

Delusional Jellyfish



Developed by Tanish Baranwal and Andrea Jia

[Click here for our Cloud 9 workspace](#)

Table of Contents

Table of Contents	1
Brainstorming	2
Explanation of Solution	3
Our Results	5
Gallery Walk	6
Conclusion	7
Daily Log	8
Image Log	11
References	12

Brainstorming

- Our client - Client # 4: A Developer
 - Create a script to manipulate images into one composite in a unique way, ideally one not possible through simple Photoshop workflow
 - A photomontage made from at least two distinct original images
 - how to incorporate geometric shape into the image – drawn on, as a border, or as a mask.
- Ideas:
 - Neural Style Transfer [Tanish will head the algorithm while periodically explaining to Andrea]
 - Object detection
 - Image captioning
 - Algorithmic image redesigning
 - for example, crystallization by making colors geometric shapes
- Sketch:



- Pair feedback:
 - Another pair recommended a few things we could add to our project. For example, they suggested making a border for our modified image, that could possibly be styled with the style image along with the content image. Additionally, we could create a watermark for the images, or also create our own logo to put in the corner after modifying images. The pair also made a suggestion of selectively styling - for example, styling only the background of the content image but not the main object itself. We are considering this option, however, the code required to do this requires image segmentation strategies that are beyond the scope of this project, and the image output may not be as visually appealing as our original idea.

Explanation of Solution

- Using the activations of intermediate layers in [VGG19](#) pretrained weights to extract the information from the content and style images, the output image was optimized to minimize 3 types of loss:
 - Content Loss: Defined as the reduced squared difference between the content image's pixels and the output image's pixels.
 - Style Loss: The reduced squared difference between the style image's gram matrix and the output image's gram matrix.
 - The gram matrix is made by multiplying the image matrix by the transpose of it.
 - The gram matrix is used because the style of an image is not dependant on the pixel values but the relationship between the pixel values; the gram matrix de-localizes all of the information about the style image, such as texture, shapes, and weights: exactly what style is.
 - Total Variation Loss: The variation of the images pixels. The purpose of this is to smoothen out the output image and reduce the "graininess" of the image.

content loss

$$\ell_{feat}^{\phi,j}(\hat{y}, y) = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2$$

style loss

$$\ell_{style}^{\phi,j}(\hat{y}, y) = \|G_j^{\phi}(\hat{y}) - G_j^{\phi}(y)\|_F^2.$$

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'}.$$

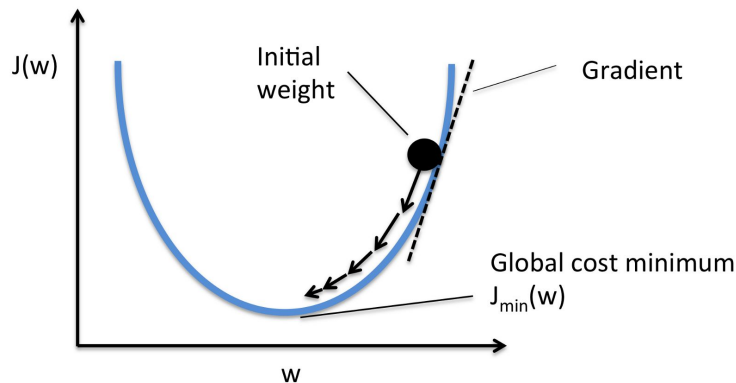
total variation loss

$$\begin{aligned} E_{\mathcal{U}}(\mathbf{w}) &= \sum_{x_k \in \mathcal{I}} \Theta(\bar{f}(x_k; \mathbf{w})) = \sum_{x_k \in \mathcal{I}} |\nabla f(x_k; \mathbf{w})| \\ &= \sum_{x_k \in \mathcal{I}} |\nabla_X f(x_k)| + |\nabla_Y f(x_k)|. \end{aligned}$$

[Source](#)

- The losses were optimized using the Adam Optimizer, which optimizes the output image by subtracting the scaled partial derivative of each loss function in respect to a pixel from that pixel. On a high level, it minimizes the output image

to the minimum by slowly driving the values down hills in the function to the optimal point.



