EE 148 Final Project
Exploration and Improvements of Neural Style Transfer

Andrew Kang

1 Introduction

Until recently, computers have not been able to emulate the fundamental style of different artistic works as well as humans due to the complex symbolic nature of such works. A recent development that makes use deep neural networks to transfer the style of one reference image onto another reference image has shown that an algorithmic approach to art style is possible. We explore this development and experiment with optimizing parameters to visualize the inner workings of this process. We also add our own methods to the existing work, namely better initilization, multi-style transfer, and segmented style transfer.

2 Previous Work

In the paper *A Neural Algorithm of Artistic Style* by Gatys et al., the authors accomplish style transfer by taking two images' representations in the convolutional layers of a convolutional neural network pre-trained for object recognition, specifically VGG-Network. To obtain the content representation of an image, they take the responses of higher layers in the network. To obtain the style of an image, they use an original feature space that measures the correlation between responses in a layer.

To accomplish this, they use loss functions to achieve the content and style representations. A variation loss is also used in the literature, in which the image is shifted and the norm of the difference is calculated, to make the image smoother. These loss terms are weighted and summed, and gradient descent is performed from a white noise image to get a final image that minimizes the total loss, which we refer to in this paper as the "combination image".

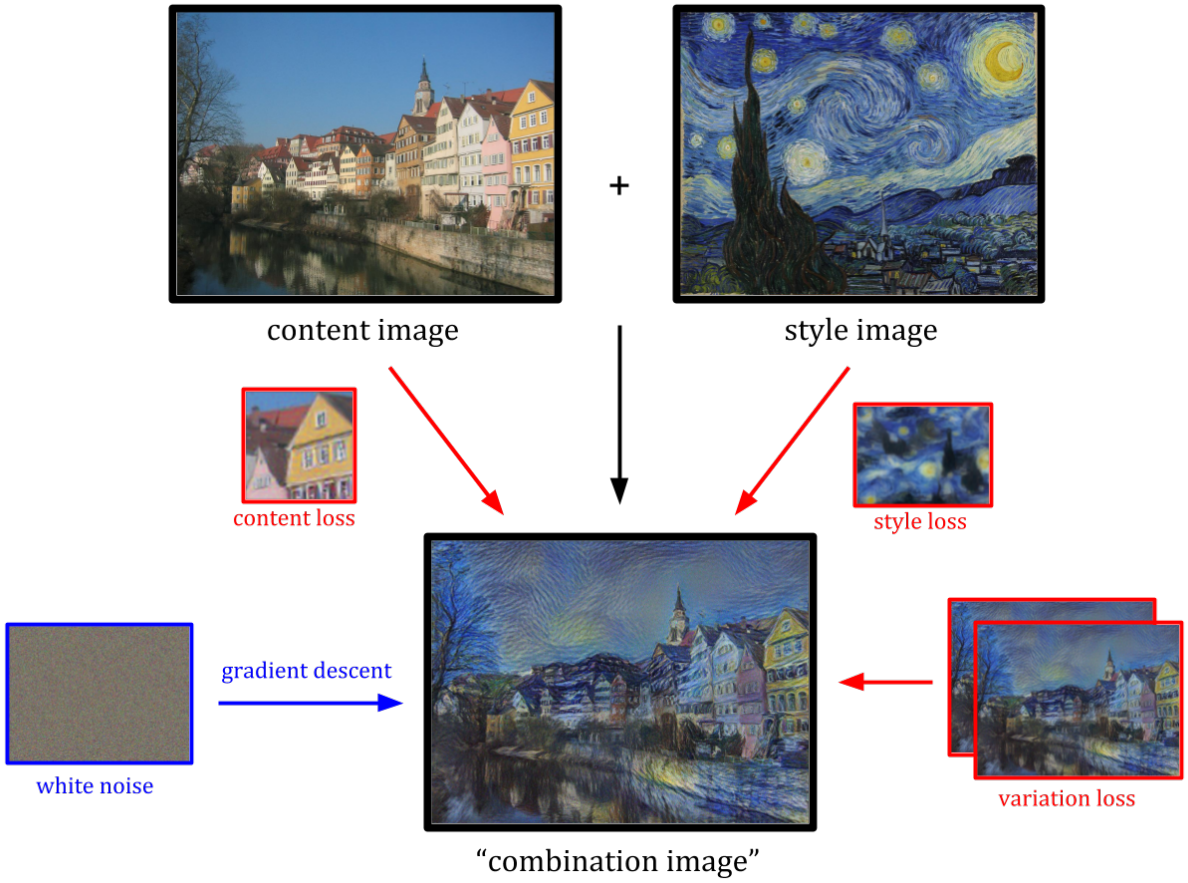

Figure 1: The style transfer process. A total loss function is defined, consisting of the content loss (for the reconstruction of the content image), the style loss (for the style transfer itself) and the variation loss (for a smoother final image). Then gradient descent from the white noise image is performed to get the final image that minimizes the loss.

3 Approach

We adapt the approach of Gatys et al. to formulate the loss function. We use the VGG19 network and use Keras and SciPy to implement the network and minimization, along with Keras' backend to calculate the gradients of the total loss.

As with the approach of Gatys et al., we weight the different loss terms. For instance, a greater style weight means that the final image will resemble the content image less and contain more of the abstract features of the style image. We refer to the weights as $w_{\text{content}}, w_{\text{style}}, w_{\text{variation}}$.

In our experiments, quality was determined by qualitatively analyzing three things:

- Aesthetics
- Image smoothness
- Resemblance to content and style images

4 Experiments

Initially, we experimented with the different weight terms to find the optimal parameters and visualize the inner mechanisms. For these experiments we used a photograph of the "Neckarfront" in Tübingen, Germany as the content image and *The Starry Night* by Vincent van Gogh as the style image.

4.1 Style Weight

The first experiment shows the results of increasing the style weight term - with greater weight terms, the image gets increasingly more distorted but features more of the stylistic blue and yellow swirls in *The Starry Night*.

