1 Part 1

1.1 Feature inversion

Overall, the optimization process converges effectively, the noise image can closely approximate the target feature map.
The inverted image exhibits similar patterns and features as the original target image. This demonstrates the ability to reverse-engineer and understand the representations learned by the VGG-19 network.

1.1.1 Shallow Layers vs Deep Layers

By experimenting with different layers at various depths of the VGG-19 network, one can gain insights into the hierarchical nature of deep neural networks:
As can be observed from the following experiments (inversion of layers of different types and at different depths):

Figure 1: Feature inversion over the first convulsion layer

Figure 2: Feature inversion over the fifith convulsion layer

Figure 3: Feature inversion over the tenth convulsion layer

Inversion over the shallower layers results in better performances and converges faster. Since features inversion relays on features that are more spatially localized and have a direct correspondence to pixel-level information in the input image , this observation is consistent to the shallower layers' characteristics:
As shallower layers have smaller receptive fields they tend to capture low-level features (such as edges, corner , colors), meaning they focus on local regions of the image (while deeper layers encode more global, complex and abstract concepts).

1.1.2 Relu Layers vs Pooling Layers

Figure 4: Feature inversion over Relu layer

Figure 5: Feature inversion over pooling layer

ReLU layers are effective for extracting detailed textures and edges because introduces non linearity into the network, which helps capturing high frequency features from the target image which are typically associated with fine textures, edges, and small-scale patterns. On the other hand, the VGG19 nn also uses max pooling layer, which downsample the feature maps, reducing their spatial dimensions while retaining the most prominent features.

Inconsistency to the fact that feature inversion relays on features that are more spatially localized.

1.2 Texture synthesis

In the next following section we will focus on the "Picasso" texture as its appear in the creation "Seated Nude, 1909 by Pablo Picasso":

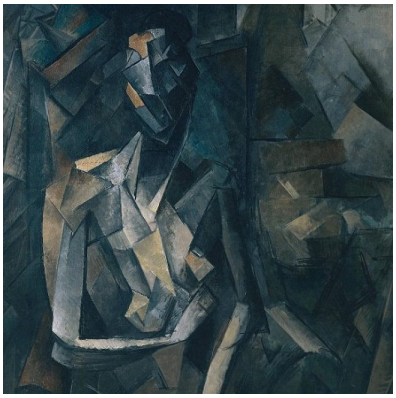

Figure 6: Picasso Texture

1.2.1 Shallow Layers vs Deep Layers

As can be observed from the different experiments (with inversion of layers of different types and at different depths):

Conversely to the previous experiments, dealing with texture synthesis (using Gram matrices), re-

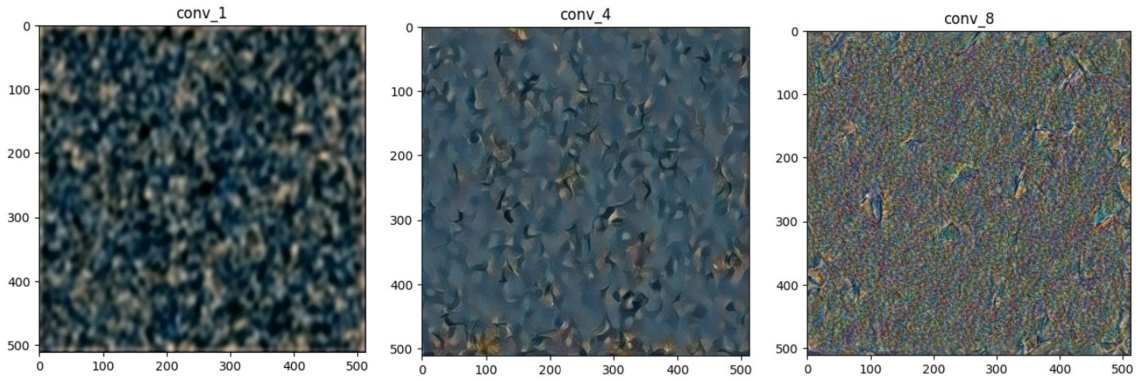

Figure 7: Texture inversion over the first or the fourth or the eighth layer only

