



## Automatic tuning for Neural Style Transfer

Project Report: CS 7643

Group name: Artuners

Ran Tavory  
Georgia Institute of Technology  
rantav@gatech.edu

Xenia Kramarenko  
Georgia Institute of Technology  
xkramarenko3@gatech.edu

### Abstract

*In this project we quantify the trade-off between preservation of content and preservation of style in Neural Style Transfer. We employ an empirical approach by using different content and style images and measure the ability of a neural network to classify the original style and content. Our results demonstrate that when too much style is applied, the resulting image accurately preserves the style, but the content is not preserved; conversely, when too little style is applied, content is well preserved, but style is not. We show that a balance can mechanically be found without human intervention at the point where content and style can be recognized with equal accuracy.*

*To extend our work we perform deeper analysis of style transfer and classification and discuss the source of some interesting results found during our research.*

### 1. Introduction, Background and Motivation

Neural Style Transfer (NST) [4] was introduced and studied in class and course assignment. There are multiple ways to implement NST, our method is based on [7], a method that results in visually pleasing images, while re-

ducing the number of tuneable parameters to a minimum<sup>1</sup>. Our method requires four parameters, `content_weight` and `style_weight`, `num_steps`, the number of iteration of image updates and `start_image`, determining whether to start from the content image, the style image or just white noise image.

Our experience shows that tuning NST parameters requires human inspection of quality of the result and therefore limited in scale and speed. In this project we suggest a mechanical way to quantitatively evaluate the trade-off between the preservation of content (content cohesion, e.g. is content still recognizable) and that of style and therefore eliminate the requirement for a human in the loop. We find that the midpoint where content and style identification accuracy equal, produces visually pleasing images in a fully automated fashion. This would enable users to apply NST in their applications in a fully automated manner.

We train two NNs, one to classify artistic style and another to classify object class. We then apply NST to images of varying style and content while applying different parameters to the content and style weights and measure the accuracy of style and content identification on the resulting im-

---

<sup>1</sup>Unlike the home assignment, our implementation did not include the Total Variation loss and it applied a uniform style weight for all layer

ages. We then qualitatively assess the correlation between machine classification to human evaluators, e.g. is there agreement between human and NN classifiers? Finally we display the trade-off visually with a content-style trade-off chart, a tool we hope would be useful in future research.

## 2. Approach

Three different networks are used, a style classifier NN, an object classifier NN and a neural style transfer network. Our method follows the general process below

1. Train a style classifier to recognize the style of one of four artists (see 2.1)
2. Train an object classifier to identify one of five classes of animals (see 2.1)
3. Apply NST to a large set of content style image pairs. Set all parameters to constants except `content_weight`
4. Measure the accuracy of style and object identification on the styled images as a function of `content_weight`.
5. Plot the results and suggest conclusions

For each artist we select multiple style images and for each object class we select multiple content images, resulting in 740 generated style transferred images per each value of `content_weight`. We apply different values of `content_weight` and then on the resulting transferred images we run the two classifiers to report their accuracy. For example if out of 100 style transferred images only 50 were correctly identified for their style and 70 had been identified with the correct object class then our style accuracy is 0.5 and content accuracy is 0.7.

We started by training style and content classifiers and after analysis came to realize the results of content classification indicate that there is still room for improvement. We pursued this small side research as an extension to the main objective of the project: enhancing the content classification outcomes of styled data.

See [B](#) for full source code of the project.

### 2.1. Data

Two datasets are used: `Impressionist_Classifier_Data` [2] for style classification and `animals-10` [1] for content.

`Impressionist_Classifier_Data` is a database of impressionist painters from which four painters were selected: *Hassam*, *Matisse*, *Renoir* and *VanGogh*. Each artist has exactly the same number of train, validation and test images, 399 train images, 99 validation images and 66 test images per artist.

`Animals-10` is a database containing different animal groups from which five are selected: *cat*, *dog*, *squirrel*, *cow* and *chicken*. The train/validation/test sets have 1000/200/100 images per object class (animal) respectively.