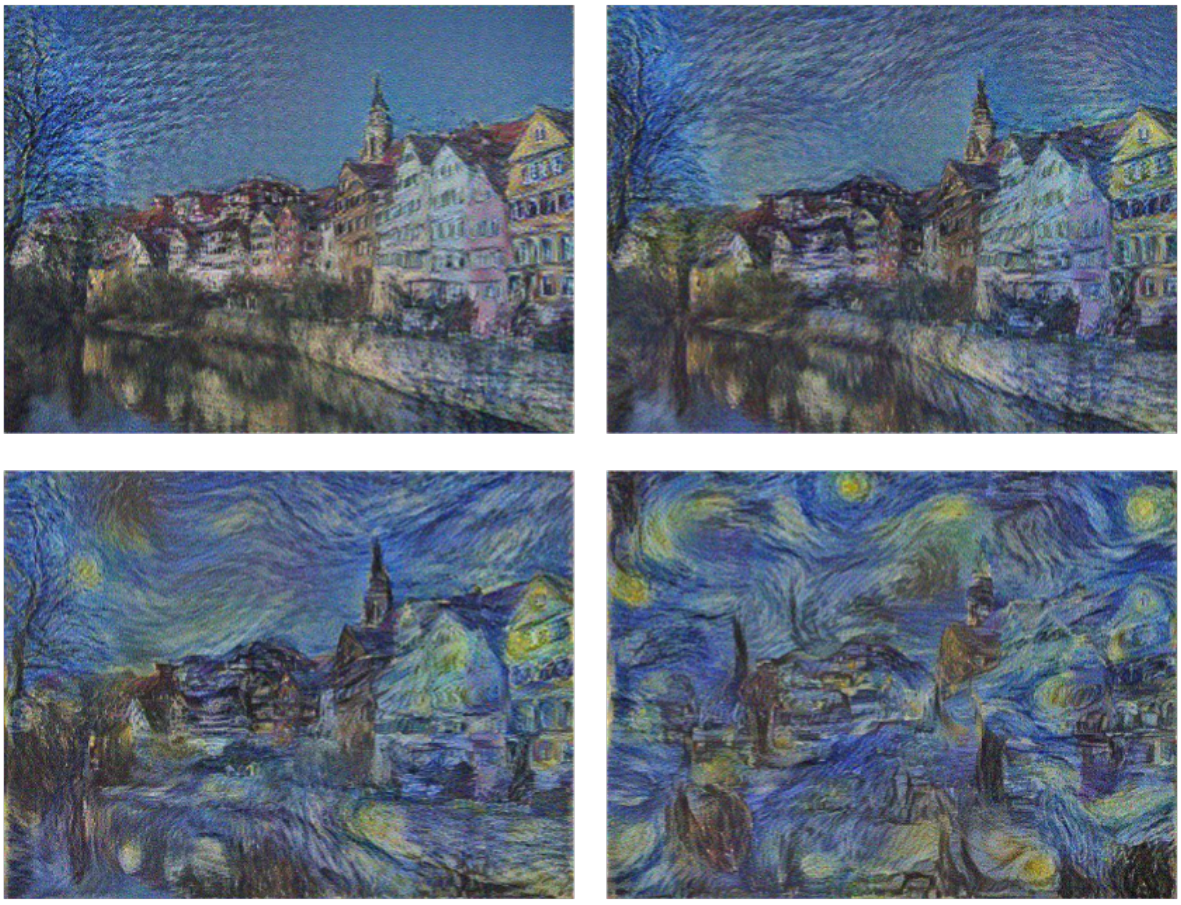

Figure 2: Results of continually increasing the style weight term by one order of magnitude. The greater the style weight, the less the final image resembles the content image and the more abstract features it gains. From left to right: $w_{\text{style}} = 10^2$, $w_{\text{style}} = 10^3$ (top row), $w_{\text{style}} = 10^4$, $w_{\text{style}} = 10^5$ (bottom row). Other parameters used: $w_{\text{content}} = 1, w_{\text{variation}} = 0$, iterations = 1000, height = 400.

4.2 Style Layers

The second experiment shows the results of adding increasingly more style layers. Again, with more style layers, the image becomes more distorted and more abstract.

Figure 3: Results of including 1 to 5 of the designated style layers. The weight on each of the n included layers was $\frac{1}{n}$. From left to right: 1, 2, 3 layers used (top row), 4, 5 layers used (bottom row). Parameters used: $w_{\text{content}} = 1, w_{\text{style}} = 10^4, w_{\text{variation}} = 0$, iterations $= 1000$, height $= 400$.

4.3 Variation Weight

The third experiment shows the results of increasing the variation weight term - with greater weight terms, the image gets increasingly smoother but blurred at the same time. Since the variation loss consists of calculating the norm of the differences in a horizontal shift and a vertical shift, much of the distortation becomes axis-aligned. From this experiment, the variation weight was determined to be unnecessary.

