# RAG-Inspired Style Transfer for WebToons Comics

Zach Meurer

December, 2024

**Abstract**

To address the problem posed by state-of-the-art style transfer models lacking the ability to interpret art styles of the titles of niche media, I have created a RAG-inspired model to allow for effective style transfer of niche media. The model focuses on interpreting the styles of webcomics scraped from WebToons given solely the series title. Additionally, the model incorporates image similarity search on a given content image to find an ideal style image to pair with it for Gatsy's neural style transfer [1].

## Introduction

State-of-the-art style transfer applications (e.g. DALL-E3, Adobe Firefly, OpenArt) lack the context to understand the styles of niche media solely based on text prompts. Given a prompt like *Recreate this image in the style of Tower of God*, modern AI style transfer models fail to truly capture the style of the inputted niche WebToons series (Figure 1). All of the AI had an understanding that the series was animated in an anime or manga style, but they could not interpret the true style based on the text prompt alone. To give the models credit, they perform well at interpreting the styles of popular media like *SpongeBob Squarepants* or *Avatar: The Last Airbender*

To interpret the true style of niche media like webcomics, I have designed a unique neural network architecture to style an input image given a webcomic series name. The architecture is inspired by the Retrieval-Augmented Generation (RAG) model first proposed by Lewis et. al [2]. Using a style image database for webcomic series look-up and image embeddings, my model circumvents learning a text-to-style connection for the simpler approach of content and style image inputs for style transfer.

My model and its code are documented on my GitHub.


Figure 1: State-of-the-art style transfer AI fails to successfully capture the style of Tower of God, a popular webcomic series on WebToons. The same prompt (text and image) was given to ChatGPT, Adobe Firefly, and OpenArt's style transfer model. The actual style of the show is displayed on the left-most side of the figure

## Related Work

*This is where you give a brief overview of any prior work by others (or yourself) that is relevant to the problem and solution you are proposing. Cite any papers using the citation and bibliography syntax illustrated below.*

My model's style transfer AI is based on Gatsy's *A Neural Algorithm of Artistic Style* [1]. This was the first paper to introduce the neural style transfer model as opposed to other methods of style transfer. The algorithm extracts feature maps from the inputted content and style images by feeding them through a pre-trained CNN model. They extracted feature maps from VGG-16 from each convolutional layer following a max pooling layer [3]. The feature maps are used to compute Gram Matrices which represent the style of an image. The algorithm generates an image by reconstructing the content image from noise or the content image while aligning the styles of the generated and style images. This is achieved through a unique loss function dependent on a combination of the content image reconstruction MSE loss and the gram matrix MSE loss.

Initially, I attempted to implement Adaptive Instance Normalization as it was a significantly quicker form of neural style transfer with a similar level of quality. This technique was first introduced by Xun Huang and Serge Belongie in 2017 [4]. Paper reconstructions of their produced work have bore limited success. Attempts at reconstruction online have shown results dissimilar to those appearing in the paper [5].

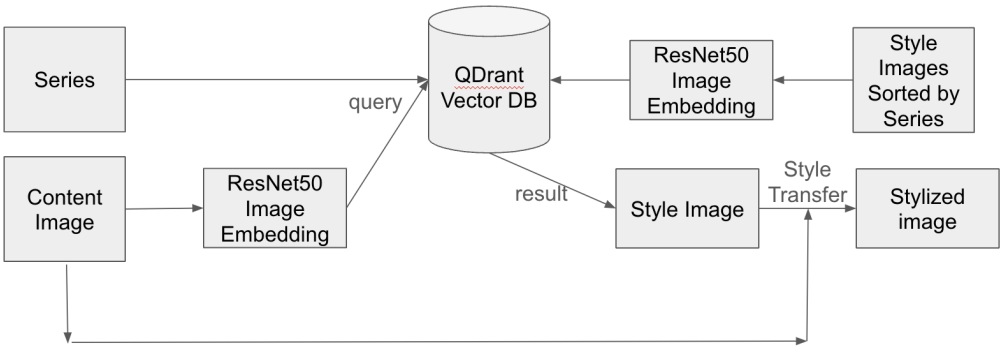


Figure 2: RAG-inspired model architecture.