The train and validation sets are used for fine-tuning the SOTA networks used for style / content classification. The test images are used to test the classifier’s accuracy and later as a source for the style transfer images. It is important to note that, we did not use the train or validation sets for style transfer, we used the test images, images on which these classifiers were not trained on.

### 2.2. Classifiers

Two types of style classifier were tested. The first, we refer to as “simple” is an improved implementation based on [11]. We start with a trained ResNet18 network [6] and fine-tune it on the four artists training set. The last fully connected layer of ResNet was replaced with a linear FC layer with 4 output neurons (4 artists) and the criterion used was Cross-Entropy. The second style classifier was based on a trained VGG19 [10] and a gram-matrix applied to its 14th convolution layer, followed by FC-RELU-FC. We ended up not using this network, however more details of that work are described in 4.2.

The content classifier was implemented in a similar manner, we started with a trained ResNet18 network [6] and fine-tune it on the five animals training set. The last fully connected layer of ResNet was replaced with a linear FC layer with 5 output neurons (5 animals) and the criterion used was Cross-Entropy.

Before addressing our research question we made a few pre-research experimentation in order to come up with an efficient and accurate experiment. They are described below.

### 2.3. Resolution Matters

We experiment to test the effect of resolution of the style classifier training dataset on its ability to later recognize style after applying NST (Table 1). We conclude that the best setup is obtained not by training on full sized images, rather by training on images of width 256px (and preserved aspect ratio) and a configured style transfer of similar size, also 256px width.

It is interesting to note that although technically a classifier may operate on any size of input image by resizing it while in the general case preserving its accuracy, in particular for style identification, resizing may result in loss of accuracy; unlike the case for object classification or other tasks. We hypothesize that the cause for that might be that style identification sometimes focuses on the relations between nearby elements, local features (e.g. brush stroke and other local features) therefore resizing an image may interfere with these local features by losing some of them. We

ST Output Training Size	128 px	256 px	512 px
Full Size	0.45	0.65	0.75
128 px	0.55	0.75	0.45
256 px	0.60	<b>0.80</b>	0.60

Table 1. Comparison of style classifier training input size in pixels. We find that matching the style classifier training image size to that of style transfer is best, in our case we chose 256px

therefore continue with a classifier trained on 256px images and set our style transfer to also output images of 256px.

#### 2.4. Determining which parameters to tune

Our implementation of NST includes several parameters and since we decided to focus on content v/s style, two parameters are relevant, `content_weight` and `style_weight`. To make things simple we want to reduce them to just a single parameter that we can tune. The formula at [4], translated to our code is

$$\begin{aligned} \text{style\_score} &:= \text{style\_score} \times \text{style\_weight} \\ \text{content\_score} &:= \text{content\_score} \times \text{content\_weight} \\ \text{overall\_loss} &:= \text{style\_score} + \text{content\_score} \end{aligned}$$

Where *style\_score* is calculated based on Gram matrices diffs and *content\_score* is calculated based on image to content image MSE.

This means that the overall loss is simply a weighted average between the style and the content losses. We therefore conclude that only the ratio between the two matters (to the extent of numerical precision). We empirically validate this and the results are displayed in Table 2, we test multiple configurations of different values for style and content weight and observe that the best style accuracy is obtained for the ratios of  $10^6$  or  $10^7$ . For ratios lower than that the content accuracy degrades in a similar manner regardless of whether the style weight is modified or the content weight. We therefore focus our efforts now on the single ratio parameter.

### 3. Experiments and Results

We run NST with different values of `style_weight` and measure the content and style accuracy. We select 10 arbitrary style image for each artist and 10 arbitrary content images from each object class (animal).

In Fig 3 we see the trade-off between style and content when applying style transfer with different values of style weight. We measure the classification accuracy of over 400 images per each value of `style_weight` parameter. Even though there are a few spikes in the chart, a trend can be observed, style weight affects style and content accuracy in a

style weight	content weight	Ratio	Style Accuracy
-	-	Test set	0.9
1e7	1	1e7	0.8
1e6	1	1e6	0.8
1e5	1	1e5	0.575
1e4	1	1e4	0.425
1e6	10	1e5	0.6
1e6	100	1e4	0.375

Table 2. Using the classifier trained on the 256px art images we test multiple values for style and content weight and observe that what matters is only their ratio

For comparison the first row contains the accuracy of the non transferred original test set.

