CS 464
Machine Learning
2021 Fall

Final Report

Neural Style Transfer of Paintings to Portraits and Selfies

Group 7

Ömer Ünlüsoy (21702136)
Ata Berk Çakır (21703127)
Doğa Tansel (21802917)
Mustafa Hakan Kara (21703317)
Ahmet Cemal Alıcıoğlu (21801700)

Table of Contents

1. Introduction 3

2. Problem Description 4

3. Methods 4

4. Datasets 6

5. Experiments and Results 6

5.1. Alpha Ratio Experiment 7

5.2. Learning Rate Experiment 8

5.3. Epochs (Hypothetical) Experiment 9

5.4. Example Runs with Different Style Images 10

6. Discussion 11

7. Conclusion 12

8. Appendix 13

9. References 14

1. Introduction

Selfies have become a very common way of taking personal pictures in the recent decade. The commonality is most prominent in social media where people like to use many different aesthetic effects with the selfies they take. The same is true for portrait photography as well. Although portraits are used in much more formal settings like LinkedIn, they are also widely used. Selfies are more widely used among young people compared to adults and older people. In the end, both of these photo styles represent a very personal angle for the people in the picture, and so many people like to edit these photos using different techniques.

Style transfer is a method of applying the features of one input image (photo, painting, etc.) which is called Content Image onto another input image, Style Image, to produce an output image, Target Image, with the content of one and the style of the other [1]. Style Transfer is a well-known and common practice in Machine Learning and the most simple versions can be used to teach the fundamentals of it. Moreover, this production of a new image from two different images is considered as an art since it allows the user to adjust the blend ratio and the artistic users can produce beautiful results [2].

In our project, we used Convolutional Neural Networks for the style transfer implementation. We transferred styles of famous paintings onto photographs in the form of portraits and selfies, especially focusing on the selection of certain Artists such as Van Gogh, Francisco Goya, El Greco, etc.

We used the pre-trained VGG - 19 model from PyTorch to extract vital features from different layers of the images. We passed content and style images to this model to extract features from the different layers as previously mentioned. We smoothened the optimization procedure by calculating the gram matrices of the style features. Style losses were also part of our calculations, which were passed to the optimizer we used, Adam Optimizer, and reduced by it. The optimizer trained the final product on the target image, which means every pair of style-image and content started the training procedure from scratch.

After our progress presentation, we reworked some of our procedures, as style transfer creates a large runtime issue as the optimizer runs on the images themselves rather than on the model. We started using different style images (paintings) but we chose to not add a new target content image. The quality of the target image was enhanced by experimenting with the hyper-parameters. We also finished video creation of the style transitions. In the end, we obtained successful style transfers on the target image and had many different end products. We conducted three different experiments for several hyperparameters, alpha ratio, learning rate, and epoch number.

2. Problem Description

In this project, we have created a convolutional neural network that blends the style of an image with the contents of another image. This image stylization technique is called Neural Style Transfer (NST) [3], [4]. With this technique, users can easily create artistic images out of regular photographs. Our focus in this project is particularly on human portraits and selfies. The model transfers the style of paintings to these types of images. Several examples of such style transfers are shown in the figure below.

Figure 1: Style Transfer of Four Different Paintings to a Portrait

In addition, we allow users to select an artist and restyle their selfies according to the style of one of the paintings of that artist.

3. Methods

The main method we are using to implement style transfer is Convolutional Neural Network. We are using a VGG19 pre-trained model with its parameters frozen to extract the convolutional layers of style and content images. The layers we use to extract content are: {conv_1_1, conv_2_1, conv_3_1, conv_4_1, conv_5_1} and the layer we use to extract style is conv_4_2.

Using these layers, the loss function can be calculated using the following procedure:

The convolutional neural network algorithm for style transfer tries to optimize a total loss function that is defined as a weighted sum of the style loss and the content loss [4]. The ratio of these weights is a hyperparameter called alpha ratio and can be played with in order to improve the target image.

$$J = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$$

Where α and β are the content weight, and the style weight respectively, S is the style image, C is the content image and finally G is the output (resulting) image for the cost function. The ratio of the content loss and the style loss, called the alpha ratio, is a hyperparameter which should be tuned by us.