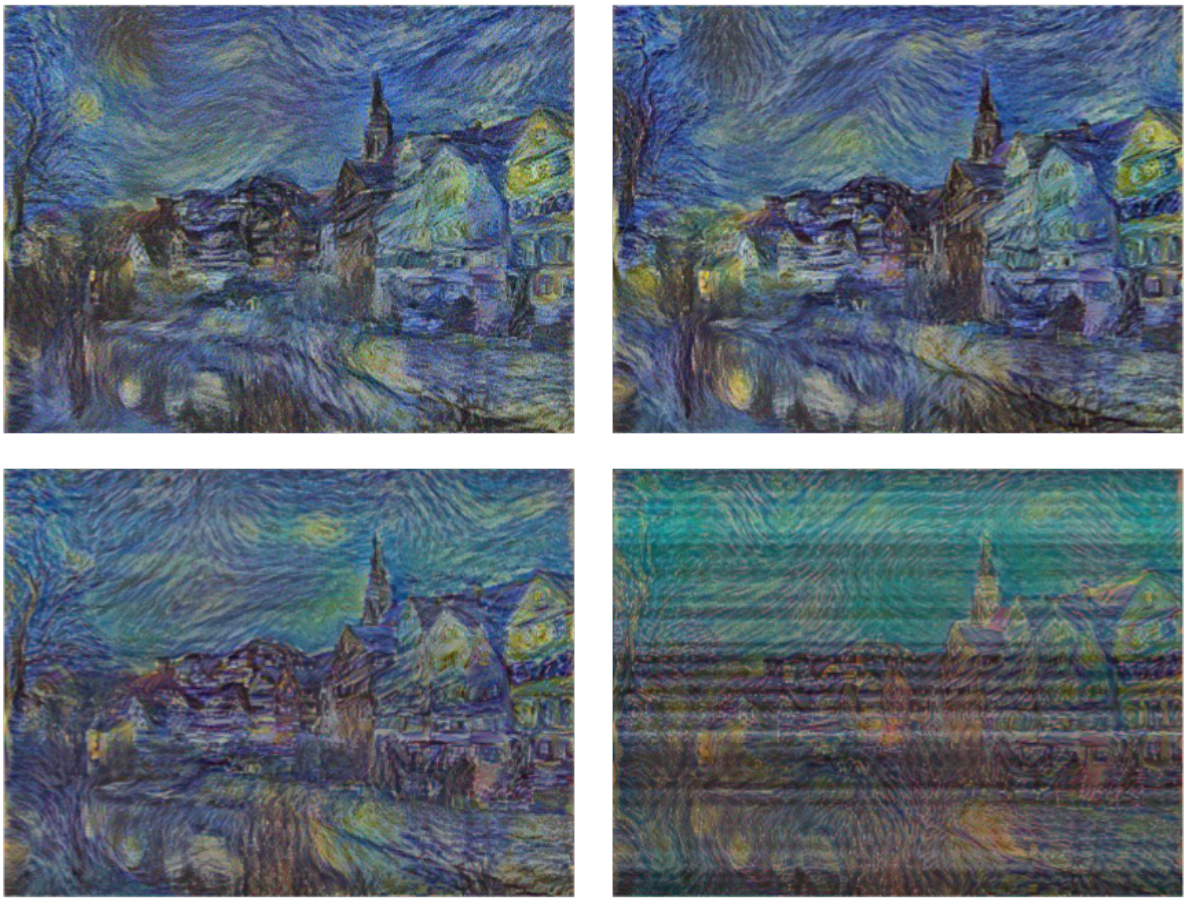

Figure 4: Results of continually increasing the variation weight term by one order of magnitude. The greater the variation weight, the smoother but more distorted the image is. A greater weight term also means that the image blurs together and takes on a hue of the image's average colour (in this case, blue + yellow = green). From left to right: $w_{\text{variation}} = 10^0$, $w_{\text{variation}} = 10^1$ (top row), $w_{\text{variation}} = 10^2$, $w_{\text{variation}} = 10^3$ (bottom row). Parameters used: style weight = 10^4, variation weight = 0, height = 400.

4.4 Gradient Descent

To visualize the process of gradient descent to the final image, the fourth experiment shows the results at intermediate iterations of the minimization using the L-BFGS-B algorithm.