Webtoon created a style transfer model in 2023 [6]. This model performs style transfer and text-to-image generation, specifically focused on the adaptation of the style. Despite the model being produced by WebToon, it is a general model, not specialized in WebToon comics art styles.

## Approach (or Methodology)

My model's unique architecture is detailed in Figure 2. The model takes an input of a content image (to be styled) and a WebToons comic series title. The content image is embedded as a vector by using the logits of a pre-trained ResNet50 image classification model. The content image embedding vector is of size 1000. This embedding vector encodes the semantics of the content image. Images with cosine similar vectors thus have similar semantics. The content image embedding and series are used to query the vector database which has been populated by WebToons style image vector embeddings sorted by series. The query finds the $k$ most similar style images from the given series to the content image. Similarity between content and style images combats interference from the style image in content image continuity, which I have observed to be a major issue with Adobe Firefly. Finally, each of the $k$ style images are used to style the content image with Gatsy's style transfer [1] to output $k = 2$ styled images.

No training of the network is required for operation because the neural style transfer is "trained" to style each individual image. The only prerequisite to use is populating the vector database. Once the images are processed to vector embeddings with ResNet50, they build a Qdrant vector database, sorted into classes based on the series they belong to.

For the neural style transfer, I implemented gatsy's [1] loss function which I detailed in

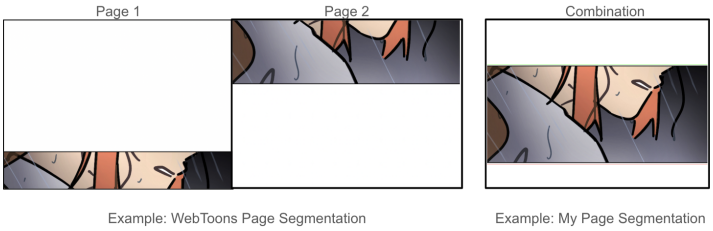


Figure 3: On the left is an example of WebToons cutting an illustration into two parts arbitrarily. On the right is the result of my image segmentation method.

the related work section. The content loss and style loss weights I chose were 1 and 0.01 respectively. The training is optimized by Adam with a learning rate of 0.001 for 600 epochs.

## Datasets

For content images, I downloaded a subset of Microsoft's 2017 COCO dataset from Kaggle.

For style images, I scraped a dataset of 3,000 high-resolution images of illustrations throughout WebToons comics. Each of these images is paired with the series to which it belongs. There are almost 500 unique series. The dataset was scraped concurrently in batches using Python's BeautifulSoup, PIL, and asyncio libraries. The process consisted of first, scraping the entirety of each episode for a particular series, constructing one extremely tall image of all the pages stacked on one another. Next, I segmented the episode image based on appearances of horizontal lines of one distinct color. Between these segments lie each of the full, continuous illustrations. Originally, WebToons segments the pages arbitrarily, splitting illustrations between pages. This resulted in unappealing images. Thus, my image segmentation method proved fruitful and resulted in continuous pages. Lastly, I scraped the $k = 3$ most square images from each episode, excluding any that were too wide or too thin. I could have scraped far more images than I did but I chose not to due to size constraints.

## Evaluation Results

My Results can be evaluated qualitatively. There are no commonly agreed upon quantitative measures for style transfer as the quality of the results is based on taste. The common practice of novel style transfer models is to compare results to other models. Thus, I will compare two examples of my model competing against a state-of-the-art model, Adobe Firefly, with the same inputs.

