1.

Figure 1. Comparison of sinusoidal PE and CAPE using content images with repetitive patterns. From figure 6 in section 4.4 in (Deng et al., 2022)

To prove that CAPE is scale invariant, the researchers used input images of resolutions different from the training data. In the case of sinusoidal positioning which is scale variant, the style transfer generated many visual artifacts as a result of the different resolutions. However, in the case of CAPE, there were no visual artifacts observed. Results shown in figure 2.

Figure 2. Source images from 512x512 resolution used on a model trained on 256x256 images. From figure 7 in 4.5 in (Deng et al., 2022)

Critique and Discussion Overall, I found it interesting how the researchers found inspiration from chatGPT and other transformer-based natural language processing solutions and applied it to the realm of image processing. It's very creative to incorporate other neural network techniques into already existing ones to create more effective methods. I believe the researchers claims on their algorithm being content-aware allows for a more robust image processing technique. One of the weaknesses of using transformers is that the computation time is much slower than CNN-based approaches. It would be an interesting topic to research the computation speeds of using CNN compared to transformers for style transfer.

4. Review of Paper 2: ArtFlow: Unbiased Image Style Transfer via Reversible Neural Flows

4.1. Storyline

High-level motivation/problem While there are many methods to achieve style transfer, there has not been a way to address content leak. Content leak is where the original image content is not preserved after several rounds of the stylizing process. This tends to happen because the first part of style transfer uses a fixed encoder for image embedding which over time will accumulate in image reconstruction errors brought by the decoder. This paper aims to fix the issue of content leak by using reversible neural flows and an unbiased feature transfer (An et al., 2021).

Prior work on this problem There have been many ways to achieve style transfer, with each paper having been built on previous methods. For example, style transfer started with convolutional neural networks (Gatys et al., 2015). Next, iterative optimization and feed-forward networks have been used to improve either the visual quality or computation efficiency. None of these previous methods provide a good, generalized method for style transfer. Furthermore, all of these methods have issues with content leak which will cause images to lose the styling of the original images over multiple iterations.

Research gap This paper attempts to answer the problem of content leak in style transfer. Currently, content leak is one of the biggest issues with style transfer. Content leak is either caused by accumulated image reconstruction errors from the decoder or biased training of either the decoder or the style transfer model. There is also not a generalized style transfer method for images. The existing style transfer methods are typically specialized or perform better in their own specific cases.

Contributions Firstly, this paper provides a method via reversible neural network flows which will lower the bias in the encoding of style transfer. The reversible neural flows support both forward and backward inferences and operates in a projection-transfer-reversion scheme. As a result of lowering the bias in encoding, it will prevent content leak. Furthermore, the paper provides a universal style transfer method that has generalized methods for style transfer.

4.2. Proposed Solution

This paper addresses the issue of bias in encoding by creating an unbiased style transfer framework called ArtFlow. Because creating an unbiased method to transfer style will prevent content leak, ArtFlow will be able to overcome content leak, one of biggest challenges in style transfer. Instead of using a encoder-transfer-decoder structure, the paper provides a method to use forward and backward inferences to formulate a project-transfer-reversion timeline using a pipeline called a Projection Flow Network (PFN). This is based on neural flows which will prevent bias because it makes style transfer based on deep features and reconstructs the stylized images via reversed feature inference.

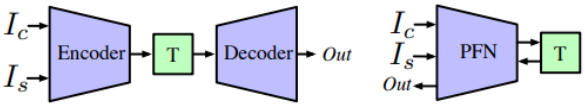

Figure 3. A graphical comparison of auto-encoder based style transfer framework and the PFN based ArtFlow framework. This allows for reversible transformations. From figure 2 in section 1 in (An et al., 2021)

4.3. Claim-Evidence Structure

Claim 1 This paper claims that pre-existing style transfer algorithms all have issues with content leak and that it will identify the three main causes of content leak.