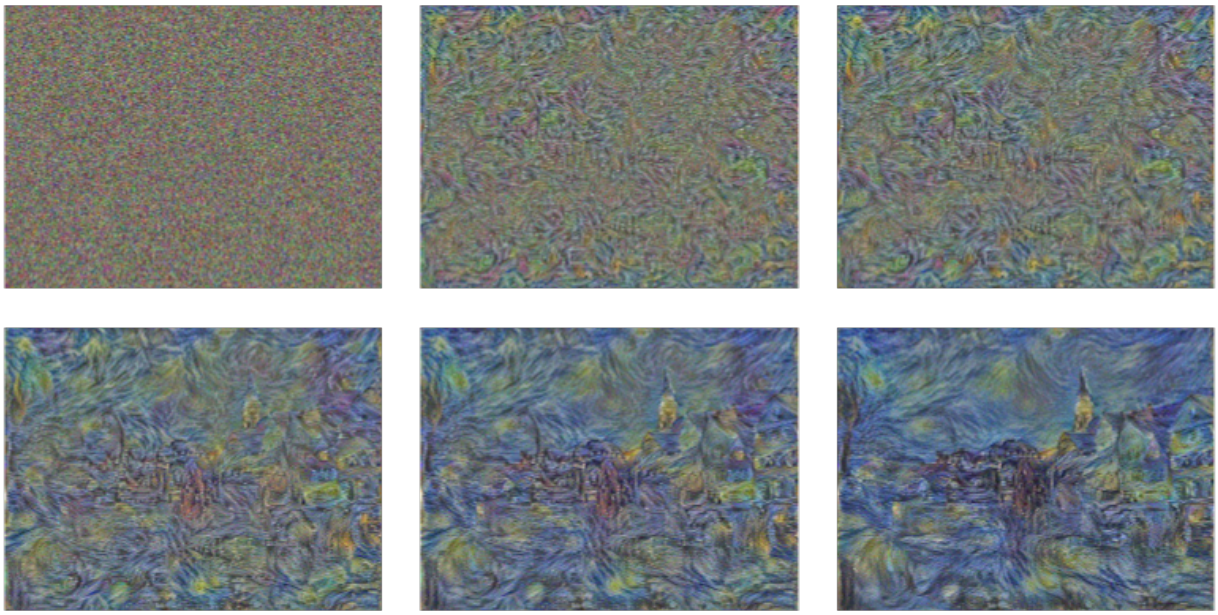

Figure 5: Evolution from the initial white noise image to the final combination image. The loss decreases at an exponentially decaying rate, and the minimization converges typically around a few hundred iterations. From left to right: 0, 20, 40 iterations (top row), 100, 200, 400 iterations (bottom row). Parameters used: $w_{\text{content}} = 1, w_{\text{style}} = 10^4, w_{\text{variation}} = 0$, height $= 400$.

4.5 Final Image

We can use the results of the experiments to determine that a good parameter set is $w_{\text{content}} = 1, w_{\text{style}} = 10^4, w_{\text{variation}} = 0$, iterations $= 1000$, height $= 400$. However, extended experimentation (which we do not go into depth here) showed that the larger the image, the greater the style weight term must be to compensate. Typically, with height $= 768$, the best style weight was $w_{\text{style}} = 10^5$. Thus, we can now show the final results of "The Starry Neckarfront":