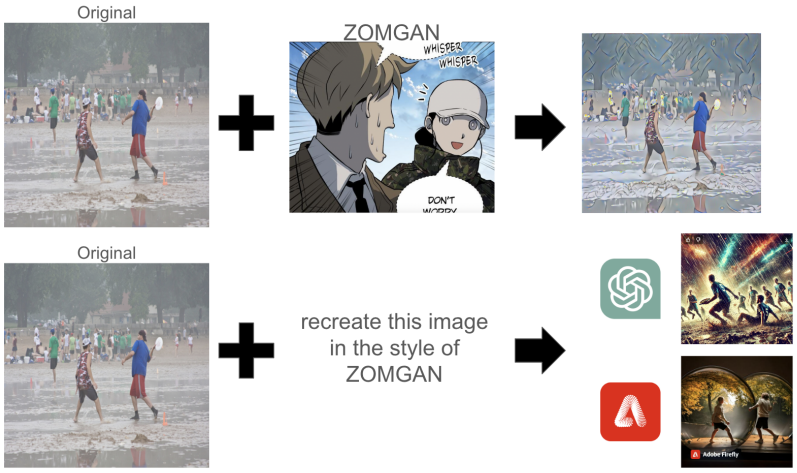


Figure 4: Comparison between my model's styled image and ChatGPT & Adobe Firefly's styled image given the same task. The series chosen is WebToons' ZOMGAN

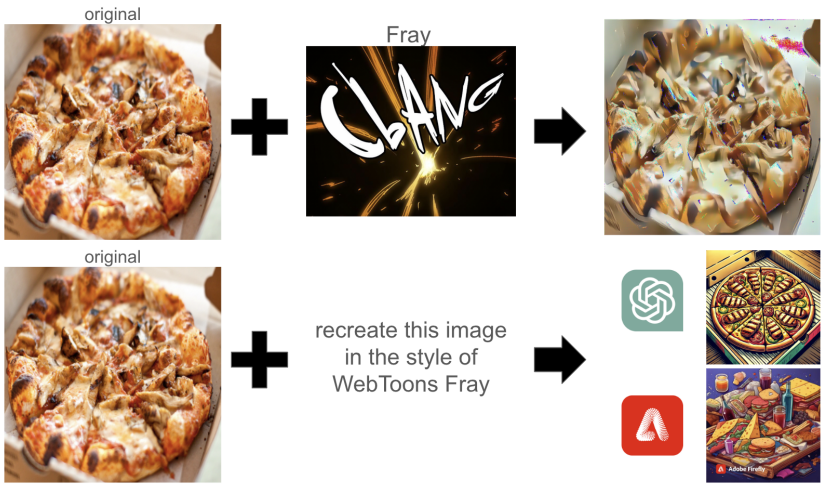


Figure 5: Comparison between my model's styled image and ChatGPT & Adobe Firefly's styled image given the same task. The series chosen is WebToons' Fray

## Conclusion

My project circumvents the problem of AI's text-to-style understanding by simplifying the problem to image-to-style understanding. This task is far easier for an AI than text-to-style. My unique RAG-inspired architecture facilitates this task by implementing a webcomic series-based lookup table. This query system paired with image similarity helps my model perform better at style transfer with text and image inputs than state-of-the-art AI.

Future work on the RAG-inspired style transfer architecture could involve improving the neural style transfer method. Gatsy's neural style transfer algorithm [1] was the first of its kind, introduced in 2016. Since then, numerous papers have been published improving on the approach such as Huang and Belongie's adaptive instance normalization approach [4]. Although my attempt at training an AdaIN model was unsuccessful, there are many other neural style transfer approaches for my architecture to adopt.

Another opportunity for future work involves populating the vector database with more images. With my existing data scraping code, it is easy to scrape far more (¿100,000) images than I have for this project.

## References


[1] Leon Gatys, Alexander Ecker, Matthias Bethge. *A Neural Algorithm of Artistic Style.* Journal of Vision, 2016.

[2] Patrick Lewis et. al. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.* Facebook AI Research, University College London, New York University, 2021.

[3] Karen Simonyan, Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition.* ICLR, 2015.

[4] Xun Huang, Serge Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization.* Department of Computer Science & Cornell Tech, Cornell University, 2017.

[5] Isaac Fung, Hersh Vakharia, Ali Baker, Anthony Ke *Adaptive Instance Normalization Style Transfer.* University of Michigan, 2020.

[6] Namhyuk Ahn, Junsoo Lee, Chunggi Lee, Kunhee Kim, Daesik Kim, Seung-Hun Nam, Kibeom Hong. *DreamStyler: Paint by Style Inversion with Text-to-Image Diffusion Models.* NAVER WEBTOON AI, Harvard University, KAIST, SwatchOn, 2023.