Improving the Neural Algorithm of Artistic Style with Adversarial Training

Daniel Shats (956919864), Topaz Aharon (305302127)

Technion – Israel Institute of Technology

Author Note

For 236860 Digital Image Processing Fall 2020-2021 Semester

Introduction

The problem of style transfer is one that researchers in Computer Vision and Digital Image Processing have been working on for a long time. In the days before deep learning, this field was known as "Non-photorealistic Rendering" and typically relied on classical methods of texture transfer to achieve something that looks like style transfer. But as the paper "A Neural Algorithm of Artistic Style" by Gatys et al. describes, they used mainly non-parametric techniques performing operations directly on the pixels of a given input, rather than higher level features, for example. This results in lower quality renderings than modern approaches have achieved using deep neural networks.

More recently, this idea of "Non-photorealistic Rendering" is called "Style Transfer" in Gatys et al. paper, and is also sometimes referred to as "Image to Image Translation" such as in "Unpaired Image-to-Image Translation using Cycle Consistent Adversarial Networks" by Zhu et al. But, sometimes image to image translation includes more than just style transfer in its field of use cases, such as super imposing zebra stripes on horses. So, for the purposes of this report, we will refer to "Non-photorealistic rendering" as "Style Transfer" and ignore the additional aspects of "Image to Image Translation".

Our (small) innovation in this topic comes in a fairly little-discussed aspect of this topic, in the form of model robustness. We were curious as to why exactly, after so many more accurate models have been released, VGG is still the best at neural style transfer (not including GANs). As it turns out, we don't really know, but there is a lot of evidence to support that the features which the newer, deeper models are learning (e.g. resnet, inception, etc) are actually not great features. This makes them very brittle to adversarial attacks. On the other hand, VGG is an

older network that seems to suffer less from this issue, leaving it with better capabilities for style transfer. We dive deeper into these ideas in this report.

Literature Review

Although we have cited all the various works we read which contributed to this report, it is a good idea to have a deeper dive into all of them. We will begin with the older methods used to tackle non photorealistic rendering, and then review newer methods that utilize deep learning.

To begin with, the oldest paper we cited, was “Paint By Numbers: Abstract Image Representations” by Paul Haeberli. This paper was quite funny to us because its main focus was on a interactive experience (a GUI of some sort) between a user and a program to create non-photorealistic renderings. These early papers are rather difficult to categorize as style transfer since style transfer typically (in todays literature) hints at doing it in an arbitrary sense. Meaning, in theory, one could take the style of any artist or artwork and apply it to a real photograph. But, older research such as Haeberli’s concentrates more on applying a particular style to an image. Moreover, in this case, its up to the user to apply a style. For example, one could choose to apply some paint strokes in particular areas of the canvas to create an impressionist effect to their sample image. His algorithm then applies the chosen strokes using some simple hyperparameters to achieve a desired level of the effect. Haeberli’s work goes further to demonstrate that there are ways to automate some of these artistic processes, such as finding gradients in the original image after applying a smoothing filter and using the gradient direction to control the application of brush strokes. Again, this can really only be used to inject a few chosen styles into a painting, and definitely is limited in its scope.

A slightly newer paper, from SIGGRAPH 2002, takes a different approach while still using a human-in-the-loop way of constructing styled images. The paper “Stylization and Abstraction of Photographs” by DeCarlo and Santella uses an ingenious method of tracking a persons eyes while they are looking at a photograph to intelligently determine which areas of the image are the most “meaningful”. They go on to describe how useful this is in terms of marketing for example, where its best to maximize only the most useful elements of an image to a consumer, presumably due to the limited time watching, and smooth out all that is unnecessary, as in a silhouette. This results in something very “artsy” though once again the method is not very flexible and not true style transfer. The final paper we’ll discuss which doesn’t really fall into the realm of arbitrary style transfer is “Non-photorealistic Image Processing: an Impressionist Rendering”. This paper is the first which presents a novel and fully automated approach to producing fairly high quality artistic renderings of real images. It does this by sampling a random set of pixels from the original canvas and then substituting them with color spots. Then, their algorithm covers the canvas with these spots. It’s a very simple algorithm and does a fantastic job at creating impressionist paintings and pointillism.