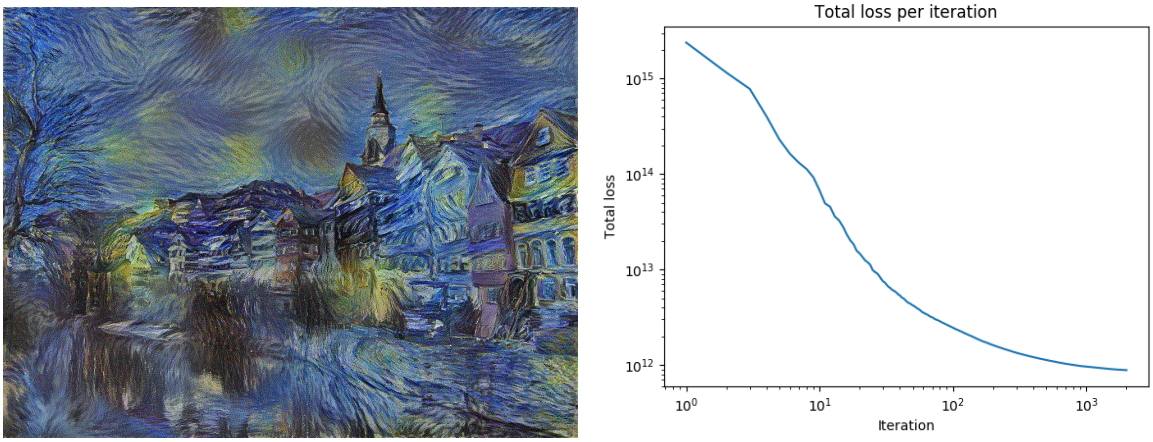

Figure 6: "The Starry Neckarfront" with the best parameters, and the total loss per iteration plotted on a log-log scale. It is noteworthy that the style weight term must be increased for larger combination images, and the images become smoother and higher quality with the increased size (as expected). Parameters used: $w_{\text{content}} = 1, w_{\text{style}} = 10^5, w_{\text{variation}} = 0$, iterations = 1000, height = 768.

5 Improvements

We make three additional improvements to the original stipulation:

- Better initialization
- Multi-style transfer
- Segmented style transfer

5.1 Better Initialization

The original paper uses a white noise image as the initial image to introduce a stochastic element to the process. However, the noisy initialization leads to remnants of the noise in the combination image, which is undesirable as the image is not smooth. We thus explore initialization using a blank image and the content image, using Caltech's Millikan Library as the content image and Hogwarts as the style image:

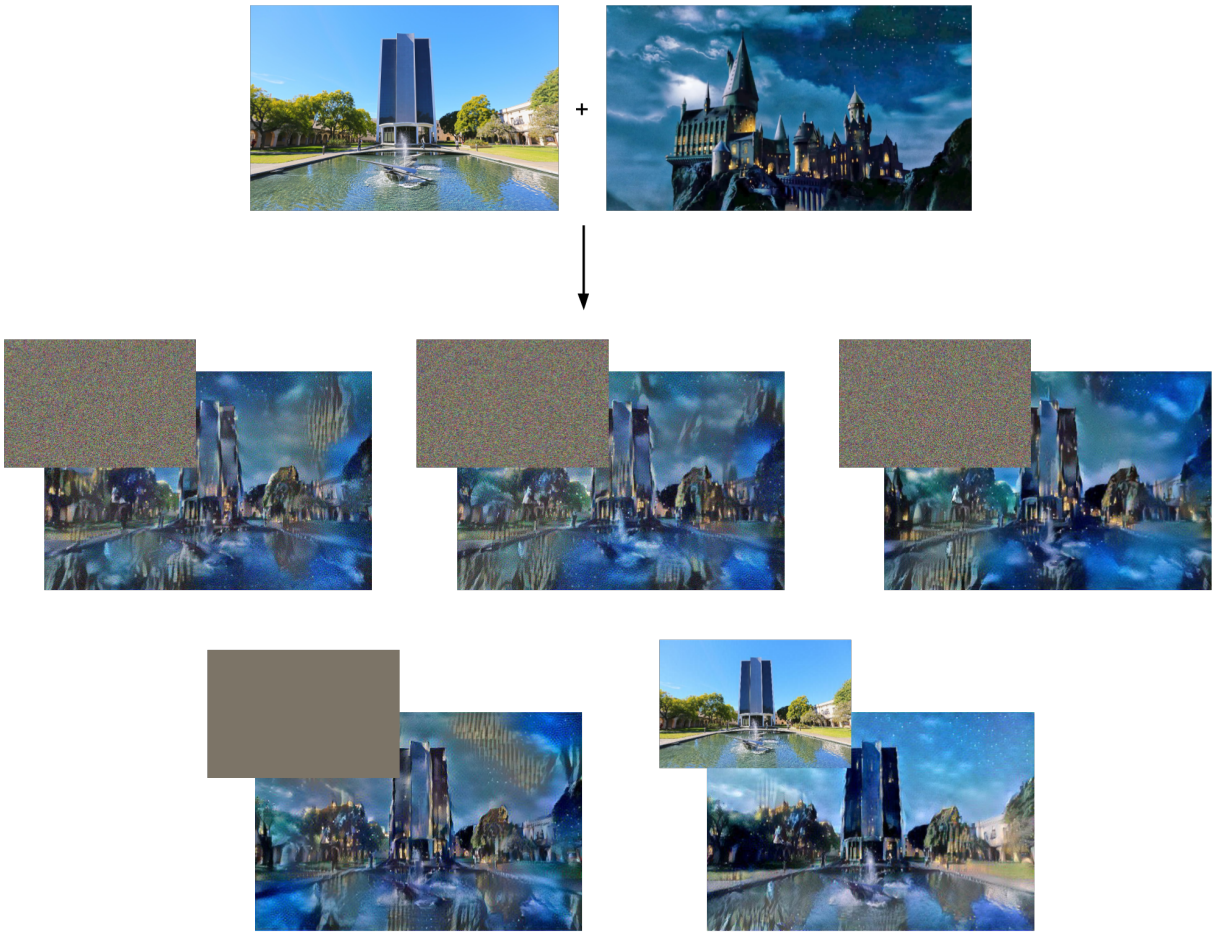

Figure 7: Results of using different initializations to get the combination image. The random initializations suffer from black borders and undesirable streaks across the background, while the other initializations are cleaner and do not have the same artifacts. From left to right: white noise initializations 1, 2, 3 (top row), blank image initialization and content image initialization (bottom row). Parameters used: $w_{\text{content}} = 1, w_{\text{style}} = 10^4, w_{\text{variation}} = 0$, iterations $= 1000$, height $= 400$.

Figure 8: Closeup of results from different initializations. The white noise initialization has a lot of background noise while the blank image initialization is smooth, as adjacent pixels started from the same value. The content image initialization is not only smooth but portrays the initial content image more accurately. From left to right: white noise initialization 1, blank image initialization, and content image initialization.

5.2 Multi-Style Transfer

We experiment with using more than one style image. To accomplish this, we expand the style loss term to be a sum of the style loss for each style image. We use the face of the primary author as the content image and Picasso's cubism works as the style images. We believe that this improvement is useful for capturing a more general style of a body of work.