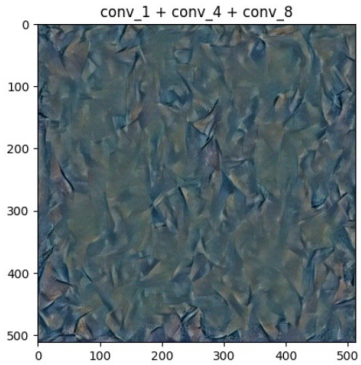

Figure 8: Texture inversion over the first the fourth and the eights layers

quired both deep and shallow layers. Intuitively, texture synthesis among others things, aiming to capture global patterns and color variations.
In one hand, as explained before, color features and high frequencies features are better captured in the shallow layers of the NN because they are closer to the input image (3 channel RGB), in contrast to the deep layers where the layers' channel are higher and struggling to capture the color patterns (compare to the shallow layers).
On the other hand, global patterns and semantic concepts (which are crucial to the texture synthesis) are better captured in the deep layers.
As can one see, a successful synthesis occurs when the loss is taken over layers from the shallow ones and from the deep ones. Synthesis that performed over the shallow layers only, are indeed captured relevant pattern, mainly, the color patterns and the high frequency features (such as rapid changes in pixel intensity or the presence of fine details). Where synthesis that performed over the deep layers only, indeed capture the global pattern such as structure, content, or statistical properties. but lacking to capture the fine grind details (e.g, color variation). From this experiment one can conclude that for texture synthesis more complex pattern and data need to be driven from the image.
Those empirical results are consistent to the VGG-19 architecture, as the deeper layers have a bigger receptive field , because of the stacking of convolution layers with pooling operations allowing neurons in deeper layers to capture more global and high-level information, and capable of capturing more complex and abstract features. These deeper layers have access to a larger context and can integrate information from a broader region of the input image, enabling them to learn higher-level representations and semantic concepts (larger receptive fields and semantic information). can capture these long-range dependencies and generate textures that are consistent over larger spatial regions.

1.2.2 ReLU Layers vs Pooling Layers

Empirically, compare to the part of feature inversion (where we notices optimized over the second layer, ReLU performed better then the pooling layer) dealing with texture synthesis the performance over the ReLU and the Pooling (both over the second layer) where quite equals. Nevertheless when we used ReLU layer we were able to preserve fine-grained textures and local variations from the target image, while pooling layer was able to capture larger-scale patterns and the overall structure of the texture. we also noticed that the image generated using ReLU layer we got moderate color and texture variation compare to pooling layer where we got extreme color and texture variation

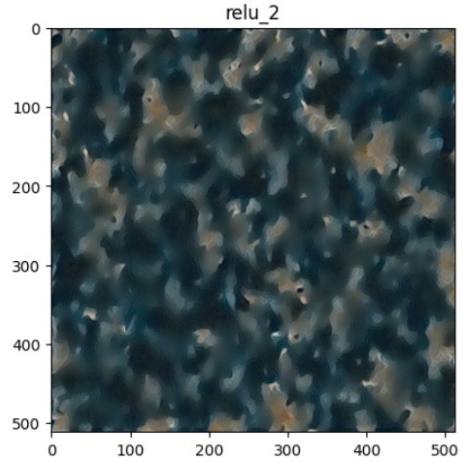

Figure 9: Texture inversion over the ReLU layer (second layer)

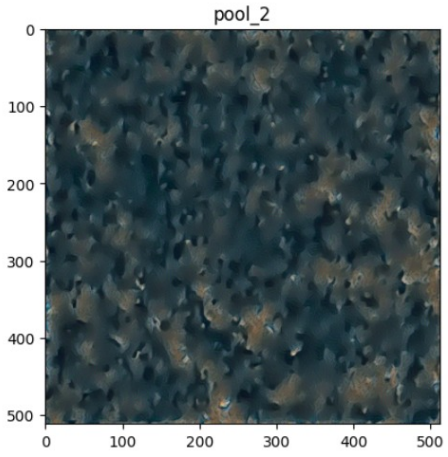

Figure 10: Texture inversion over the Pooling layer (second layer)

Comparing optimization between Convolution layers and ReLU layers both over the first the fourth and the eighth layers:

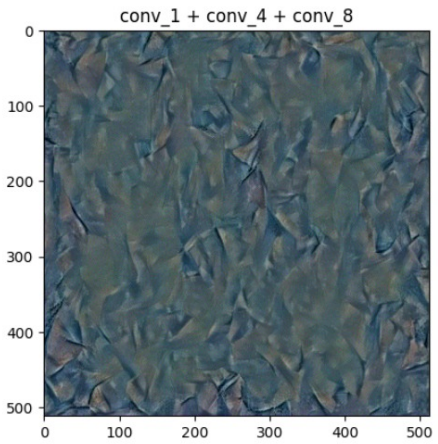

Figure 11: Texture inversion over the Convolution layers (over the first the fourth and the eighth layers)