The first paper we reviewed that offers a true arbitrary style transfer (though maybe not a very good one, by todays standards) is “Fast Texture Transfer” by Ashikhmin. In this paper, the author delves into the fact that for previous tasks of texture synthesis, there exist fairly dependable criterions for success (like loss functions) which can accurately describe how well your algorithm has done. But, in the case of style transfer (i.e. synthesizing two images), this criterion is much less quantitative, relying on user preference. We infer that perhaps this is why it took so long to get style transfer working with neural nets, where we must explicitly write loss functions upon which to calculate a gradient and update the parameters of a model with. In this

paper, an optimized version of an already existing algorithm is shown. Specifically, the coherent synthesis technique. As he describes, it basically scans the content image, chooses a small neighborhood around each pixel (L-shaped, for some reason), searches the style image for similar neighborhoods (based on some criterion like L2 distance), and synthesizes a new image with that neighborhood. It's a fairly simple method, and this process was described in greater detail in his paper "Synthesizing Natural Textures." Though the results of his research were fascinating for the time, now we have much more powerful methods, which can synthesize style transfer much closer to our abilities as humans, using models that much more closely model our brains (granted, likely still very far from actually mimicking our brains).

Now, well discuss an area of research even more recent than the work we built on for this paper. This is the area of Image to Image translation. Specifically we will discuss the work of "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks" by Zhu et al. This paper is quite famous for the fact that it blew away nearly every other method for things like texture and style transfer, but also introduced a new set of problems such as changing zebras into horses, and set a very high bar as a baseline for that as well. CycleGANs are currently considered state of the art in the fields the paper introduced, as we can see in the computational painting Kaggle competition: "I'm Something of a Painter Myself" found here: https://www.kaggle.com/c/gan-getting-started/overview. The way they work is by leveraging a few revolutionary ideas in deep learning that have risen in the last few years. The main idea is using a Generative Adversarial Network (GAN). These were introduced by Goodfellow et al. in 2014. Basically, There are actually 2 networks. One is a discriminator (which classifies whether or not a sample comes from the real data distribution) and another is a generator (which produces these samples that hopefully lie in the real data distribution). Although very interesting, GANs

suffer many issues for their great complexity (relative to a regular CNN). One of these issues is mode collapse. In this case, the model finds one image which is readily classified by the discriminator as perfectly within the data distribution, and maps all (random noise) inputs to this sample. This makes the update gradient for the generator 0 and its parameters no longer change, leading to phenomenon known as mode collapse (and as we learned in 236781 Deep Learning, mode collapse is very easy to get). The solution that Zhu et al. found was to impose a cycle-consistency loss. The paper explains this concept nicely with math, but well use words. In any case the formula is below:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = &\mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &+ \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].\end{aligned}$$

Basically, the cycle consistency loss creates explicit need for a bijection from a sample in the latent space to a generated sample. That is to say, it enforces the fact that one generated sample cannot map to multiple samples in the latent space. Although this mostly alleviates the mode collapse issue, it increases the computational complexity of the model significantly. This is because now the model doesn't look like a regular GAN, but more like two transformers. Traditional GANs were already quite finnicky to train, never mind the mode collapse issue. Now with the added parameters, the issues could be even worse. This is part of the reason that we wanted to go back to an older method of doing style transfer. Even though the use of GANs is fun, they may not be the best solution to the problem due to their inherent instability.

A Review of "A Neural Algorithm of Artistic Style"

This is the paper by Gatys et al. which we decided to review and design an improvement for. But, before we get to the improvement, let us do a review of the paper and analyze some

details about their method. We will also discuss some work regarding adversarial training of neural networks to precede our improvement.

So, let's begin with the review. Gatys paper is a wonderful introduction to using deep learning methods to model style transfer. As far as we know, there were a plethora of ways to do classification, regression, prediction, etc. with deep learning models, but something like style transfer hadn't yet been successfully attempted. His paper basically deconstructs a popular Convolutional Neural Network at the time, VGG-19. We will discuss why this is important and why this hasn't been improved upon a little later.