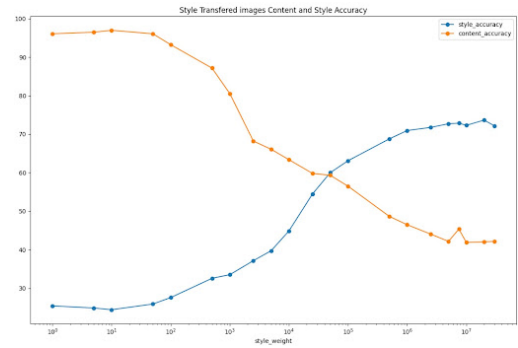


Figure 1. Accuracy of content and style for different values of `style_weight` (X axis is log scale)

visible way and the two are in contrast with one another. Equilibrium is reached around `style_weight=50k` (setting `content_weight=1`, as mentioned before only the ratio matters).

When style weight is small, content is easily classified but style is not (accuracy of 0.25 means pure luck since there are 4 styles) and as style weight is increased we see content being much more challenging to identify, yet style more easily identified, reaching up to 0.74 accuracy.

We find that at the equilibrium the style transferred images are visually pleasing, they seem to preserve a good balance of style and content and therefore this is our answer to automating this process.

### 4. Further Analysis

We first must systematically check whether what seems like preservation of content to the NN is indeed considered as preserved by humans. We then enrich our result with yet a different kind of classifier, a Gram-matrix based classifier. Lastly we deepen our analysis of our content accuracy classifier.



Figure 2. Left chicken - accurately classified content by human and NN. Right - Same chicken image with `style_weight=30M` incorrectly classified by both humans and the NN classifier

style_weight	Humans accuracy	NN accuracy
1	1.0, 1.0	0.94
50k	0.8, 0.9	0.58
1M	0.7, 0.7	0.45

Table 3. Content Classification Accuracy of the NN classifier v/s two human classifiers aged 6 and 10 of the style transferred images.

#### 4.1. Do humans agree with the machine?

In this part we qualitatively assess whether when a NN says that style is preserved, humans agree with it and when a machine says that content is preserved, do humans agree with it? In general and in most cases we find that Yes. Images with low style weight seem to be completely “style-less”, e.g. they just look like the original image for humans and they are also in the general case incorrectly classified by the style classifier. Let us observe one example, in Fig 2 we see style transfer of the same content and style images but with different style weights. The chicken with `style_weight=1` is correctly classified as Chicken but style incorrectly classified as Renoir; the same image with `style_weight=30M` is barely recognizable by both a human and the machine (the NN incorrectly classifies it as a Dog and not a chicken), however the style classifier NN is able to correctly identify the style as Matisse.

We now systematically compare the accuracy of the NN content classifier for the style transferred images to that of a human classifiers (family members, young children aged 6 and 10). We do not ask the human classifier to classify the artistic style because we think this is too hard for them so they only classify the object class. Table 3 displays the result of this experiment, showing that as a trend, humans tend to classify correctly when style weight is low and incorrectly when it is high, albeit the humans are somewhat better classifiers than our trained NN.

#### 4.2. Gram Matrix based style classifier

In our project proposal we suggested using Gram matrices [12] of several convolution layers in order to classify

the style. The intuition is that if gram matrices are useful for NST’s style loss they might also be useful for style classification. We implemented a classifier that uses a single gram matrix of one of the convolution layers and which was able to classify the artistic style with similar accuracy to the aforementioned described simple classifier and is actually able to classify style transferred images with higher accuracy (up to 0.78 style accuracy, relative to 0.74). See Fig 3

Details of the gram matrix based classifier are interesting and will shortly follow, yet it is also instrumental to note that for the sake of displaying the trade-off between style and content in the transferred images which is the focus of our work, it does not matter that the overall accuracy of the gram-matrix based classifier is better, what we care more about is relative accuracy as it changes as a result of applying different parameters to style transfer.

