A Review and Comparison of Neural Style Transfer Methods

Marc Ruiz
American University
4400 Massachusetts Avenue NW,
Washington, DC
mr4162a@american.edu

Abstract

This report replicates two implementations of popular Neural Style Transfer algorithms. The first of which is the Color-Aware Multi-Style Transfer method which appears to be a promising advance in the NST space. The second is the color preservation technique implemented by the authors of the first NST paper that inspired many others. Both methods are very exciting, however, there was a lack of comparisons between the two in the literature. This report will compare evaluation metrics and give a subjective analysis of the resulting images from implementations of the methods from both papers.

1. Introduction

It can be difficult to compare the Neural Style Transfer techniques across different sets of images considering that it is not always clear how a given style may alter one image or another. This occurrence is especially unfortunate for a casual reader of the literature when interesting ideas are presented by different researchers, but we readers don't immediately receive the comparisons we would like!

Well, this report will not solve that phenomenon. However, it does hope to address one specific instance of it in the NST space. The motivations for this investigation were derived purely out of a curiosity for two promising methods.

2. Background

The practice of image style transfer has seen a large surge in experiments and techniques aiming to refine the methods that exist. Most popular is the Neural Style Transfer method first published by Gatys et al. in 2016 [3]. Their work demonstrated a framework for transferring feature maps from 'content' and 'style' images in the form of a Gram matrix generated from layered iterations on a CNN pretrained in top-of-the-line texture capturing [3]. In simpler terms, the method involves breaking down images and building them back up in iterations with different weights to create a new image where the 'content' is preserved but the 'style' has been replaced. Gatys et al. write out the formula for the loss between two feature presentations in equation I [3]:

$$\mathcal{L}_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}\left(F^l_{ij} - P^l_{ij}\right)^2 . \qquad \text{(I.)}$$

The authors take the derivative from these operations as well in order to represent the combination process of feature maps of images into their Gram matrix in equation II [3]:

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}. \qquad \text{(II.)}$$

This key understanding of how neural systems could manipulate carefully represented images has resulted in a phenomenal response from the computer vision community. Very many researchers are eager to get their hands on the techniques and results that exist in the broader domain of style transfer.

Among the eager crowd of talented researchers in the community are the authors of the Color-Aware Multi-Style transfer paper (CAMS), Afifi et al [1]. Their paper contributes to the literature an algorithm that simply, but efficiently extends the work of Gatys et al. [3]. Whereas the original relied on a single Gram matrix for the representation of an entire image's style features, Afifi et

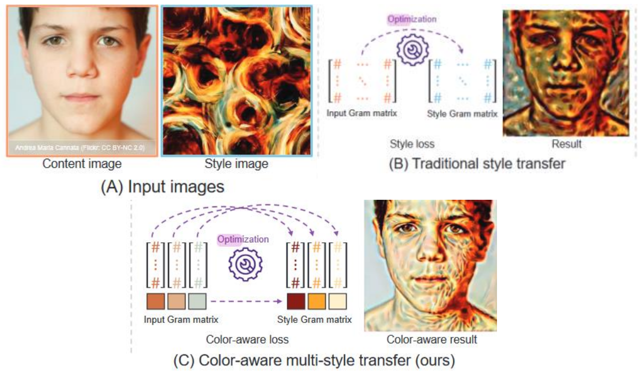

Figure 1: . Afifi et al demonstrate single gram vs. multi gram abstractions.

al. outline and implement a method that creates multiple Gram matrices for each image as seen in figure 1 [1].Their objective with this practice was to simulate a locality based capturing method.

Afifi et al. make the claim that the NST method had a clear lack of color preservation, from content image to generated image. Their proposed solution to this effect was a creation of color masks from each pixel to serve as weights against the feature maps in the equation for a Gram matrix, as seen in equation III and IV [1]:

$$\hat{F}^l = F^l \odot \mathrm{W}_\mathrm{t}, \qquad \text{(III.)}$$

$$G_{ij}^{l(\mathrm{t})} = \frac{1}{m^l} \langle \hat{F}^l_{(i:)}, \hat{F}^l_{(j:)} \rangle \qquad \text{(IV.)}$$

Further clarification of the variables can be found in their detailed methodology.