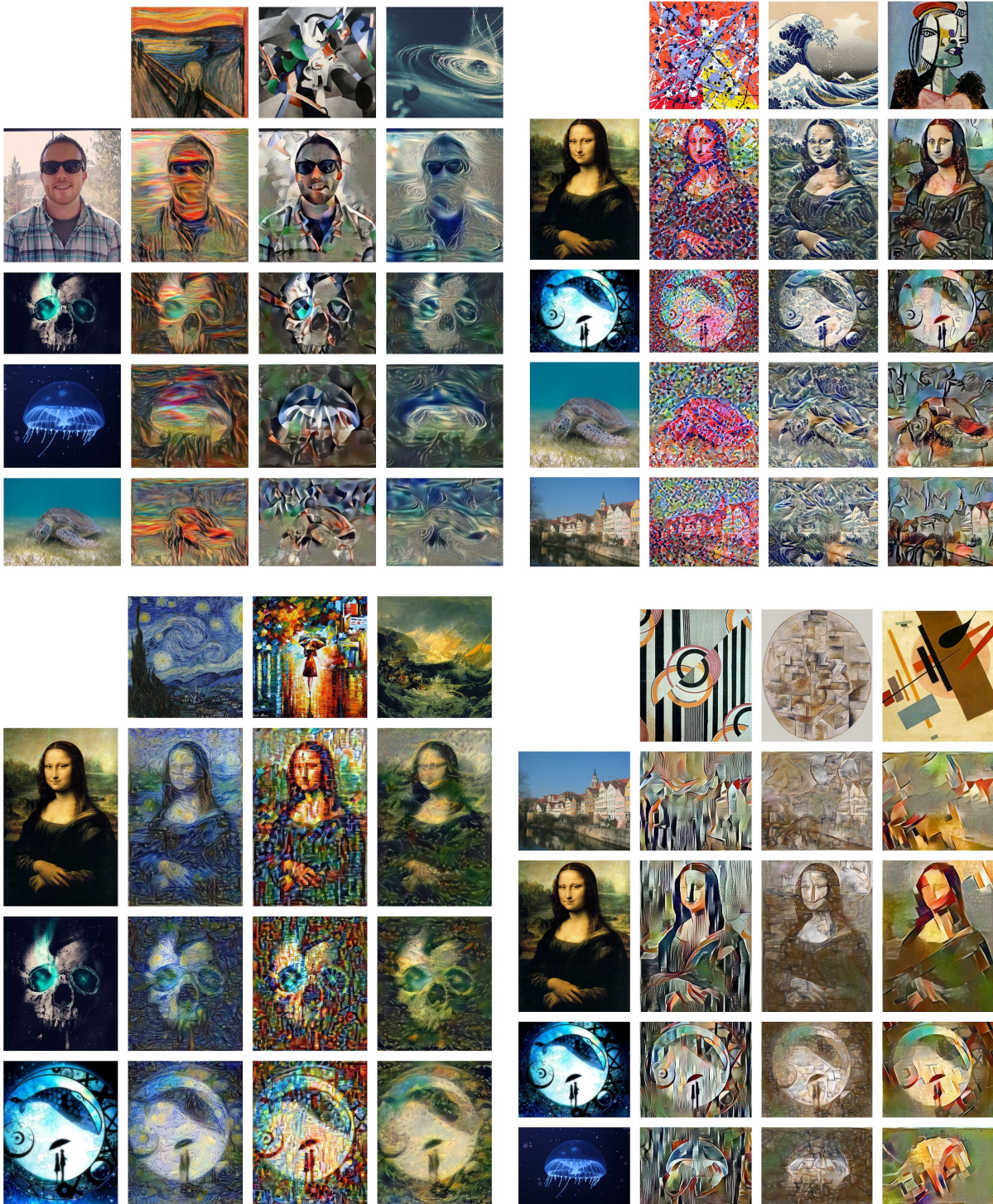
[Source](#)

- Evaluation:
 - Our solution to our client's need works to take two images, one content and one style, and combines them in a way that keeps the main objects of the content image and portrays them in the way of the style image. Additionally, we incorporated geometric shape by creating a border that changes depending on the image used.
 - Strengths:
 - This solution can take any kind of image file and image shape to combine content and style.
 - The border is flexible in that it adjusts depending on the size of the image.
 - The program allows interaction from client by providing several options, including the images they want to edit and the resolution they want.
 - The user has complete freedom about what kind of style and content image to choose and complete freedom on how the image is generated.
 - Weaknesses:
 - The code requires an extended amount of time required to run on the image(s) due to technological limitations.
 - Some content did not transfer correctly when combined with certain style images, resulting in a product image with vague outlines.
 - Some style images tend to work better than others, and may not produce a satisfactory product.
 - Highly time consuming on cloud 9
 - Improvement: This project can be extended to use Generative Adversarial Networks or Restricted Boltzmann Machines to eliminate the content and style images and allow the algorithm to generate new images, but these techniques are beyond the scope of the project.