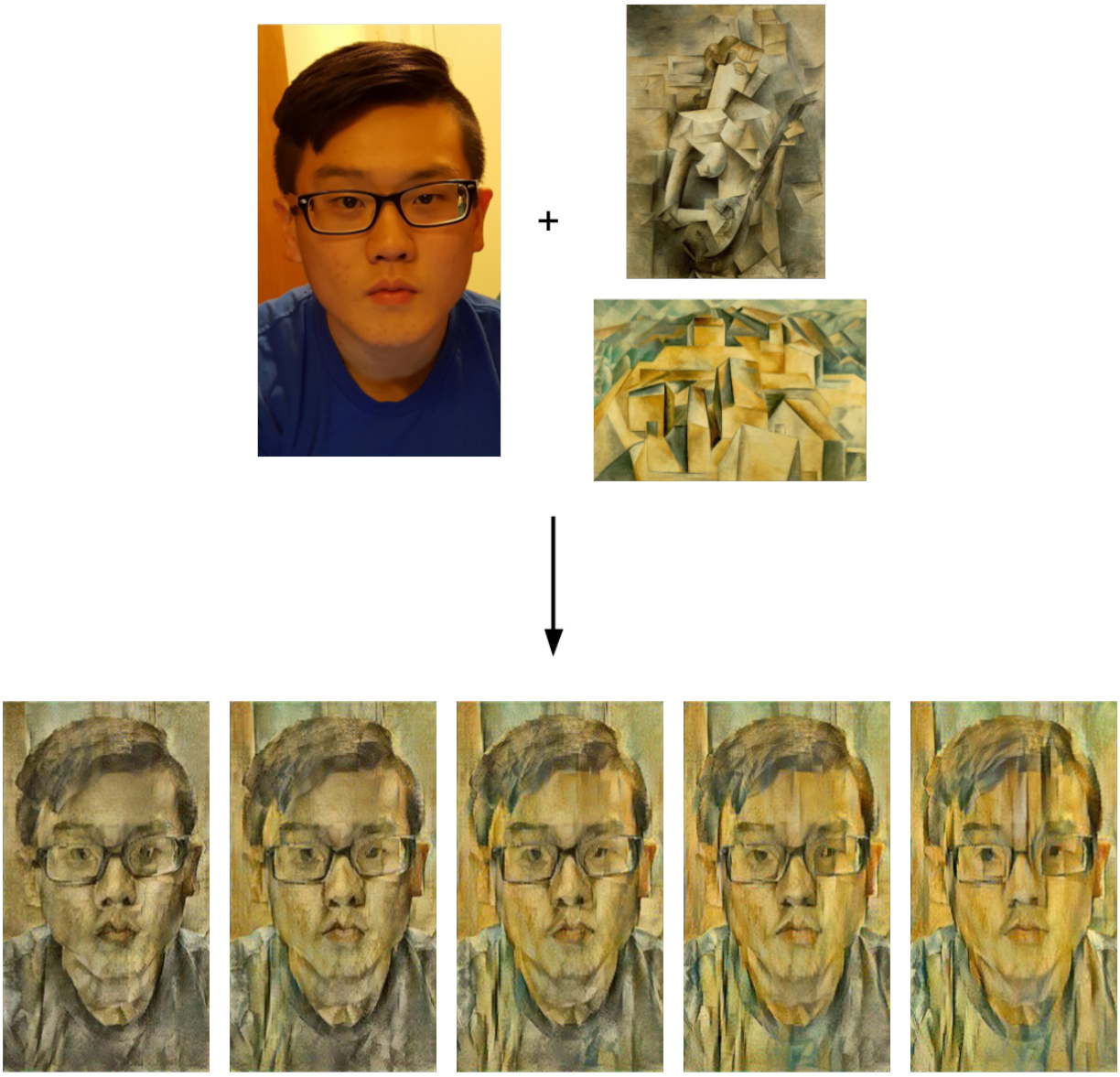

Figure 9: Results of putting different weights on the two style images. The image in the centre achieves a more general feel of cubism than the images at the ends by incorporating both style images rather than one or the other. The image in the centre is also more human-like. From left to right: $(1, 0), (0.25, 0.75), (0.5, 0.5), (0.75, 0.25), (0, 1)$, where the tuples are relative weightings on the top and bottom style images. Parameters used: $w_{\text{content}} = 1, w_{\text{style}} = 5 \cdot 10^3, w_{\text{variation}} = 0$, iterations $= 2000$, height $=$ 400, blank image initialization.

5.3 Segmented Style Transfer

Finally, we experiment with segmented style transfer: one problem with the previous style transfer examples was that the subject did not stand out enough from the background. To remedy this, we segmented the subject of importance from the background and performed two separate style transfers on the subject and the background, where the background style transfer had a smaller style weight term. The images were then joined back together. The segmentation was performed manually.

Figure 10: Results of segmented style transfer. Two style transfers were performed, one on the face alone and one on the entire image, before the two combination images were joined back together. Compared to the previous cubism examples, this result is visually more appealing as the subject does not blend into the background and instead stands out. The dark green and blue streaks are artifacts of the segmentation process, not the style transfer. Parameters used: $w_{\text{content}} = 1, w_{\text{style}} = 5{\cdot}10^5$ (subject), $1{\cdot}10^3$ (background), $w_{\text{variation}} = 0$, iterations = 2000, height = 400, blank image initialization.

6 Discussion and Conclusions

Style transfer works well for the right images, but the final results can feature noise and roughness, as well as a diminished resemblance to their reference images. A few ways to fix this, as shown in the experimentation and improvements, are:

- Running the style transfer for more iterations
- Increasing the style weight term for larger images
- Starting the style transfer from a blank image initialization or a content image initialization
- Using multiple style images
- Segmenting the subject and performing style transfer separately

Style transfer is a broad field, and it should be further extended from the results of this paper to address a few issues:

- Automation of parameter tweaking

- Increased smoothness and photorealism

Overall, style transfer works well with the right parameters, and with the proposed improvements, it can be used in not just art, but also photographs and sceneries.