The most efficient algorithm to optimize the feature weights such that the target image is an acceptable blend of the style image into the content image is running gradient descent on the total loss function. We will use the Adam Optimizer in our implementation as it is complex enough and fast. Firstly, we are assigning G (resulting image) randomly and then with the help of the update rule given below, parameters are going to be tuned.

$$G^{(t+1)} = G^{(t)} + \nabla G(t)$$

Generally, the content loss can be determined from the specified content layer of the model. Current values of the content image and target image from this layer are extracted and their difference is squared.

Content loss can be formulized as:

$$J_{content}(C, G) = \frac{(a^{[l](C)} - a^{[l](G)})^2}{2}$$

Although the procedure is very similar for the style loss, two factors should be accounted for during the style loss calculation. First one is that generally there are more than one style layers and the second one is that gram matrices should be used for the style loss calculation. Calculation of the gram matrix for each style layer is pretty straightforward, multiplying the tensor with its transpose.

$$Gram_{kk'}^{[l](G)} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a_{i,j,k}^{[l](G)} a_{i,j,k'}^{[l](G)}$$

$$Gram_{kk'}^{[l](S)} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a_{i,j,k}^{[l](S)} a_{i,j,k'}^{[l](S)}$$

Using these Gram Matrices, the style loss function can be calculated by taking their difference squared for all pixels, then summing for each layer

Style Weight

$$L_{style} = \sum_{l} \sum_{i,j} (\beta G_{i,j}^{s,l} - \beta G_{i,j}^{p,l})^2$$

Style Loss

Gram Matrix for Style Image

Gram Matrix for Target Image

To optimize the target image such that it has minimal total loss, we run Adam Optimizer on the target image. The hyperparameters we have determined are:

- Epoch (number of iterations)
- Learning Rate
- Alpha Ratio
- Style Layer Weights

4. Datasets

We have used two datasets that consist of classical paintings with different styles: Van Gogh Paintings [5] and Best Artworks of All Time [6]. We are using the classical paintings in these two datasets as style images. While the Van Gogh Paintings dataset is organized based on the locations, the second dataset is organized according to the names of Artists. This organization allowed us to enable users to choose the Artists they want for the style image.

The significance of these datasets is to provide us different types of style images in order for us to better fine-tune our hyperparameters. These datasets are used to generate the target image, they are not used to train the model as the model we are using, VGG19, is pre-trained.

At the start of the program, users can select either Van Gogh or All Artists. If they have selected Van Gogh, they are presented with all locations in Van Gogh paintings and expected to choose a location for the style image. If the user chooses All Artists, the program expects the user to choose an artist from the listed artist names.

5. Experiments and Results

We conducted three controlled experiments in which we observed the change in the target image with respect to the increase/decrease of one of the parameters.