To train the gram matrix based style classifier we started with a tuned VGG19 network as a baseline (simply because we found VGG easy to work with and modify) and fine tuned the network on the art training set. We removed its head and all layers following the 14th convolution layer and applied a gram matrix to conv14. The result of the gram matrix, after flattened was then fed to a FC layer with 100,352 unit and then applied a RELU to allow for non-linearity and then to another FC(128) with an output of 4 (number of classes). The number of parameters of this network is large and is due partially to the size of the gram matrix (100,352). This creates a challenge of training run-time and memory as well as risk of over-fitting given our relatively small dataset. However, this architecture, while may be improved, does prove effective, reaching accuracy of 0.88 on the test set and 0.78 on the style transferred images. Further optimizations can be made (regularization, improved training regime) but we find these results to be satisfactory for our cause.

Why was the 14th layer chosen? Different convolution layers were tried, most of them did not yield significant improvement over chance (accuracy of 0.25) but several did, of which 14 was best. We hypothesize, that layer conv-14 creates features that correlate well with the artistic style but we did not make further research on that since this was not the focus of our work.

#### 4.3. Content Classification - Performance Improvement

Prior to testing the classification accuracy of the NST processed images, a benchmark was created by using the accuracy of the content classifier, trained on the non-styled data. Its accuracy was 0.88.

To test the classifier functionality on the styled data, we first prepared the test set - converting partial animals dataset to different degrees of style (from 1 to 40K). The degree (weight) of style was chosen randomly in the range which



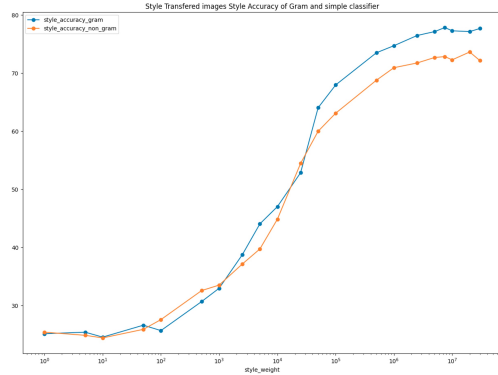


Figure 3. Accuracy of Gram matrix classifier and simple style classifier for different values of `style_weight` (X axis is log scale)

is easily classified by the human eye. The basic classifier achieved 0.67 accuracy on this dataset.

As a first step towards improvement, we trained the classifier on the styled training data. This resulted in performance improvement, up from 0.67 to 0.76.

The challenge with training a classifier on the style transferred images is that it is resource intensive, generating the images requires extensive processing. A cheaper solution would be to apply augmentation - manipulations such as: adding Gaussian noise, saturation / contrast / sharpness change (with random values) and mirroring, rotating, and switching RGB to BGR). We then performed multiple iterations, while on each iteration we doubled the training data, performed tuned training and tested performance on our styled test data. The results displayed in (Table 4).

Another hypothesis was tested, that the usage of silhouette or edges can improve content identification accuracy of the style transferred images. The motivation for that is based on the fact that different degrees of style application could impact the color and texture and that in turn could create unnecessary noise, for example usage of colors that do not exist in nature. We applied the edge filter on each image from the training data and performed another round of tuning of the model trained on style transferred data. The test results were not satisfying (accuracy 0.57), so we tried to observe the data and noticed that some of the examples still retain noise resulting from style even when edges are extracted. We then performed a round of image de-noising of the training set (Fig 4), retrained the model and tested it. The performance was better than in the previous round (0.67), but still worse than that of the classifier which was trained on augmented styled data. In conclusion, our hypothesis was proved incorrect, edge extraction did not improve object classification accuracy.

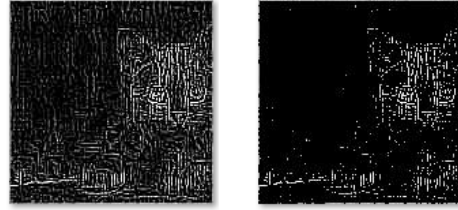


Figure 4. Left - cat styled image edges, right - same image went through de-noising process

Number of training images	Accuracy
250	0.76
500	0.785
1000	0.8
2000	0.807

Table 4. Results of using data augmentation in content classification

#### 4.4. Content Classification - score difference between styles

Despite the fact that for a general task of content classification we applied random style on each image, it is instructive to mention a side experiment showing that for a given style weight (in our case `style_weight=50k`), the application of different style images can lead to a difference in animal classification accuracy.