Evidence 1 The paper first begins by explaining what is content leak in order to explain why it occurs and how to empirically test for it. Content leak is the loss of content information due to stylization, so to prove content leak is happening, the researchers took an image and repeatedly applied style transfer to it for 20 rounds. At the end, there is almost none of the detail remaining from the original content image. Thus, this proves content leak is a real phenomenon.

Next, the paper explains why content leak happens. There are multiple reasons why this occurs, and the paper identifies three main reasons why: reconstruction error, biased decoder training, and biased style transfer module. Reconstruction error explains why content leak happens because algorithms cannot achieve lossless image reconstruction and as a result, each subsequent stylization round will cause

more and more leak.

For biased decoder training, it causes content leak because the auto encoder is trained to trade off stylization rather than trying to reconstruct images perfectly, which after multiple rounds of stylization will result in greater and greater amounts of style loss. Biased style transfer module is another cause of content leak which is similar to biased decoder training. The style transfer module causes content leak because the stylization process is not reversible, and thus each patch of the image that is replaced with a new image cannot be recovered, causing content loss with each subsequent stylization round.

Claim 2 The provided neuro flow-based algorithm will result in unbiased and lossless feature transfer.

Evidence 2 The researchers provide the neuro flow-based algorithm that is reversible which allows for it to take the forward and reverse inference. To do this, the researchers created a style transfer framework that contains a Project Flow Network (PFN) which allows for a pipeline to the forward and reverse inferences. This makes the algorithm lossless as it can access its previous image patches before it is replaced. This allows for the algorithm to avoid the influence of the image reconstruction error during the stylization process and prevents content loss.

Claim 3 The researchers claim to provide a novel style transfer framework which allows for a universal style transfer framework. The new framework will integrate pre-existing style transfer methods and will not have issues with content loss.

Evidence 3 The researchers integrated ArtFlow with numerous style transfer methods and compared the results to each other. In each integrated framework, they measured zero content loss and zero style loss, meaning that the ArtFlow method is completely lossless. Because of this, the new framework will not have any content leak. The researchers then conducted a user study where they compared multiple pre-existing style transfer methods with their integrated frameworks and ran the style transfer algorithm 20 times on each framework. They then asked users to select their favorite generated images out of a list of 1000 results and ArtFlow won with 314/799 of the votes.

4.4. Critique and Discussion

This is a very interesting approach to combating content leak. The main paper being used does not try to address this problem in their approach. Something interesting is that ever paper identifies different issues with the current leading style transfer methods and tries to address a different issue. Overall, I don't think this is a super important issue in the long-run. To me, the point of style transfer is to have a little bit of content leak and to allow for a little bit of degradation so the outlines don't look too harsh when the style is applied.

5. Review of Paper 3: Text to Image Generation with Semantic-Spatial Aware GAN

5.1. Storyline

High-level motivation/problem Another use for transformers besides style transfer is to have semantic recognition on text prompts. This is used in text-to-image synthesis, a popular topic in the computer vision and generative art community. An issue with the current frameworks is that it's difficult to generate photo-realistic images that are consistent with the text descriptions. Many generated images suffer the same issue in that some parts or regions are not consistent with the words in the input sentence. This means that the generated image is not spatially aware of its surroundings.

Prior work on this problem There has already been work done on text to image generation, as it is a relatively common topic. Current accomplishments are in semantic recognition and generative art. Generative algorithms are able to generate images based on text prompt, although they are not always necessarily accurate to the prompt. The most recent methods are multi-stage refinement frameworks that generates an initial image from noise with sentence embedding and refines the details with fine-grained word embedding in each following stage. Each stage has a generator and discriminator to synthesize higher-resolution images. This work in embedding was also used in the transformer-based style transfer method.

Research gap When using pre-existing methods, multiple generator-discrimination is used to decide whether the generated image is realistic enough. Using this leads to higher computation cost and more unstable training processes. Furthermore, the quality of the image generated in the early stages decides whether or not the generated image is poor. Another issue with current methods is it very hard for text-to-image synthesis to generate photo-realistic images with semantic consistency from the input text.