5.1. Alpha Ratio Experiment

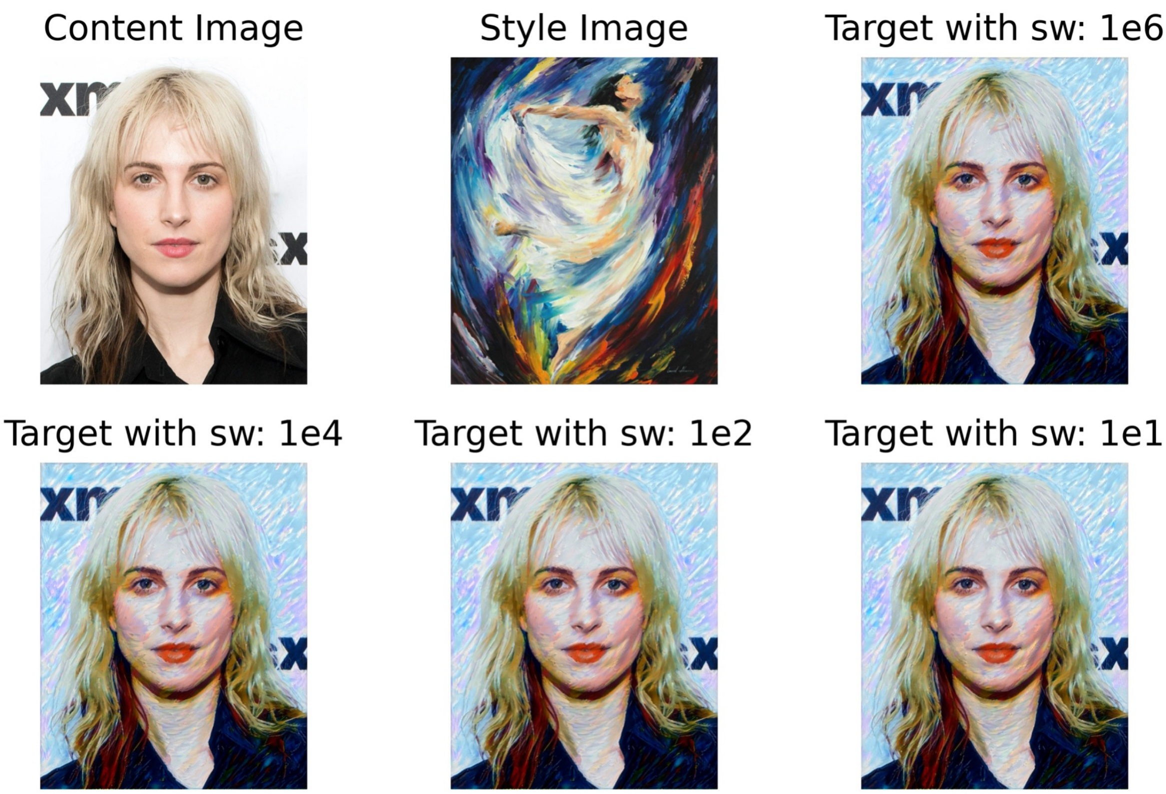

Figure 2: Alpha Ratio Experiment

The first experiment was regarding the weights of the style and content loss functions which is known as alpha ratio. As mentioned previously, the weighted sum of these functions gives the total loss function, which is the objective function that the Adam optimizer tries to minimize.

The alpha ratio is calculated by dividing the content weight with the style weight. In our experiment, we fixed the content weight to one and decreased the style weight from 1e6 to 1e1 while observing the target image. As the total loss is the weighted sum, it decreased as we decreased the style weight. However, loss is not a direct measure of the quality of the target image as the final result, the target image, should be considered as a piece of art rather than a function to minimize. It is observed again and again that minimizing the loss function does not necessarily increase the quality of the target image. That was actually the reason we did not do a (proper) experiment for the number of iterations (epoch), it totally depends on the image and the person looking at it.

If we come back to the alpha ratio experiment, it is observed, as it should be, that the content of the target image is spoiled when the style weight is increased. However, the change is much smaller than we expected and cannot be seen by comparing the target images one by one.

That is the reason we created an image collection consisting of 6 images and the initial images and created the videos of each transformation.