We started by applying a basic classifier (trained on the original animals-10 data) to a randomly styled (style for each image is chosen randomly) test dataset, which yielded 0.67 animal classification accuracy. Then two styles with natural colors (Rococo and Early Renaissance) and two non-natural styles (Popart and Abstract Expressionism) were applied to the same test dataset, respectively yielding success rates of 0.76, 0.7, 0.54, and 0.55. This result is consistent with the findings of [5] from assignment 2, which suggests that some classification networks may be strongly biased towards texture and color, and thus accuracy may be greatly affected by the distortion of natural colors.

## 5. Conclusion and Further Research

In this work we show how style transfer can be automatically tuned to find the right balance between the preservation of content and that of style. Future research may include improvement of the style classifier, in particular based on gram matrices or other techniques described in [8], [3] and [9]. In addition, we feel that further improvement in both style and content identification can be made using different augmentation techniques.

As an extended application of our findings we suggest

that the accuracy tradeoff charts be extended to include a “safe region interval” rather than the point estimate which we provided here, in which both content and style are preserved around the equilibrium point.

## 6. Work Division

Table 5 lists our work division in this project

Student name	Contributed aspect	Details
Xenia Kramarenko	Object classifier Data augmentation Measure content/style tradeoff Write report	Define content classification data and implement the classifier Test and report augmentation techniques Implement and measure independently content and style tradeoffs Done together
Ran Tavory	Style classifier Style classifier - Gram Style transfer Benchmark Human testing Write report	Define style data and implement the simple style classifier Implement the Gram matrix style classifier Implement style transfer Implement automation, run full data tests and create result charts Implement and report human testing Done together

Table 5. Work division

## References

- [1] Corrado Alessio. Animals-10 dataset on kaggle, 2019. [2](#)
- [2] Pancham Banerjee. Impressionist\_classifier\_data on kaggle, 2019. [2](#)
- [3] Yaniv Bar, Noga Levy, and Lior Wolf. Classification of artistic styles using binarized features derived from a deep neural network. In Lourdes Agapito, Michael M. Bronstein, and Carsten Rother, editors, *Computer Vision - ECCV 2014 Workshops*, pages 71–84, Cham, 2015. Springer International Publishing. [5](#)
- [4] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. [1](#), [3](#)
- [5] Rubisch Geirbos. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. 2019. [5](#)
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. [2](#)
- [7] Alexis Jacq. Neural transfer using pytorch, 2022. [1](#)
- [8] Jeremiah Johnson. Neural style representations and the large-scale classification of artistic style. 2016. [5](#)
- [9] Sergey Karayev, Aaron Hertzmann, Holger Winnemoeller, Aseem Agarwala, and Trevor Darrell. Recognizing image style. *CoRR*, abs/1311.3715, 2013. [5](#)
- [10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. [2](#)
- [11] Abhishek Singh. How to classify the paintings of an artist using convolutional neural network, 2020. [2](#)
- [12] Wikipedia contributors. Gram matrix — Wikipedia, the free encyclopedia, 2022. [Online; accessed 11-December-2022]. [4](#)

# Appendices

## A. Nice looking images

This is not part of the scientific report, still we thought it might be nice to show some of the visually appealing images we generated. Many beautiful images exist, of which for some reason cats stand out, maybe we are programmed to think they are cute.

Table 6 lists them along with their generating images

## B. Source code

Full source code for this project is available on our project page <https://github.com/rantav/cs7643DL-project>



























Style image	Content image	Style transferred
		
		
		
		

Table 6. Interesting and good looking images. All images were generated with `style_weight= 50k`, the point in which style and content accuracy are equal



Style image	Content image	Style transferred
		
		
		
		

Style image	Content image	Style transferred