Contributions This paper proposes a novel one-stage framework SSA-GAN for image synthesis from text. This method uses sentence embedding during synthesis. Overall, this new method requires less computation and can be trained more efficiently and stably. The method also uses sentence embedding during the synthesis process which lowers computation cost. Finally, the paper provides a semantic

mask predictor that is trained in a weakly supervised way.

5.2. Proposed Solution

The paper uses a novel one-stage framework called SSA-GAN for its image synthesis. It begins with using a pre-trained encoder that is a bidirectional LSTM. Next, the SSA-GAN contains a semantic-spatial aware (SSA) block that takes an encoded text feature vector and image feature maps and outputs image feature maps which are then fused with text features. After that, a weakly-supervised semantic mask predictor takes the upsampled image feature maps as input and predicts a semantic mask. This mask is predicted based on the current generated image feature maps. Finally there is normalization applied and a one-way discriminator is applied which concatenates the features extracted from the generated image and guides the generator to synthesize more realistic images.

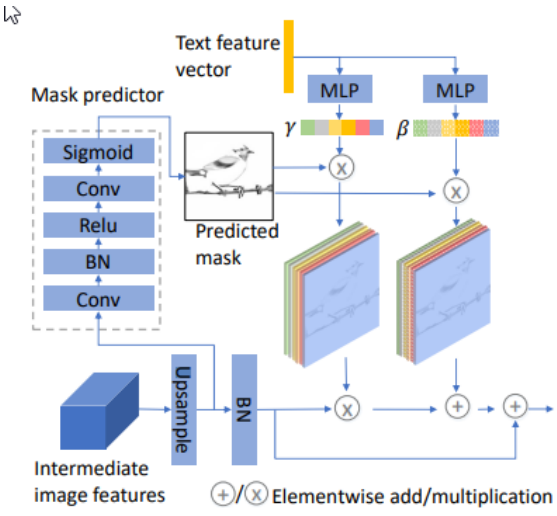

Figure 4. The structure of the SSA block. The text-aware affine parameters are learned and the semantic mask is predicted from current image features in order for Semantic-Spatial Condition Batch Normalization. From figure 3 in section 3.2 in (Liao et al., 2022)

5.3. Claim-Evidence Structure

Claim 1 The SSA-GAN will produce better and more high-quality images than existing GAN methods of text-to-image generation.

Evidence 1 The researchers compared multiple GAN methods on text-to-image generation to SSA-GAN. In order to quantify performance, they used the evaluation metrics of Inception Score (IS), Frechet Inception Distance (FID), and R-precision. IS computes the KL-divergence between the real and generated images. FID computes Frechet Distance between the features distribution of the generated and real world images and R-precision evaluates image-text semantic consistency. Results from testing show that SSA-GAN performs significantly better in IS, has the second best FID score, and has a better R-precision than all but one method.

Claim 2 The paper claims SSA-GAN will generate images with semantic consistency from the input sentence.

Evidence 2 The researchers compared generated images from SSA-GAN to three other state-of-the art GAN Models and generated various types of birds. In the result, it showed that SSA-GAN has a more realistic background and are semantically consistent with the prompts. Some of the other methods do not reflect some of the more specific aspects of the prompt such as a bird with ”red eyes” or ”black bills.” Overall, SSA-GAN has shown that it can generate photo-realistic images of birds with semantic consistency.

Claim 3 The semantic masks in SSA-GAN will affect the performance of the network and assist in building more realistic and semantically consistent images.

Evidence 3 The researchers conducted an ablation study which goes over the SSA block and the DAMSM. First, it checks the SSA blocks by comparing it to a baseline GAN. When the SSA and DAMSM is added to the network, the overall performance improves, and the IS and FID scores show remarkable improvement. Next, the semantic mask is tested. The semantic mask provides spatial information in each SSA block which affects the text-image fusion process. To test this, they add a mask predictor one by one from the last SSA block and compare the performance to an image without the semantic mask applied. As a result, applying 7 stages of semantic masks will optimize the IS and 4 stages of semantic masks will optimize the FID score. This concludes that the SSA-GAN improves performance overall.