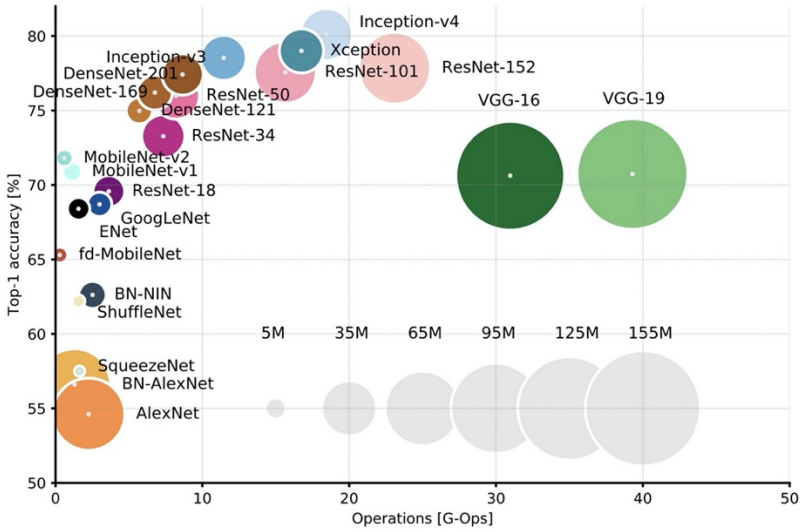
The paper begins with a wonderful, though brief, description of how humans use our visual cortex to create art and do other things like object detection, classification, etc. Or it doesn't do it at all, because the truth is we really don't know (and the paper admits this). Then the paper goes into fairly fine detail about feature hierarchy and what convolutional nets actually learn in each of their layers, as opposed to treating them like black boxes. This is important because the entire method relies on the ability of a CNN to learn different features for style and content of an image at different layers. The authors define two very interesting and critical loss functions to learn a separate representation for the style of an input and the content of an input. Its really crazy to us that we can define something as analytical as a loss function to measure something as abstract as the style or content of an image. And moreover, its as simple as squared error! But as we researched and learned more, this is not really the case. We believe this is a major Achilles heel in this paper as it masks what these losses actually are. For example, here is the content loss:

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^{l} - P_{ij}^{l} \right)^2 .$$

It is not a “content” loss in the abstract meaning of the word content. It is simply a loss that works to minimize the difference between the activations of some original image and some white noise image at a particular layer in the network. The authors **hope** that this learns the “content” representation **if** they use the right layers of the network, but by no means is this necessarily the case. In fact, we find that this may be exactly the reason why newer, deeper networks like ResNet do much worse with this same method. These deeper networks seem to learn much more complex (and less robust) features that don’t easily separate what the authors describe as “style” and “content”. We think the authors should have made this clearer, as it may have prevented this fantastic idea to be shadowed by the use of GANs. We believe that perhaps there is more to be found here. We elaborate on this in our suggested improvement (next).

Suggested Improvement

So when we first chose this paper, we figured this is going to be easy. Since the paper was written using an older, shallower architecture from a few years ago, we will just apply the same method with a better performing architecture (such as ResNet or Inception, for example), and call it a day. Those architectures do much better on ImageNet than VGG:

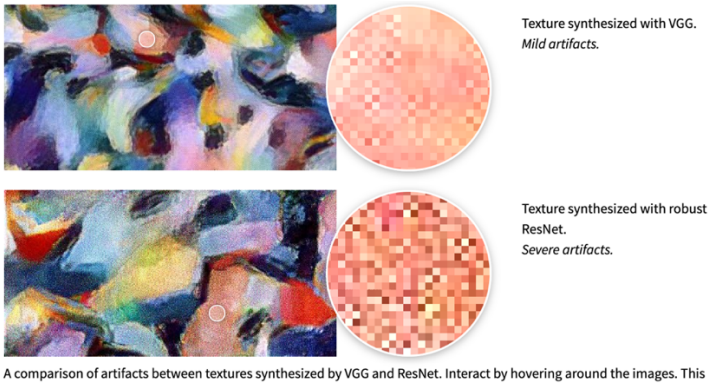


Moreover, they do so with less operations. So one would reason that they are probably learning better features, and are better at separating the content of an image from its style. Well, one would also be wrong. The article “Neural Style Transfer With Adversarially Robust Classifiers” by Reiichiro Nakano was critical in helping us reach this conclusion. In it, he did exactly what we suggested and compared the results of ResNet vs VGG in style transfer and found VGG to be much better. In it, and in the paper “Adversarial Examples Are Not Bugs, They Are Features” by Ilyas et al, a concept known as robust and non-robust features are described. These are features which in theory, humans use to do things like classification. If you show a human an image of a turtle, there is no real way to add any level of noise to that image to the point where its unrecognizable to one human, yet everyone else thinks it’s a turtle. But this is the case in a non-robust network. Its easily fooled just by adding some carefully chosen (and totally undiscernible to the human eye) noise. You can make a turtle look like a gun, or a car look like whatever you can imagine. Madry’s Lab at MIT does a lot of work in this field. The blog post from Reiichiro presents this graph to visualize “how well a particular architechture is able to capture non-robust features in an image.” Clearly, VGG is much worse at this task, and so in theory, it has managed to capture much more robust features.

This theory is greatly supported by Reiichiro’s work. Using Madry labs “robustness” library, he compared a ResNet50 trained on ImageNet using standard training procedure to a ResNet50 trained on a smaller version of ImageNet, but with adversarial training from the robustness library. Even after being trained with a much smaller dataset, it is clear that the robustly trained ResNet performs significantly better in style transfer than the other:

Since adversarial training of deep neural networks is a research field in and of itself, describing exactly the method is out of the scope of this paper, but we will give a basic overview of what it is. In essence, we want to add to the training dataset all of the small permutations of the original images which result in an image that would still classified be by humans as its original label, but makes the model believe its something totally different. Once a strong set of these images are added to the training loop, the model is less likely to find non-robust features to rely on to strengthen its accuracy.

One last important thing to note is that the author of the blog post shows that even after training a robust ResNet, its results still leave a lot to be desired in the domain of style transfer when compared to a VGG with regular training. As pictured, there are still quite a lot of artifacts which may simply be an inherent trait of the residual architecture, or something else (we really don't know). See the artifacts here:



A comparison of artifacts between textures synthesized by VGG and ResNet. Interact by hovering around the images. This diagram was repurposed from Deconvolution and Checkerboard Artifacts by Odena, et. al.

Our approach was really quite simple. We wanted to improve neural style transfer using just one network and realized that just using a more modern network architecture wasn't going to work. Every attempt at doing so failed. That doesn't mean that there aren't network architectures much better suited than VGG for the task, but rather we haven't been looking for them, so VGG is the best we have. Moreover, all the VGG experiments available have only used a VGG trained using the standard training procedure. Although VGG is fairly robust, we figured "More is always better, right?" So, we made the already robust classifier even more robust by training it with the robustness library.

Some caveats of our training procedure and results are that we didn't really have access to the power required to train with ImageNet (though that would have been ideal) and also had trouble downloading ImageNet. So, we settled for CIFAR10. Although this is a much smaller dataset, we can still extrapolate meaningful information from our results and if Professor Alex would like, we would be happy to re-run our code on the course servers and see if our results stay consistent when run on more powerful hardware with a bigger dataset.

Results

Please note that our results are not nearly as good as the results from the original paper by Gatys et al, but we believe that if we had the compute power to train our model adversarially on the whole ImageNet, then we would have state of the art for single network neural style transfer (i.e. not using GANs). Another thing to note about our experiments are that we used the VGG implementation found here: https://github.com/kuangliu/pytorch-cifar/ as opposed to the original implementation. The difference is that there are now batch norms, and average pool at the end, and no dropout. These are small differences that have been accepted as almost universally better in modern research. Using the following famous images for style and content, respectively, our

results are below. In the following figure, we can see the results of training VGG using a standard training procedure (left) VS using adversarial training (right).