Our Results

(columns are content images, rows are style images, and the intersection shows the content image painted in the style of the style image)

[Border not visible due to image size and cropping, for border see image on cover image and slideshow]



Gallery Walk

[Click here for our slideshow,](#)

and read our technology explanation below!

Type: `./modify_images_windows.py --interactive`
 (if you need help, type `./modify_images_windows.py --help`)

*Note to Mr. Brown for grading: Code will take quite a long time to run on Cloud 9.
 To run the code in ipython, type "activate" in the bash terminal to start the virtualenv
 with the required python packages to run all of the code.*

Pros	Cons
<ul style="list-style-type: none"> - Rather impressive I like how it is all interchangeable like each style can go to any content and etc. - Super complex code - Tons of different filters and shading stuff - The most artistic project I've seen so far, makes it look like a painting. - Lots of geometric creativity. x2 - Good explanation of equations used above - Code has documentations of comprehensible length - Some very good looking jellyfishes - it makes it look like a painting - Overall your code is very impressive. I think it is really cool how you blended the images together and it overall looks amazing! 	<ul style="list-style-type: none"> - Takes too long to load - Multiple code? - Very difficult to understand coding x3 - There is an invalid syntax - Bad smiley face at the front of the presentation - it was a great project but smiley faces are like this: :) not :] - santa claus - Where would this be used? (application) - For the brainstorming sketch I don't think you should use an image from your final project.

Conclusion

Andrea	Tanish
<p>The team dynamic for this project was good. We split the work of code, documentation, and presentation between the two of us to complete the project at maximum efficiency. By working like this, we both improved by learning from the other's strength, with Tanish's being analysis of complex algorithms and mine being attention to detail. Our design process was facilitated by open discussion during class and voicing our individual opinions to settle on a compromise. Areas for improvement include cooperating more on coding so that both receive equal understanding of the code. Next time, we could refine our ability to pair program by periodically switching roles in order to enhance each person's contribution to the program as well as their understanding of its core algorithms. Our team dynamic was also significantly improved by analyzing the way we worked together and increased communication throughout the project.</p>	<p>Our team dynamic was good. I handled the technology and the algorithm part, while Andrea focused on the user interface. We worked well together, discussing issues along the way. Splitting up the work this way allowed each of us to work to our strengths and complete the topic as efficiently as possible. Our design process involved discussing on the visual appeal of the image produced to decide on the optimization of the algorithm parameters by combining our opinions and compromising. On the documentation side, I focused more on the content while my partner focused on the overall layout and the details. Areas for improvement include collaborating more on the coding, so that both people enjoy an equal perception and understanding of the code. We could ameliorate this by switching roles of programmer and overseer more often to incorporate each person's unique coding style.</p>

Daily Log

See version history in git commit log:

<https://github.com/tekotan/csp-neural-style-transfer/commits/master>

Note: To view complete version file, go to the github link and click "View File"

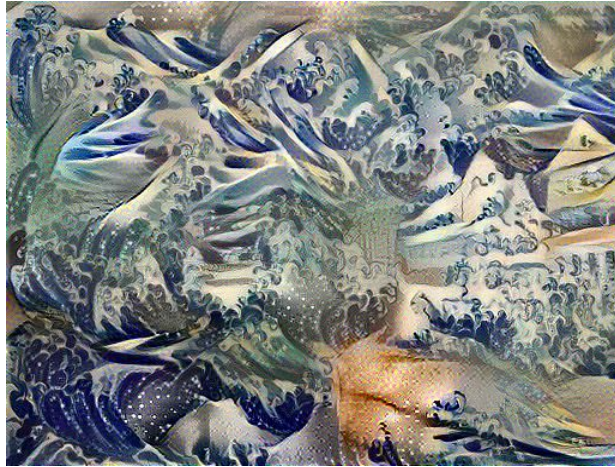
Day	Tasks	Team Reflection:
Tuesday 2/26/2019	<ul style="list-style-type: none"> - Began documentation and its outline - Brainstormed ideas - Decided on a reference for project idea - Studied reference - At home, Tanish finished the pipeline for image conversion 	<ul style="list-style-type: none"> - We worked well together
Wednesday 2/27/2019	<ul style="list-style-type: none"> - Finished and tested basic code 	<ul style="list-style-type: none"> - We worked well together
Thursday 2/28/2019	<ul style="list-style-type: none"> - Added Total Variational Loss to the code and tuned some of the loss weights. - Added code for running the function through all images in a directory - Version 1 of .py 	<ul style="list-style-type: none"> - We worked well together
Friday 3/1/2019	<ul style="list-style-type: none"> - Received pair feedback from another group - Version 2 	<ul style="list-style-type: none"> - We worked well together
Monday 3/4/2019	<ul style="list-style-type: none"> - A bit of explanation of the core concepts that allow the system to transfer style by Tanish - Searched up content and style images and cited them as references 	<ul style="list-style-type: none"> - We worked well together
Tuesday 3/5/2019	<ul style="list-style-type: none"> - Added docstrings to the code and ran pylint to check for code cleanliness - Added output to the program for user to see - Ran into a problem where the code failed to save the modified 	<ul style="list-style-type: none"> - We worked well together