5.2. Learning Rate Experiment

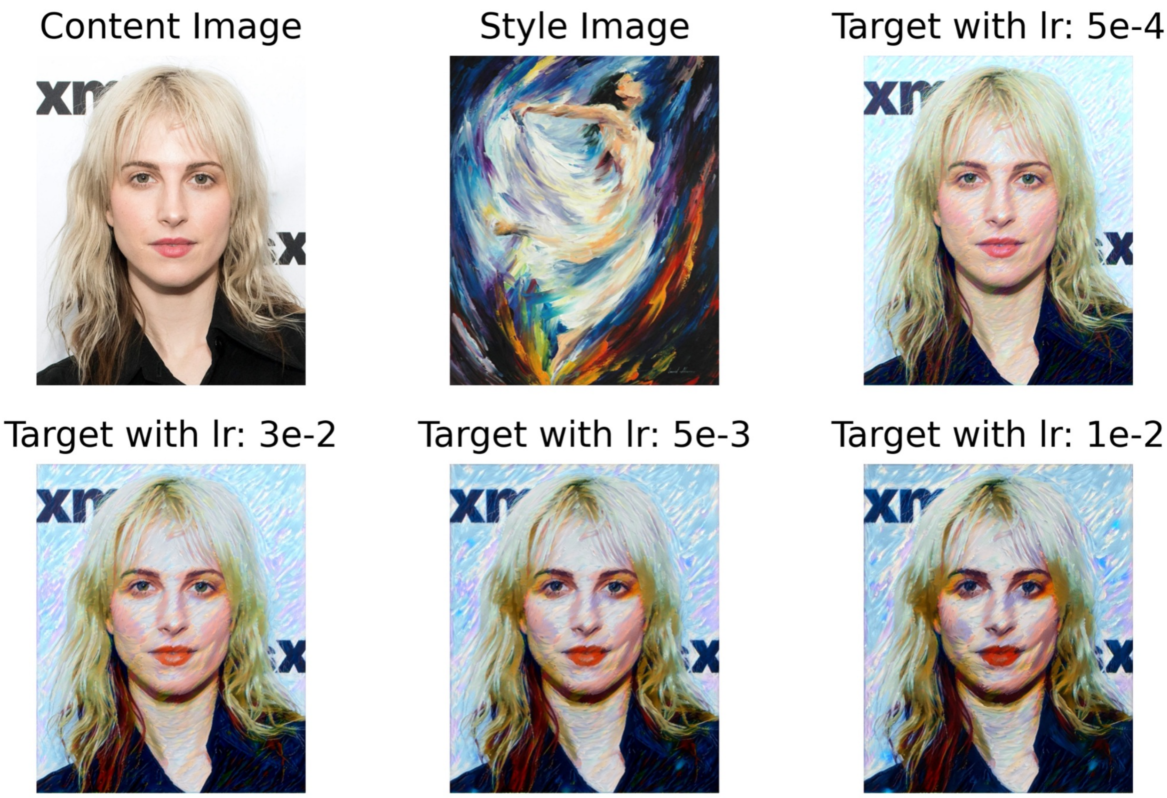

Figure 3: Learning Rate Experiment

In the second experiment, we chose the learning rate hyperparameter as the independent variable. The change in the learning rate directly affected the duration of the total loss function's convergence. As we decreased the learning rate, not only did it take longer for the algorithm to terminate for one iteration, but also, we needed to increase the number of iterations (epochs). On the other hand, when we increased the learning rate, we were able to obtain quite a faster convergence. However, this convergence provided a suboptimal solution and the style transfer was not as successful. As mentioned earlier, decreasing the total loss is not the measure of the quality of the target image.

While the largest learning rate spoils the target image, the smallest one has not enough time to converge. Thus, it is suggested that the one of the middle ones such as 2e-3 or 5e-3 should be considered. Note that the 2e-3 is printed as 3e-2 on the comparison image wrongly. If the user wants to increase the style of the target image, 5e-3 can be chosen while 2e-3 gives more mild tones of the style.

5.3. Epochs (Hypothetical) Experiment

The third experiment was related to the number of epochs. It should be noted that the proper epoch experiment is impossible since its optimal value changes from one content image to another. However, we found a simple trick to observe the effect. In order to easily observe its results on the target image, we have saved images from the training process (1 image per 7 epochs) and converted all the image array into a video. By doing that, we manage to show the process of style transfer to the user and allow users to observe the change depending on the epoch number.

Since we have started our target image from the content image to decrease the training time, it is observed that when the number of epochs increased, the texture of the style image appeared more in the target image, the target image tried to converge to the mixture of the style and content from the pure content. That is, it produced a similar result to increasing the weight of style loss.

All the results of the experiments with the proper log files can be found in the script submission.