5.4. Critique and Discussion

Something that irks me is that SSA-GAN didn’t outperform some of the other leading text-to-image generation methods. Overall, it performs on some specific datasets but it does not perform as well as other leading GAN methods when trained on datasets containing a disparate variety of images. I think the prompts are very specific and had a lots of minor requirements that caused potential issue with semantic consistency. if the prompts were to be simpler, the generation could be a lot more semantically consistent. Overall, this paper showed me the other potential uses of transformers besides just the style transfer. Seeing as this paper cited the Style Transfer with Transformers paper, it shows the other potential uses of transformers in image generation.

6. Implementation

6.1. Implementation Motivation

My plan is to implement a method to give multiple style images and have them blended together instead of just having a single style image. I have changed this plan from the previous checkpoint's plan to introduce text-based prompts for image generation because I found the computational power required to train the generator and discriminator networks effectively exceeded the time frame and the hardware capabilities of the project. Furthermore, my preliminary attempts at generating images were nowhere near realistic and had the limitation of my hardware prevented me from generating images within a realistic time frame. With my new plan, I hope to learn how different aesthetics will be mixed if the style images are completely disparate. Overall, I think that because multiple styles will be mixed, the final image will not have a very cohesive appearance. Because of this, I want to add a weighting metric that will allow for each style image to have a different amount of influence in the final image. That way, I will be able to tune the image for better aesthetics. In summary, I believe mixing multiple style images together could aid in the artistic process by creating some interesting images.

I am hoping to accomplish a somewhat aesthetic looking style transfer. While looking at the supplied images in the paper, I noticed that the supplied images with the best results are aesthetically similar images. Because of that, I would like to test style transfer with very disparate images. Furthermore, I am looking to play around with the training parameters to see what kind of images I can generate. Personally, I do not mind if there is a little more content loss in the name of less style loss due to my artistic tastes.

6.2. Implementation setup and plan

First, I will rerun the experiments claimed in the paper such as testing the content-aware positional encoding, content leak over multiple iterations, and evaluate if the style transfer algorithm is scale invariant (Deng et al., 2022). To test the CAPE, I will generate an image where the photo is duplicated four times in a 2x2 array and run the style transfer algorithm on it. After that, I will evaluate that the image to see that all four quadrants of the same image were stylized in the exact same manner. Second, I will test content leak by running the style transfer algorithm on the same image for 20 rounds. I will then confirm that the final image has retained most of its original contents. Finally, I will test scale invariability by using images of different resolutions and evaluate it the image generated is successfully stylized without any artifacts caused by the difference in resolution.

After rerunning the experiments, I will move into implementing my extension of allowing for multiple style images. For datasets, I will be using the wikiart image database for the styles as it will provide a wide variety of artistic styles and I will be using Common Objects in Context 14 (COCO14) and pictures taken by myself and friends for my content dataset. As a base, I will take the code StyTR2: Image Style Transfer with Transformers: the first paper reviewed. Next, I'll refactor the code to allow for it to run on Google Colab and reuse the helper functions that help set up the images for stylization. After that, I'll modify the code to allow for multiple style images to be blended together and add functionality for deciding the weighting for each style image. This means I will have to rewrite the transformers code, the StyTR2 code, and the code that executes the style transfer algorithm. Getting multiple style images to work is my highest priority in this, and adding the weights will be a secondary goal. Overall, getting the weighting and the blended style images will allow for interesting images to be generated.

6.3. Implementation details

To set up the project, I used the code supplied from StyTR2: Image Style Transfer with Transformers: the first paper reviewed. I reused the helper functions provided by the authors that prepare the images for stylization and set up the tensors. I rewrote the transformer, StyTR2 model, and testing code to allow for multiple style images to be used as well as the weights for each image. For libraries, I used Torch, torchvision, and NumPy and used a pre-trained model supplied by the authors. To get all of this running, I used Google Colab as an environment to hold my training and testing python files as well as the necessary dependent files.

