Style Transfer

Student: Sotir Anca-Nicoleta
Mentor: Andrei Nicolicioiu

Universitatea din București
Virtute et Sapientia

Bitdefender

Introduction

Style Transfer is a process of modifying the style of an image while still preserving its content.

- The **content** of an image: the subjects captured in it, including their form and layout.
- The **style** of an image: its color pallete and texture.

The idea that we can separate the concepts of content and style in an image is the basis of Neural Style Transfer.
The goal of this project is to implement Neural Style Transfer exploring two approaches introduced by Gatys et al.[1] and by Dumoulin et al.[2].

Concepts

We first define the similarity between two images:

- Two images are **similar in content** if their features are close in Euclidian distance.
- Two images are **similar in style** if their features share the same statistics.

The difference in content of two images C and G is given by the following **Content Loss** function:

$$\mathcal{L}^l_{content}(C, G) = \frac{1}{2} \sum_{i,j} (C^l_{ij} - G^l_{ij})^2 \quad (1)$$

where C^l and G^l are the features of the respective images extracted by a trained classifier at layer l.

We use the concept of **Gram matrix**, which helps compute the correlations between two images' features extracted by a trained classifier.

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \quad (2)$$

where F_i and F_j are the linearised feature maps i and j in layer l.

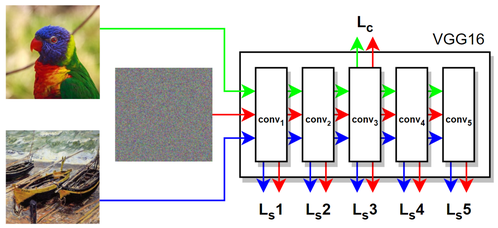
The difference in style of two images S and G is given by the following **Style Loss** function:

$$\mathcal{L}_{style}(S, G) = \sum_{l=0}^{L} w_l \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (S^l_{ij} - G^l_{ij})^2 \quad (3)$$

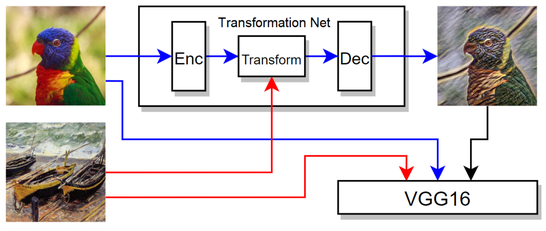
where N_l and M_l are the number and size of the feature maps, S^l and G^l are the gram matrices of the respective images at layer l. Also, w_l is the weight factor for each layer taken into consideration.

Methods and Architectures

Gatys et al.:
The idea is to **generate an image (random at first) that will be optimized** so that both content and style losses are minimized.
We will use a **VGG16 model (pretrained to classify images)** to extract the features and calculate the loss. This model is fixed and we optimize only the generated image.



Dumoulin et al.:
The idea is that we will **use an image transform network instead of beginning with a random image**. Also, to make this network have the ability of **representing multiple styles at once**, we add two parameters γ and β that are specific to each style.



The VGG16 model is like the one used in the Gatys' method. This time we will optimize only the Transformation Net, including the γ and β parameters.
The "Transform" step consists of normalizing the output of the Encoder (x) and then **applying a transformation with γ_s and β_s corresponding to the style image we want to use**.

$$z = \gamma_s \frac{x - \mu_x}{\sigma_x} + \beta_s \quad (4)$$

All styles for which the Transformation Net is trained share all parameters of the Encoder and Decoder and differ only through γ and β.

Dataset

Gatys et al.: for this method, the model can run only with one style image and one content image.
Dumoulin et al.: the model was trained on 5000 content images from the COCO dataset.

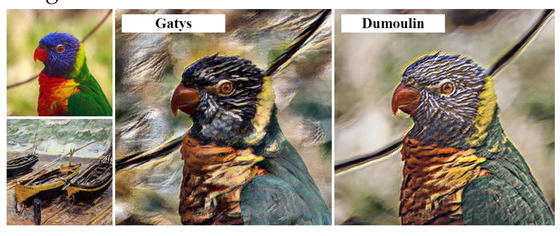
Results

Figure 1: Results for the Gatys et al. method

Figure 2: Results for the Dumoulin et al. method

Experiments

1. **Simulating the Gatys method with the Dumoulin method**: The Transformation Net is given only one style image, thus we have only one set of γ and β parameters. We do not train the model on the COCO dataset, instead giving it the same content image at each iteration.



We can observe checkerboard type patterns in the image labeled "Dumoulin": using an Encoder-Decoder type architecture for the Transformation Net causes visible noise (artifacts).
We could improve the final results of the method by changing the Encoder-Decoder architecture untill we improve the results of this experiment.

2. **Comparing Adam and L-BFGS** (Limited memory Broyden-Fletcher-Goldfarb-Shanno) **optimizers for the Gatys method**: The main difference between them is that Adam is a first-order optimizer, while L-BFGS is a second-order optimizer. To achieve the same results:
   - L-BFGS: 10 iterations, 1 minute per image
   - Adam: 1000 iterations, 4 minutes per image

Observations

- The downside of the Gatys' method is **having to generate a random image each time and optimizing it**. Dumoulin et al. provide a solution for this problem: **the combination of the Transformation Net and the style parameters**.
- The results for the Gatys et al. method are notably more detailed because it allowed me to use larger images. In the case of Dumoulin et al. I experienced **some limitations during training** (the need to use smaller images, a lot more training time)

Conclusions

Gatys et al.: I was impressed with the results I got following this approach. The images are very detailed and the transfered style really stands out.
Dumoulin et al.: Despite of the limitations met, I definetely saw that the resulting images capture multiple styles correctly. This idea is finally much more efficient and practical. Also, it is impressive that such a small number of style parameters can have such an impact on the final image.

References

[1] Matthias Bethge Leon A. Gatys, Alexander S. Ecker. A neural algorithm of artistic style. Sep 2015.

[2] Manjunath Kudlur Vincent Dumoulin, Jonathon Shlens. A learned representation for artistic style. Feb 2017.