5.4. Example Runs with Different Style Images

Figure 4: Example Run I

Figure 5: Example Run II

Figure 6: Example Run III

Figure 7: Example Run IV

6. Discussion

Although the experiments are discussed in the Experiments Section, we can rephrase some of the discussion here.

For the Alpha Ratio Experiment, the main observation was that minimizing the total loss does not necessarily produce a target image with higher quality. Although the sense of quality is objective and the final result of the style transfer should be considered as art rather than a function, general sense suggests that the target image is ruined when the style weight is increased drastically as the style convergence occurs too fast. However, the change is smaller than the expectations.

For the Learning Rate Experiment, it is observed that neither the highest nor the smallest learning rate is suitable for the style transfer. The middle has to be chosen. The main reason for this is that while the high learning rate converges to the style image too fast, low learning rate is too late to converge. This effect can be minimized with increased epoch but this means extra run time.

For the Hypothetical Epochs Experiment, the convergence of the target image can be seen very clearly. It is concluded that the required number of epochs depends on the consistency of the initial images and the person watching it.

7. Conclusion

In conclusion, style transfer is a Machine Learning application that separates itself from the other applications by running the optimizer on the target image rather than the model itself. Thus, the training procedure happens on the target image with the Adam Optimizer which minimizes the total loss function, the weighted sum of style and content loss. The optimizer changes each pixel of the target image according to the Content and Style Images. That is why, the Style Transfer is the closest application of ML to Image Processing. As the final result is an image rather than a class or a regression value, we had no accuracy to calculate although we had a loss value. There isn’t a basis or criteria for judging if the project was successful or not except the human eye.

Since the runtime of the style transfer process, we have demonstrated is too slow, it is infeasible for making a selfie application with this process. However, it can be said that the images created as an end product were successful and showed the transfer of the styles particularly well. Videos we made of the transitions also depicted this very well. All in all, we were able to provide a set of paintings to select and transfer style onto content images. Finally, we have done some experiments on the hyperparameters of the Style Transfer such as alpha ratio, learning rate, and epoch number; and discussed their results.

8. Appendix

Teammate Contributions to the project:

- **Ömer Ünlüsoy**: CNN and PyTorch Learning, Transfer Learning with VGG19, Image Retrieval, Training Procedure, Video Creation, What Is Done and What Remains to Be Done sections of Progress Report, Dataset Implementation, Experiments and Discussion Sections of Final Report (with Hakan).

- **Doğa Tansel:** PyTorch Learning, Introduction, Background Research, What Remains to be Done Sections of Progress Report, Introduction, Problem Description, Conclusion and Results Sections of Final Report.

- **Ahmet Cemal Alıcıoğlu:** CNN and Pytorch Learning, Background Research, Methods, Datasets.

- **Ata Berk Çakır**: PyTorch Learning, Introduction, Background Research, What Remains to be Done Sections, Mathematical formulations.

- **Mustafa Hakan Kara**: Presentation preparations, background research, the theoretical computation of loss functions and gram matrices.

9. References

[1] Y. Shih et al. “Style Transfer for Headshot Portrait.” Available: https://people.csail.mit.edu/yichangshih/portrait_web/2014_portrait_hires.pdf

[2] M. Lu, F. Xu, H. Zhao, A. Yao, Y. Chen and L. Zhang, "Exemplar-Based Portrait Style Transfer," in *IEEE Access*, vol. 6, pp. 58532-58542, 2018, doi: 10.1109/ACCESS.2018.2874203.

[3] L. A. Gatys, A. S. Ecker and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2414-2423, doi: 10.1109/CVPR.2016.265.

[4] L. A. Gatys, A. S. Ecker and M. Bethge, “A Neural Algorithm of Artistic Style,” arXiv:1508.06576

[5] “Best Artworks of All Time,” https://www.kaggle.com/ikarus777/best-artworks-of-all-time

[6] “Van Gogh Paintings,” https://www.kaggle.com/ipythonx/van-gogh-paintings