Abstract.

Neural style transfer is a fascinating area of study where one can reproduce a given image with a new artistic style. In this project we extend the original algorithm proposed by Gatys et. al to take two style images by modifying the loss function to include another style image with a new parameter that defines the influences of the two style images. Experimentations were carried out to explore the effects of different hyperparameter values. The report is concluded with a brief discussion on the effect of AI in the visual arts.

1. Introduction

Pastiche is an artistic work that emulates the style of another artwork [3]. Replicating this task was challenge for AI systems as it was difficult to extract features such as texture information. The enormous developments in deep learning for vision has propelled the surge in the use of computer vision to automate pastiche and this is known as style transfer [5]. Gatys et. al proposed Neural style transfer (NST), an optimization technique that takes two images (a content image and a style image) and generates a blended image. The generated image preserves the contours of the content image and adds texture and colour patterns of the style image. Inspired by the original algorithm, this project focuses on mixed/multiple neural style transfer (MNST), which uses two style images and one content image. The following sections will provide more details about the method and system, experimentation setup, and resulting observations.

2. Background

Art is an important domain that holds meaningful values in our society as it is a powerful form of expressionism and a unique manifestation of human creativity [6]. Thus, researchers have been striving to replicate visual artefacts using computational algorithms for many years. Building on earlier research into texture image generation, Gatys et. al introduced NST, a deep neural network approach that has been integral in propelling the revolution of artistic stylisation using convolutional neural networks. Although the original NST can yield quality results, it is computationally expensive to implement as every pair of content and style image requires re-training. Since the emergence of NST, researcher have been refining and optimising NST algorithms with the aim to perfectly replicate selections of styles and extend its application to audio styles, for instance [5]. Johnson et. al improved the performance of style transfers by training a feedforward network with perceptual loss functions [2]. Additionally, Dumoulin et. al trained a single conditional style transfer network to capture up to 32 various styles and observed that much information in the models for different styles is shared and does require re-training [3]. Liu et. al also proposed using depth loss [5]. This is effective at retaining spatial layout and structure content images, compared with single gram loss, thereby preserving depth maps of content images.

3. System Description

This section describes the NST model [1] and provides further information regarding the modification of NST to form MNST. Gatys et. al proposed using the pretrained VGG-19 model to calculate the content and style losses. VGG-19 is a convolutional neural network that is 19 layers deep with 16 convolutional and 5 pooling layers. A visualisation of the VGG-19 network is displayed in Figure 3.1. The CNN consists of layers of small computational units that process visual information hierarchically in a feed forward manner. Each layer of units can be interpretated a collection of image filters that extracts certain features from the input image. Hence, the output of a given layer consists of features maps, which are different filtered versions of the input image.