6.4. Results and interpretation

I set up a few reference images and styles to generate some style transferred images. First, I noticed that the CAPE algorithm is truly scale invariant. The images I supplied were all of varying resolutions, which means if StyTR2 was not scale invariant, the images would have a lot of visual artifacts and the stylization would not be successful. Since my images were successfully stylized, it confirms that the StyTR2 algorithm is scale invariant.

Next, to confirm that the content-aware position encoding algorithm was functional and was semantics aware and capable of "comprehending" repetition in images, I reran the experiment of testing content-aware position encoding by creating an image that was duplicated 4 times and ran the style transfer algorithm on it. The result was a stylized image where the photo recognized the image was repeated four times. This confirms that the CAPE algorithm is semantics-aware.

Finally, to confirm that the StyTR2 algorithm with transformers is capable of preventing content leak, I set up a

Figure 5. Style transfer results from images with mismatched resolutions.

Figure 6. Testing content-aware positional encoding in style transfer using content image with repetitive patterns.

script to recursively apply style transfer to an image 20 times. After running the algorithm 20 times, I could clearly see the original shape of the content image. In this case, it was a pagoda building which still retained a clearly defined shape after 20 iterations. Since the pagoda was still visible, it's evident the StyTR2 algorithm prevents content leak.

For the extension of the paper, I rewrote the code to allow for multiple styles and tested multiple styles blended together by using two style images instead of one to stylize a single content image. When using two styles, each style image's contributions can be clearly seen as shown in figure 8. From visual inspection, it shows that blending multiple styles together could be used to come up with new artistic ideas. Something I noticed was that when more than 3 styles are used and the images all have equal weighting, it is hard to discern each style image's contribution because there are so many styles influencing the image. Using multiple styles without manually re-weighting the image causes the final image to be very "noisy." Thus, it is imperative to have one or two style images be the highest weight in the style

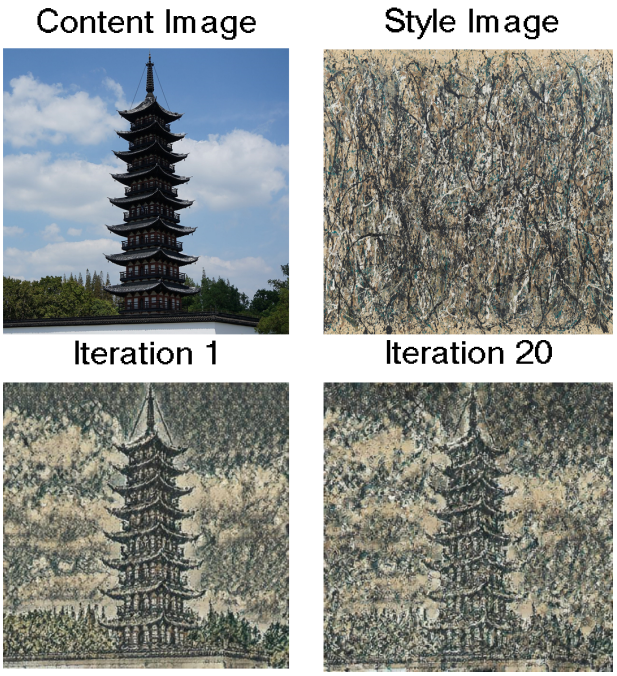

Figure 7. Evaluating content leak of style transfer over 20 iterations. The original outline of the content image is still present after 20 iterations.

contribution in order to have a cohesive final image. Another possibility with weighting is to have the weights go above the sum of one. For example, in figure 9, I set the weight of one image to be 4.0 and the entire sum of the weights equals 6. In this case, the imprint of the style is much stronger than with normal weightings and it results in some content loss. This situation results in some interesting images generated and could be used in some artistic scenarios where content preservation is not very important.