Regular Training Adversarial Training

Though we only trained for 100 epochs, it is clear that even with the lack of a large dataset, the robustly trained model does a significantly better job at capturing the style of the

artwork than the regularly trained one. The regularly trained model also seems to have some sort of failure after more than 75 epochs and gives strange results (we haven't been able to reason exactly why, though it could be due to capturing bad features from overfitting). But, clearly looking at just the 50 epoch mark, the robustly trained model is doing much better.

Suggestions for Future Work & Closing Remarks

Although it is possible that our team has achieved state of the art in the field of single network neural style transfer, we do not believe this is the most important thing we have learned. More importantly, is the direction which modern deep learning is going. Currently, models like ResNet are benchmarking higher than humans on ImageNet. Although this seems like a great thing, we need to be more cautious as to why and how they are doing this. It seems like ever since super deep networks have come about, every new iteration is less and less robust to adversarial attacks. They are learning worse features that can be easily exploitable by an adversarial agent. In a closed loop experiment, this is not necessarily a bad thing. But when we consider the fact that we are putting these highly complex black-box models in autonomous driving systems for example, we have no guarantee that somebody won't try to take advantage of their fragility. In fact, we should assume that people will.

The issue lies in the fact that researchers aren't really getting credit for building more robust architectures. They get credit for building algorithms that achieve 0.000001% better accuracy than the previous state of the art. But in the real world, we see this goal as basically useless. Morover, maybe if we focus on building models that learn better features, we have a better chance at achieving the crowned jewel that is general artificial intelligence.

Coming back to the task at hand though, we see a direction forward. We believe adversarial training is only part of the puzzle. It was a fluke that VGG is so good at doing style

transfer. We need to come up with better architectures that learn better feature of the data. This is a very tough problem to solve, but if we want better, more reliable networks that learn more like humans do, this is the only way forward.

References

[1] : https://arxiv.org/abs/1508.06576 {gatys2015neural, title={A Neural Algorithm of Artistic Style}, author={Leon A. Gatys and Alexander S. Ecker and Matthias Bethge}, year={2015}, eprint={1508.06576}, archivePrefix={arXiv}, primaryClass={cs.CV}}

[2]: @article{Haeberli1990PaintBN, title={Paint by numbers: abstract image representations}, author={P. Haeberli}, journal={Proceedings of the 17th annual conference on Computer graphics and interactive techniques}, year={1990} }

[3]: Doug DeCarlo and Anthony Santella. 2002. Stylization and abstraction of photographs. ACM Trans. Graph. 21, 3 (July 2002), 769–776. DOI:https://doi.org/10.1145/566654.566650

[4]: Sparavigna, Amelia Carolina & Marazzato, Roberto. (2009). Non-photorealistic image processing: an Impressionist rendering.

[5]: Ashikhmin, N.. (2003). Fast texture transfer. Computer Graphics and Applications, IEEE. 23. 38 - 43. 10.1109/MCG.2003.1210863.

[6]: Michael Ashikhmin. 2001. Synthesizing natural textures. In Proceedings of the 2001 symposium on Interactive 3D graphics (I3D '01). Association for Computing Machinery, New York, NY, USA, 217–226. DOI:https://doi.org/10.1145/364338.364405

[7]: "Generative Adversarial Networks." Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. ArXiv 2014.

[8]: @article{nakano2019a, author = {Nakano, Reiichiro}, title = {A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features': Adversarially Robust Neural Style Transfer}, journal = {Distill}, year = {2019}, note = {https://distill.pub/2019/advex-bugs discussion/response-4}, doi = {10.23915/distill.00019.4} }

[9]: @misc{robustness, title={Robustness (Python Library)}, author={Logan Engstrom and Andrew Ilyas and Hadi Salman and Shibani Santurkar and Dimitris Tsipras}, year={2019}, url={https://github.com/MadryLab/robustness} }

[10]: @TECHREPORT{Krizhevsky09learningmultiple, author = {Alex Krizhevsky}, title = {Learning multiple layers of features from tiny images}, institution = {}, year = {2009} }

[11]: Jun-Yan Zhu*, Taesung Park*, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks", in IEEE International Conference on Computer Vision (ICCV), 2017.