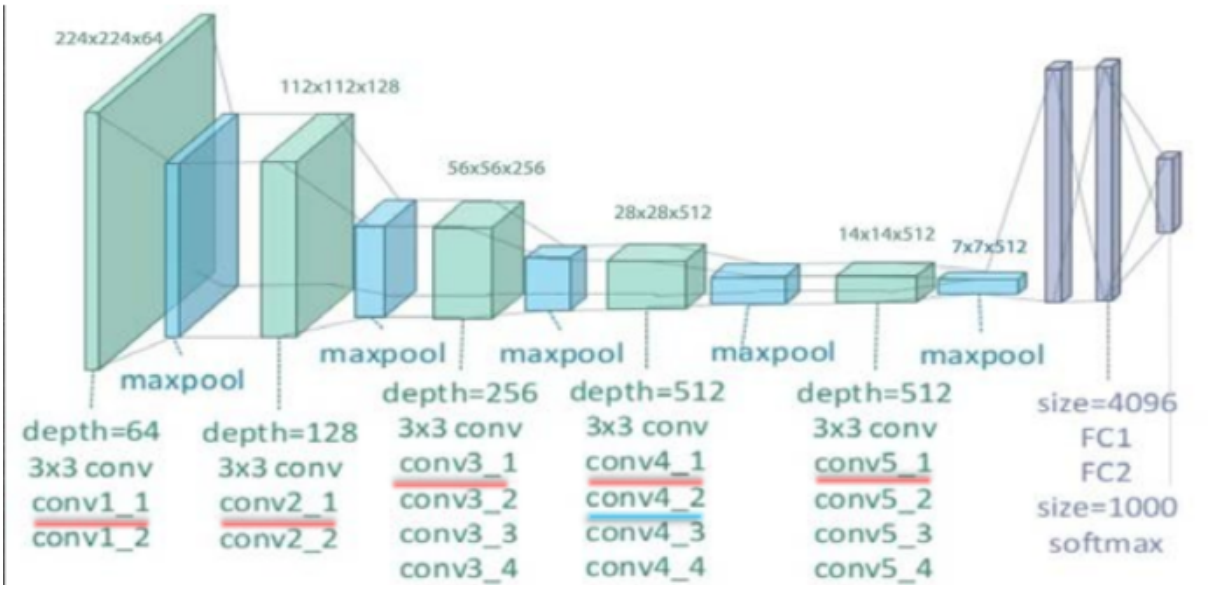

Figure 3.1: VGG-19 network visualisation. [4]

The convolutional layers are named by stack and their order in the stack. For instance, conv1_1 is the first convolutional layer that an image passes through in the first stack, and conv5_4 is the deepest convolutional layer in the network. Using VGG-19, two sets of features, the content, and the style features, can be extracted.

3.1 Extraction of the content and style features

As the network gets deeper, the features become more complex, with the best feature representation of the image near the final CNN layers. In other words, higher layers can locate and recognise high-level content in terms objects, shapes, and arrangements in the image Thus, when extracting the content from an image, the higher layers of the network are utilised. In [1], the content features were extracted from the content images using the conv4_2 layer of the model (underlined in blue in Figure 3.1).

The lower layers of the network learn to recognise lines, edges, textures, and colours, which represent the low-level details of the image. Correlations between the different filter responses over the spatial extent of the feature maps are employed to obtain a representation of the style of an input image. Hence, the style representation is a multi-scale representation that contains multiple layers of the network to capture texture information but not the global arrangement.

Hence, we can use lower layers such as conv1_1, conv2_1, conv3_1, conv4_1 and conv5_1 (see Figure 3.1) to extract style features.

3.2 Content loss, Style loss and Gram matrix

Extracting the content and style features enables us to build the NST algorithm. To do so, we optimise the image to represent the content and style of the input images. Optimisation requires the minimisation of a cost function, which is the total cost function that consists of the content loss $L_{content}$ (similarity of content image to generated image) and style loss L_{style}(similarity of style image to generated image).

The content loss is given by:

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

, where p and x are the original image and the generated image and P_{ij}^{l} and F_{ij}^{l} are their respective feature representation of the i^{th} filter at position j in layer l.

Given the original image a, and the generated image x, A_{ij}^{l} and G_{ij}^{l} are their respective style representations in layer l. The contribution of layer l to the total style loss is then:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} \left(G_{ij}^{l} - A_{ij}^{l}\right)^2$$

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l$$

w_l are weighting factors of the contribution of each layer to the total style loss. G is the gram matrix which represents a correlation between style features. G_{ij}^{l} is the inner product between the vectorised feature map i and j in layer l.

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l}$$

Thus, given a photograph p and an artwork a, the total cost function for NST is given by:

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

, where α and β is called are the content and style weights.

3.3 Mixed Neural style Transfer (MNST)

Now we will focus on the variation of the algorithm that incorporates two style images.

As shown in section 3.2, the total loss function for NST covers two aspects, content loss and style loss. For MNST, we introduce another parameter γ as a second style weight [8]. This parameter defines the degree to which a style influences the overall style loss. For instance, if γ=0.8, then the loss of the first style image has a bigger impact on the overall style loss in comparison to the loss of the second style image. Thus, the total loss function for MNST can be represented as the following:

$$L_{MNST}(c, s_1, s_2, x) = \alpha L_{content}(c, x) + \beta(\gamma L_{Style}(s_1, x) + (1-\gamma) L_{Style}(s_2, x))$$

Figure 3.2: Total loss function to be minimised for MNST. c denotes content image, and s_1 and s_2 denotes the style images for MNST. [8]

, where α and β are the weights for content loss and style loss, respectively. The variable c denotes content image, and s_1 and s_2 denotes the style images for MNST. The parameter γ as explained above, defined the degree to which style image 1 affects the overall style loss and γ∈[0,1].