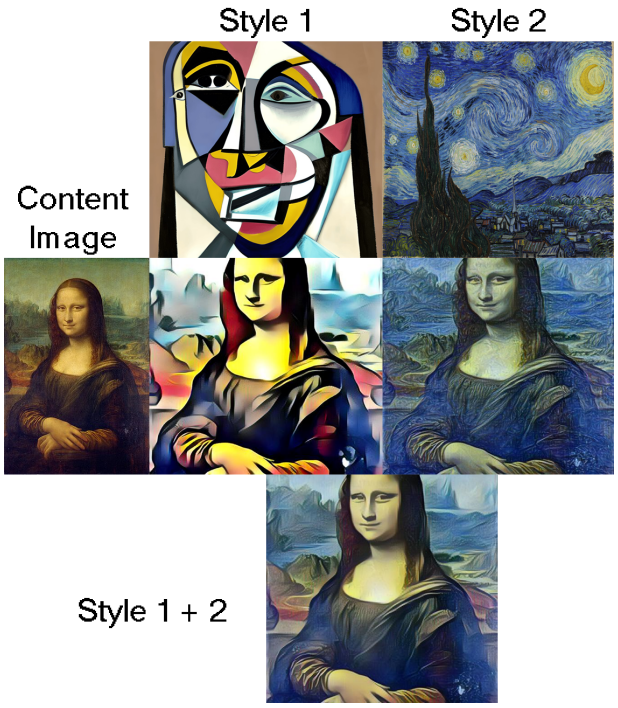

Figure 8. Style transfer with two styles weighted at 0.5 each (Total weighting sums up to 1.0). Each style image's contributions are easily discernible and the final output is a cohesive image.

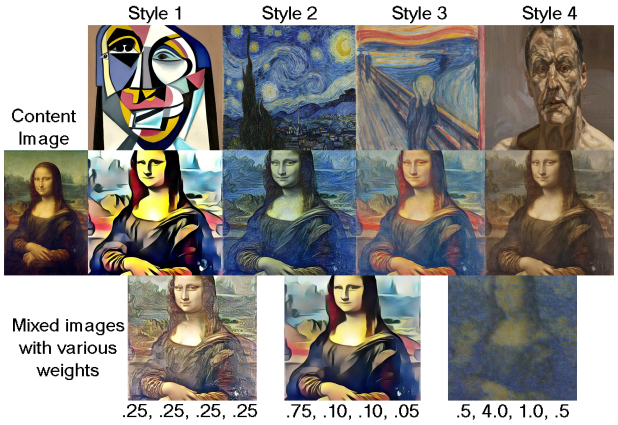

Figure 9. Style transfer with four styles with various style image weights.

7. Conclusion and Discussion

Overall, style transfer is a very interesting way to achieve artificial intelligence-assisted generated artwork. While re-running the experiments and testing my extension, I noticed another use for style transfer could be transferring color palettes onto other images such as giving color to black and white images. The usage of transformers allows for some of the key weaknesses in convolutional neural networks to be solved through the usage of content-aware positional encoding and reversibility which results in semantics-aware, scale invariant, and content-loss free images. When evaluated, all the claims from the authors were met.

Some limitations in this project include requiring reference images to generate artwork and the long computation time when using many style images. Some ways to fix these limitations could be adding in natural language processing for a text-based prompt to help generate images and reuse some parts of the traditional convolutional neural network architecture to speed up computation time.

To conclude, the idea of style transfer has a plethora of potential use cases in the art industry and each advancement in style transfer provides a better, more efficient way of producing art. In the future, this project could be extended to allow for real-time style transfer akin to Snapchat filters or Teams backgrounds or for video style transfer for usage in animation.

References

An, J., Huang, S., Song, Y., Dou, D., Liu, W., and Luo, J. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Deng, Y., Tang, F., Dong, W., Ma, C., Pan, X., Wang, L., and Xu, C. Stytr2: Image style transfer with transformers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Gatys, L., Ecker, A., and Bethge, M. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems 28*, 2015.

Liao, W., Hu, K., Yang, M. Y., and Rosenhahn, B. Text to image generation with semantic-spatial aware gan. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18166–18175, 2022. doi: 10.1109/CVPR52688.2022.01765.