	image and saved the original image instead; the problem was fixed by pulling old code from GitHub	
Wednesday 3/6/2019	- Began formatting and content for art gallery slideshow	- We worked well together
Thursday 3/7/2019	- Ran code on images to make final products	- We worked well together
Friday 3/8/2019	- Andrea was absent due to being sick - Tanish continued to run code and put final image products on the slides	- Andrea was out sick
Monday 3/11/2019	- Finish making collages for content, style, and converted images	- We worked well together
Tuesday 3/12/2019	- Generate high quality image conversions for slide and documentation - Tanish added a description of the general logic of the code and the math behind it	- We worked well together
Wednesday 3/13/2019	- N/A (block schedule)	- N/A (block schedule)
Thursday 3/14/2019	- Made a border by magnifying the original image and pasting the original image onto the center of the magnified image - Updated image log - Version 3	- We worked well together
Friday 3/16/2019	- N/A (no school)	- N/A (no school)
Monday 3/18/2019	- Added CLI (Command Line Interface) to the code - Fixed module and function docstrings	- N/A (no school)

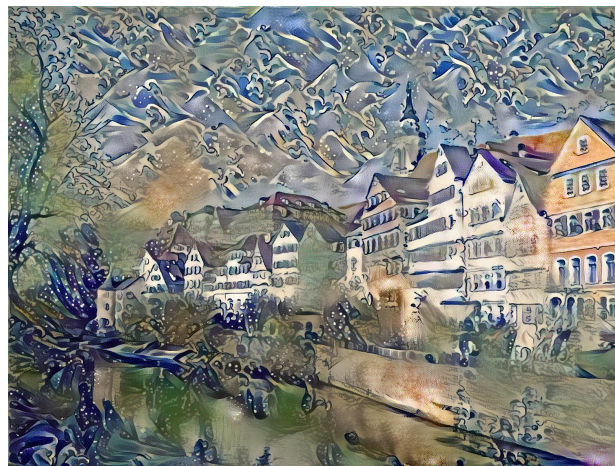
Tuesday 3/19/2019	<ul style="list-style-type: none"> - Refine slideshow images to include newly added borders - Completed the reflection questions by thinking about the team dynamic together 	<ul style="list-style-type: none"> - We worked very well together
Wednesday 3/20/2019	<ul style="list-style-type: none"> - N/A (block schedule) 	<ul style="list-style-type: none"> - N/A (block schedule)
Thursday 3/21/2019	<ul style="list-style-type: none"> - Finalizing code and documentation - Finished adding last images to slideshow 	<ul style="list-style-type: none"> - We worked well together
Friday 3/22/2019	<ul style="list-style-type: none"> - Review of code and logic behind the algorithms - Finalized documentation content and formatting - Created gif for title page display so that both original and new images can be shown 	<ul style="list-style-type: none"> - We worked well together

Image Log

First try: images are combined, somewhat messy (city looks like it is being demolished by the great wave)



Second try: The resizing was changed



Third try: a border was made by magnifying the image itself



References

Inspiration:

https://github.com/tensorflow/models/blob/master/research/nst_blogpost/4_Neural_Style_Transfer_with_Eager_Execution.ipynb

Images:

Content	Style
<ul style="list-style-type: none"> - Turtle - Tuebingen - Mona Lisa - Mr. Brown - Crystal Jellyfish - Tanish's Profile Picture - Andrea's Profile Picture 	<ul style="list-style-type: none"> - Great Wave - Starry Night - Planet/Atom Wallpaper - Picasso Untitled XI - Splatter Painting - Udnie - The Shipwreck of the Minotaur - Rain Princess - The Scream - Liubov Popova - Malevich Suprematism - Cubism