4. Experiments and Results

This section describes the experimentations conducted and results obtained. As previously mentioned above, the experiments were conducted using two style images and one content image.

Using the TensorFlow tutorial on NST, the following setup was used to implement the MNST and experimentations:

- Model: pretrained VGG19 model with ImageNet weights
- Content output layer: conv5_2
- Style output layers: conv1_1, conv2_1, conv3_1, conv4_1, conv5_1
- Different values of α and β, with α≤ β, as we are mostly interest in the effect of the style images on the content image.
- Optimiser: Adam with learning rate of 0.025, beta = 0.99, and epsilon = 1e-1
- Training conducted with 200 epochs and 20 steps per epoch

The first two experiments were focused on the effect of different values of γ on the generated MSNT images. Thus α and β were fixed to values of 0.001 and 0.01, respectively.

Style image 1

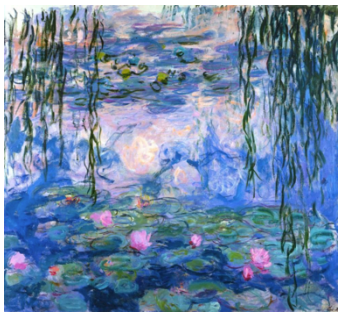

Figure 4. 1: Style image 1, water lilies painting by Claude Mont

Style image 2

Figure 4.2: Style image 2, Composition VII by Wassily Kandinsky

Content image

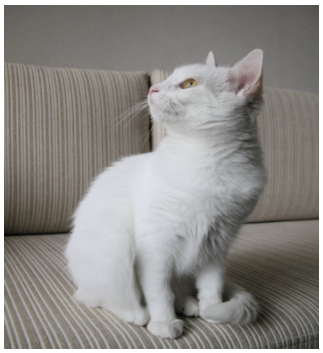

Figure 4.3: Content image, link to image in colab notebook

Using the styles images in Figures 4.1- 4.2 and content image in figure 4.3, we obtain the following results for the first experiments:

$\gamma = 0$

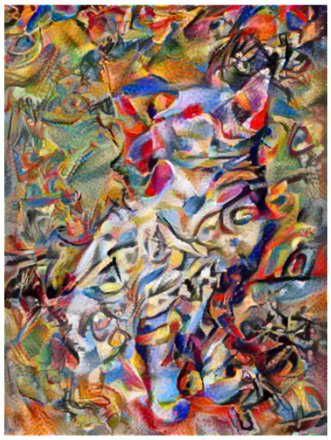

Figure 4.4: MNST result when $\gamma = 0$. Total loss after 200 epochs is 1.07e+05

$\gamma = 0.2$

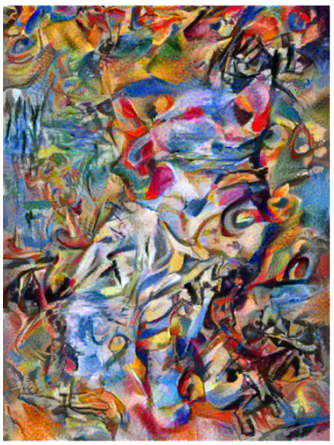

Figure 4.5: MNST result when $\gamma = 0.2$. Total loss after 200 epochs is 1.06e+08

$\gamma = 0.5$

Figure 4.6: MNST result when $\gamma = 0.5$. Total loss after 200 epochs is 1.65e+08

$\gamma = 0.7$

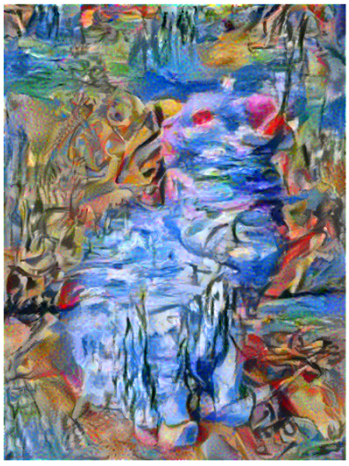

Figure 4.7: MNST result when $\gamma = 0.7$. Total loss after 200 epochs is 1.39e+08