They also hypothesized that the algorithm was suffering in cases where a style image had multiple 'styles', resulting in a total loss of clarity from the content image as the styles jumbled the generated image.

With those issues in mind, Afifi et al. provided CAMS as an approach to add a color, pixel-based weight to the iterative layer process behind this NST algorithm.

Now of course, with the many techniques available, it can be easy to overlook some of the more unknown ones. However, the other method that will be compared to CAMS is not made by someone new to the field. Gatys et al. provided a solution to their color preservation problem in a subsequent paper [2], during the same year in which the first was published [3]! This subsequent paper is even cited by Afifi et al. as among the number of algorithms they have seen emerge in recent years, but it is not clear why they do not demonstrate comparisons of it. It would seem like a competitive example to make their algorithm compete against.

In a very simple extension, Gatys et al. demonstrate the ability for their algorithm to preserve color in two scenarios [2]. One in where color histograms are used to transform the style image in order to meet the content image's colors before the neural style transfer method is run. And a second in which the neural style transfer method is applied to the luminance channels of each image, and after the neural style transfer is done, the color from the content image is applied to the results of the style transfer. Both of these techniques sound like very simple extensions to the original because they are for the most part! Gatys et al. discuss some of the pros and cons with either method, but both are highly effective at correcting the shortcomings of the originally published algorithm.

Thus, I advocate for a brief pause in order to go over a straightforward comparison between CAMS and the color preserving NST [1], [3].

3. Methods

The preparation phase began with a clone of two repositories.

The implementation of the paper by Afifi et al. was coded and cited by their very own team and can be found in the citation for their publication [1]. Considering that the team behind the paper published that repository, I actively chose not to make very few changes to their code. The assumption made here is that the state in which their color_aware_st.py file was retrieved is a true representation to their work. My first addition to their repository was the addition of an __init__.py file in order to create a package out of it, for the sake of importing their evaluation metrics to the other repository. My only other addition was the implementation of those same metrics they included such as cams_loss.py.

Unlike the first, the second repository did require a few lines of comprehensive tuning to the NST implementation [4] by rrmina in order to receive outputs most closely representative of the guidelines in the Gatys et al. paper itself [2]. Firstly, Gatys et al. describe two solutions as comparably, or even inversely, excellent. The owner of the repository implemented the NST across the luminance channel in order to get some transformed image L_t, which combined with a YIQ color extraction from the content image in the end. This is perfectly acceptable along the guidelines of the paper. The first changes began with a switch in optimizers, from the adam scheme into the lbfgs that Gatys et al. preferred in the original publication and make no mention of in the new framework [3], [2]. Next was the computation of loss; rrmina seemed to have preferred combining the content loss and style loss with another 'Total Variation Loss' in order to compute the total loss function [4]. I carefully removed this addition from the program without affecting other parts. Lastly, in combination with first repository, I initialized the CAMS directory as a package in order to import the exact same loss evaluation metrics by Afifi et al. without the need to change parameters that might affect the evaluation.

As far as which images were used to generate new ones, I used a combination of images that were included in both repositories, so as to try and eliminate bias in example selection. Eight content-style pairs were passed through both algorithms, the generated image was returned, and so were the results of the loss evaluation metrics provided by Afifi et al.

The idea behind the methodology was simply to create and/or maintain the python environment that would best represent faithful pytorch implementations of both papers.