$\gamma = 1.0$

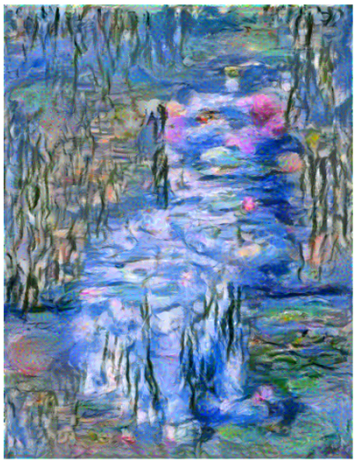

Figure 4.8: MNST result when $\gamma = 1$. Total loss after 200 epochs is 1.61e+04

The second experiments were conducted using the following style images and content images:

Style image 1

Figure 4.9: style image 1, link to image in colab notebook

Style image 2

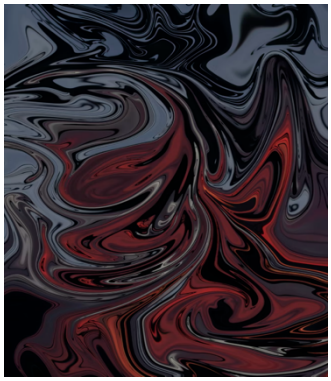

Figure 4.10: style image 2, link to image in colab notebook

Content image

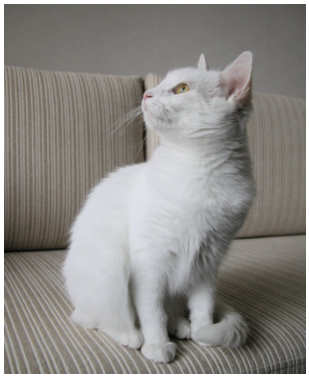

Figure 4.11: content image, link to image in colab notebook

MNST results from using images from Figures 4.9 – 4.11 are as follows:

$\gamma = 0.0$

Figure 4.12: MNST result when $\gamma = 0.0$. Total loss after 200 epochs is 5.32e+04

$\gamma = 0.5$

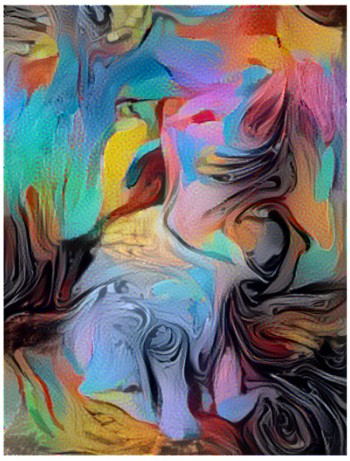

Figure 4.13: MNST result when $\gamma = 0.5$. Total loss after 200 epochs is 9.25e+07

$\gamma = 1.0$

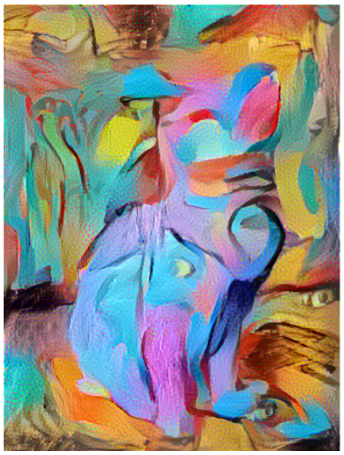

Figure 4.14: MNST result when $\gamma = 1$. Total loss after 200 epochs is 7.97e+04

Style image 1

Figure 4.15: style image 1, link to image in colab notebook

Style image 2

Figure 4.16: style image 2, link to image in colab notebook

Content image

Figure 4.17: content image, link to image in colab notebook

MNST results from using images from Figures 4.15 – 4.17 are as follows:

$\gamma = 0.0$

Figure 4.18: MNST result when $\gamma = 0.0$. Total loss after 200 epochs is 5.55e+04

$\gamma = 0.5$

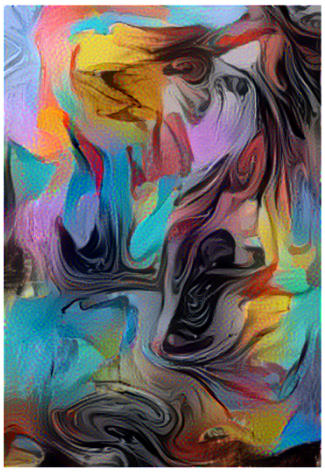

Figure 4.19: MNST result when $\gamma = 0.5$. Total loss after 200 epochs is 9.25e+07

$\gamma = 1.0$

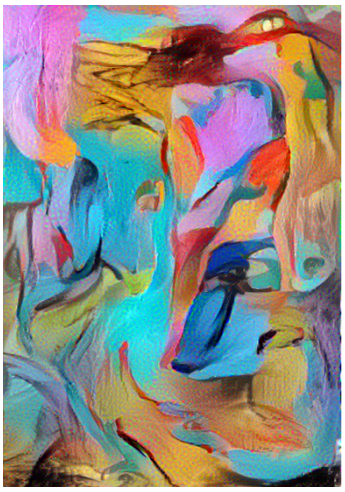

Figure 4.20: MNST result when $\gamma = 1$. Total loss after 200 epochs is 9.16e+04

Example of the results obtained from experimenting with different content and style weights ($\alpha = 0.001$ and $\beta = 100$) with $\gamma = 0.5$.

Figure 4.21: MNST result when $\gamma = 0.0$. Total loss after 200 epochs is 1.65e+12

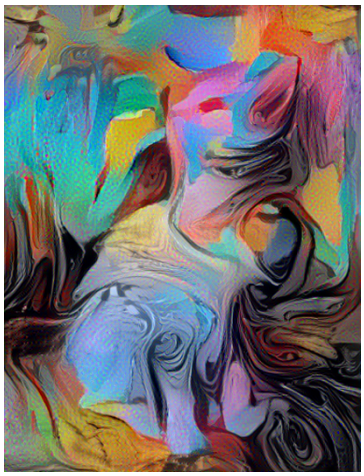

Figure 4.22: MNST result when $\gamma = 0.0$. Total loss after 200 epochs is 9.25e+11

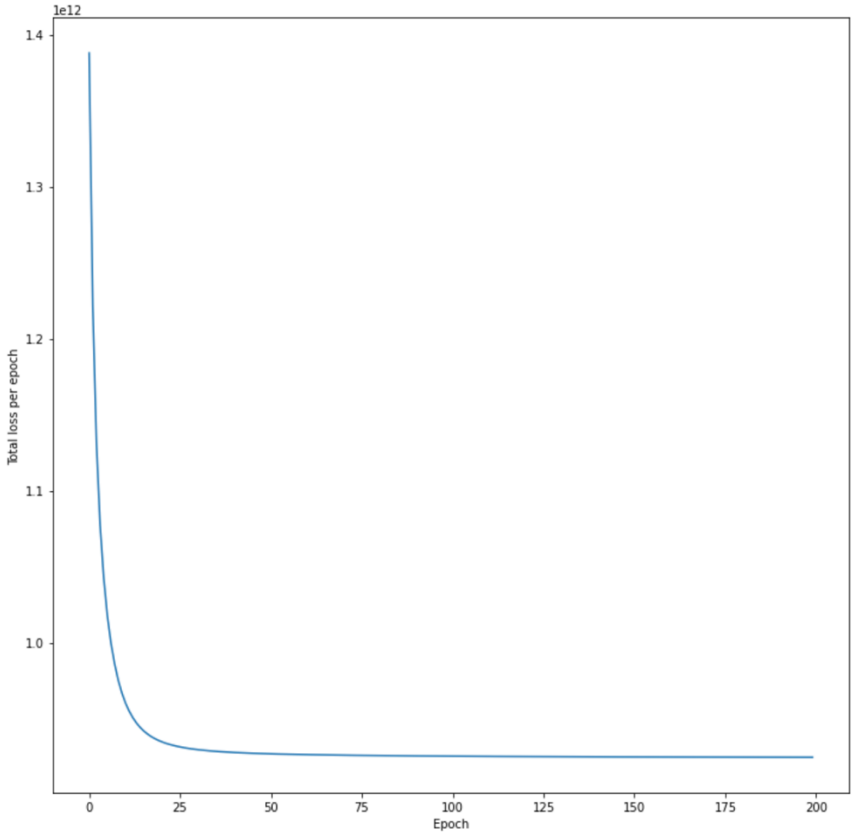

Figure 4.23: Plot of Figure 4.22 loss for 200 epochs.