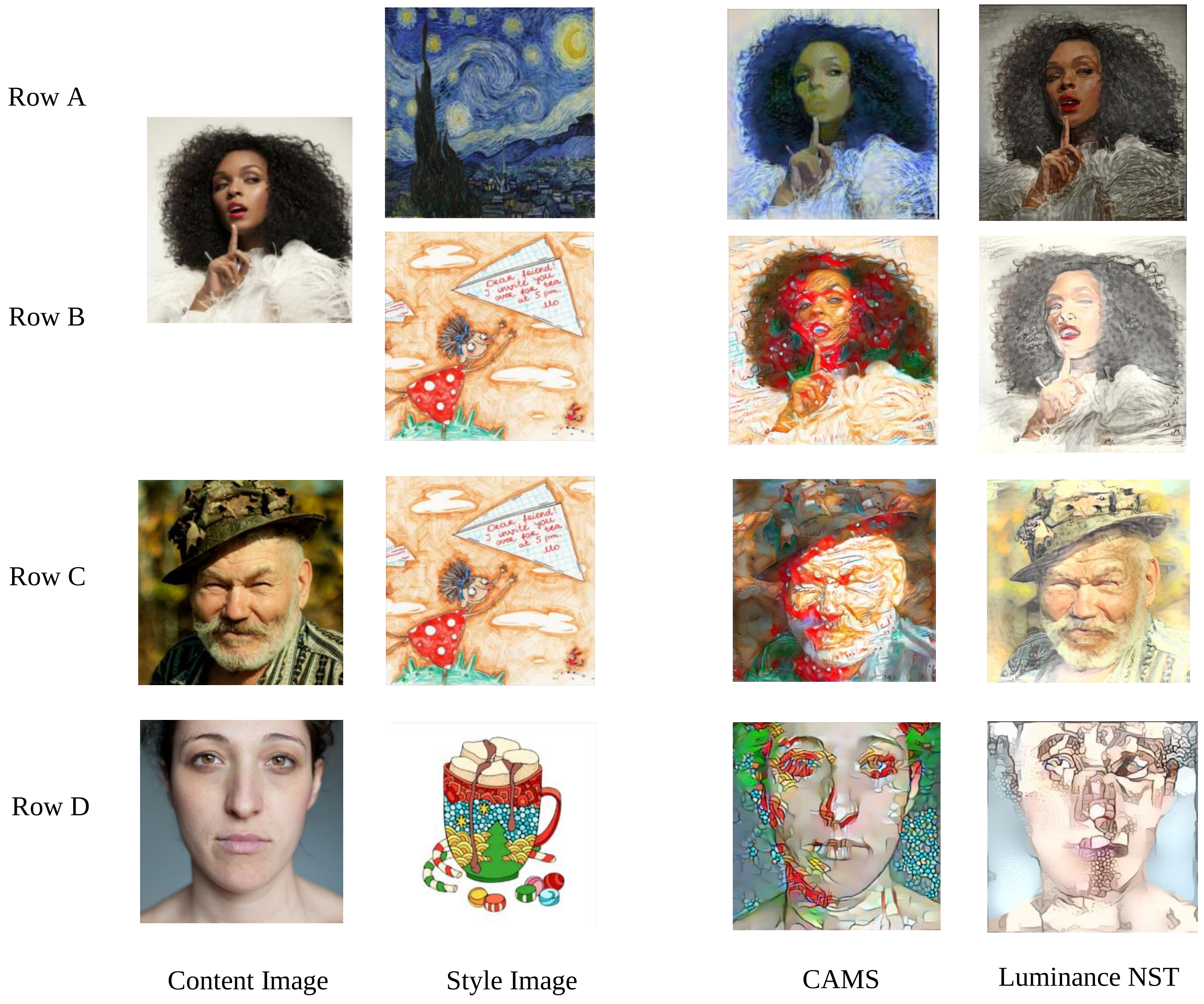

Figure 2: Style Transfer: CAMS compared against NST via luminance channel

4. Evaluation

The results as they appear side by side are rather surprising. Among the first developments that can be noted is the subjectively worse performance by CAMS than reasonably expected. The algorithm appeared to be subjectively worse at handling color in the new image examples, even though, by the images in the paper by Afifi et al., it seemed rather capable of handling both issues of style complexity and color preservation. This isn't to imply that the results from CAMS were unacceptable, only that it seemed more competitive given that the report by Afifi et al. consistently measured CAMS as the best or second-best performer in color-aware loss, style loss, and content loss [1].