5. Discussion

When investigating the effects of different values of γ , the impact is different values of γ is evident in the overall MNST result. Figures 4.4 – 4.8 shows the effect of increasing γ from 0 to 1. When γ = 0, only style 2 was used to generate the image. Alternatively, when γ = 1, only style 1 was used as the style image. Both scenarios are effectively examples of NST results. Furthermore, when γ = 0.2, we can see traces of style 1, but style 2 is predominantly observable. Likewise, when γ=0.7, style 1 is more noticeable. When γ = 0.5, the colours and textures of both styles are employed, with style 1 being mostly present on the cat (object), whereas style 2 is mainly captured in the background of the content image. Additionally, γ = 0.5 resulted in the highest total loss (1.65e+8) after 200 epochs. This is an indication that MNST is taxing for the model to generate an image where the two styles influence the final result equally.

The use of contrasting styles images with different structures and colouring yielded interesting results. When γ = 0.5, the MNST does not capture the content very well, especially in figure 4.13 where the shapes of the main objects (e.g., the skater) and the global arrangements are deformed. When comparing the output from Figure 4.13 (uses styles from figures 4.9-4.10) to figure 4.6 (uses styles from Figures 4.1- 4.2), the overall structure of the MNST in Figure 4.6 is less distorted. Thus, the artistic style of the style images also plays important role in the generated artefact.

Lastly, I found that changing the α/β ratio from $1x10^{-1}$ to, $1x10^{-5}$ for instance, did not result in substantial changes in the outputs. [1] mentioned that a strong emphasises on style leads to texturized version of the content image, but the results obtained Figures 4.21-4.22 did not

differ significantly from previous results when $\gamma = 0.5$. This could be attributed to the use of conv5_2 which captures high-level content than conv4_2. In addition, Figure 4.23 shows that the plot of the total losses converges early, which could indicate that the learning rate is too high leading to suboptimal solutions when α/β ratio is high.

Although neural style transfer can yield fascinating results, many artists and art historians oppose the use of AI to create art, as they view art as an expression of an artist's emotion, style, and expressive point of view, and that is something that an AI system cannot fulfil [6]**.** Art created using AI can be viewed as a form of inspiration and a new medium to supplement the creation of art. Therefore, we could see a rise in an effective partnership between artists and creative AI systems in future.

6. Conclusions and Further Work

The comparative study carried out using MNST under different conditions yielded a series of interesting outputs. We were able to show the effect of using different values of γ. Secondly, using contrasting style images resulted in a decrease in the global arrangements in comparison to when 2 colourful style images were utilised. Finally, the use of varying α/β ratios did not yield the expected results and we speculated that the reasons could be attributed to the learning rate and the content layer conv5_2.

It is very difficult to evaluate aesthetic artefacts such as art [5]. Thus, further work on creating an evaluation metric for NST (and MNST) is critical as the filed matures. However, evaluation metrics proposed by Boden [7] can be a good steppingstone to critic one's effort. From my experience and point of view, this project falls closely under the psychological creativity category as the idea of MNST is new to me but not a new innovative in this field. One additional action that could be investigated further is the issue regarding the α/β ratio through the experimentations with different learning rates and SGD optimiser. Work on emphasising different parts of the content image with different styles can also be further explored.

References

1. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576* (2015).
2. Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." *European conference on computer vision*. Springer, Cham, 2016.
3. Dumoulin, Vincent, Jonathon Shlens, and Manjunath Kudlur. "A learned representation for artistic style." *arXiv preprint arXiv:1610.07629* (2017).
4. Ayush Chaurasia, 2020. "Exploring Neural Style Transfer". [online] Medium. Available at: < https://medium.com/analytics-vidhya/explore-neural-style-transfer-with-weights-biases-344533d7b080 >
5. Jing, Yongcheng, et al. "Neural style transfer: A review." *IEEE transactions on visualization and computer graphics* 26.11 (2019): 3365-3385.

6. Mazzone, Marian, and Ahmed Elgammal. "Art, creativity, and the potential of artificial intelligence." *Arts*. Vol. 8. No. 1. Multidisciplinary Digital Publishing Institute, 2019.
7. Boden, Margaret A. "Creativity and artificial intelligence." *Artificial intelligence* 103.1-2 (1998): 347-356.
8. Paul Froehling, 2021." Mixed Neural Style Transfer with Two Style Images". [online] Towards Data Science. Available at: < https://towardsdatascience.com/mixed-neural-style-transfer-with-two-style-images-9469b2681b54 >