Row A

	Luminance NST
Color-aware loss:	0.2968403995037079
Content loss:	38.05992126464844
Style loss:	0.016072725877165794

	CAMS
Color-aware loss:	0.059620827436447144
Content loss:	61.029815673828125
Style loss:	0.0030623970087617636

Figure 3: CAMS Loss Metrics

Row B

Luminance NST

Color-aware loss:	0.6966368556022644
Content loss:	42.993568420410156
Style loss:	0.028794419020414352

CAMS

Color-aware loss:	0.09166824817657471
Content loss:	102.91493225097656
Style loss:	0.004647729452699423

Row C

Luminance NST

Color-aware loss:	0.5330871343612671
Content loss:	47.17280197143555
Style loss:	0.02263505943119526

CAMS

Color-aware loss:	0.09884297847747803
Content loss:	116.62101745605469
Style loss:	0.004589798394590616

Row D

Luminance NST

Color-aware loss:	3.1078193187713623
Content loss:	84.98823547363281
Style loss:	0.079314224421978

CAMS

Color-aware loss:	0.8481253981590271
Content loss:	224.38201904296875
Style loss:	0.022050704807043076

Figure 3: CAMS Loss Metrics

It can be seen on the metrics, however, that CAMS almost always has a lower amount of "color-aware" loss than the luminance NST. As per Afifi et al., table 2 and 3, the lowest value is the "best" result.

Additionally, it can be seen that both algorithms almost always have exceptionally low style loss scores, though the content loss scores vary.

Looking at the resulting scores and images, it would be fair to ask how much impact these scores have on the visual and aesthetic value to a person.

5. Conclusions

As NST algorithms are further developed, it will always be interesting and fun to compare the pretty images they produce. Of course, while there is fun to be had, it is also nice to produce some comparisons and question why we receive certain outcomes.

The most interesting comparison to me is that of the image pairs in Row A and Row B in figure 2. The metric for color aware loss is giving CAMS a significantly lower (better) score than it is giving Luminance NST. However, I would argue that Luminance NST generates an image that is much closer in color to the original content image than the image that CAMS generates. There are large amounts of red in CAMS from row b, and large amounts of blue in CAMS from row a. This is while the Luminance NST equivalent images manage to both eliminate all red and blue from the respective images and maintain the deep brown and black colors of the woman. Then we must wonder, if CAMS is receiving the better score despite not being as true to the content image color as the Luminance NST, what is being measured?

I would hypothesize that the color-aware loss metric is indirectly looking at the differences in the luminance of the before and after images. Gatys et al. do establish that the luminance values change in the neural process, so that color can be perfectly retrieved later. Then it may appear that the CAMS model as it is implemented could potentially have room for improvement.

This would require further experimentation and likely a sit down with the mathematics, but intuitively, it is an interesting thought. Looking at the images, it seems as if CAMS and the Luminance NST trade back and forth results of strongly defined edges over different image combinations.

Could a more well defined, color-preserving method be derived from a combination of the two? Further studying is required.

References

[1] Afifi, M., Abuolaim, A., Hussien, M., Brubaker, M. A., and Brown, M. S., "CAMS: Color-Aware Multi-Style Transfer", (2021). "https://arxiv.org/abs/2106.13920.

[2] Gatys, L. A., Bethge, M., Hertzmann, A., & Shechtman, E., "Preserving Color in Neural Artistic Style Transfer", (2016). "https://arxiv.org/abs/1606.05897

[3] Gatys, L. A., Ecker, A. S., and Bethge, M., "A Neural Algorithm of Artistic Style", (2015). arXiv:1508.06576

[4] rrmina. "neural-style: Neural Style in Pytorch! 🎨, (2020). https://github.com/rrmina/neural-style-pytorch/commit/994aa3fdd3973804e08e8b81ddfe792ceadedf4e