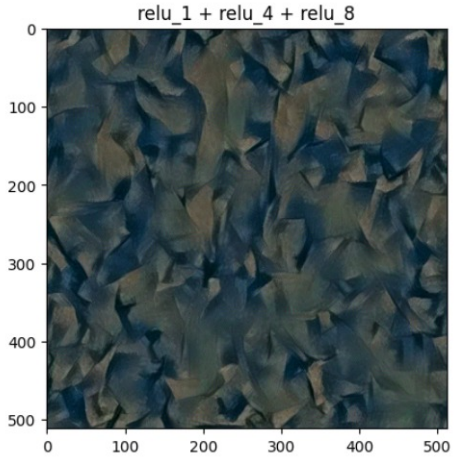

Figure 12: Texture inversion over the Convolution layers (over the first the fourth and the eighth layers)

1.3 "MeanVar" style loss

To compute the "MeanVar" style loss of each activation channel we computed the mean and the variance of every channel separately, then we concatenated the mean and the variance and got a vector. This type of loss computation encourages the input image to match the style of the target image in terms of statistical properties. First and foremost texture synthesis over the MeanVar loss results in high performances as can be observed. The optimization succeeded to minimize the loss (converge) and to synthesis quite impressively the texture. Generally, we expected for lower performances compare to the previous loss (Gram matrices) as this loss use vectors from lower dimension which yields to loss of the data of the target image- given an image with N Chanel's, the gram matrices shape is (NxN)/2 compare to MeanVar shape of 2N

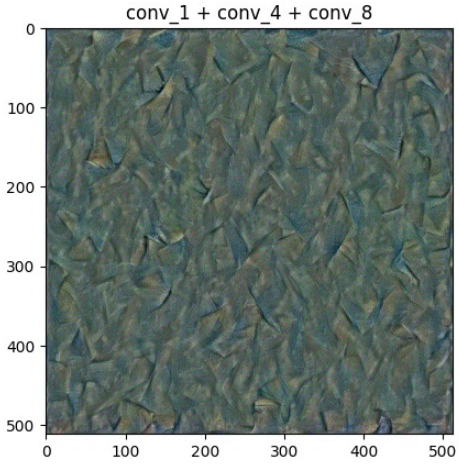

Figure 13: Texture inversion over the first the fourth and the eighth layers - loss over MeanVar

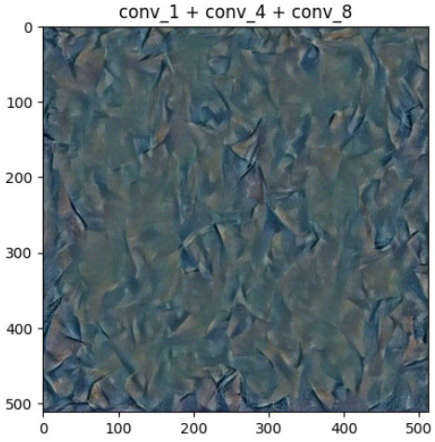

Figure 14: Texture inversion over the first the fourth and the eighth layers - loss over Gram matrices

One can noticed that calculating the MeanVar loss in shallow layers (which are crucial for capturing the color feature) results in poor performances comparing to the Gram matrices loss.

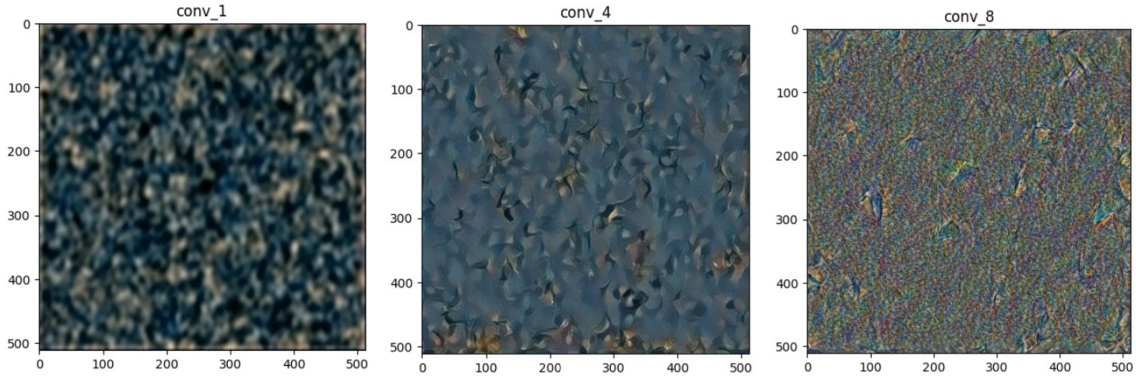

Figure 15: Texture inversion- loss over Gram matrices

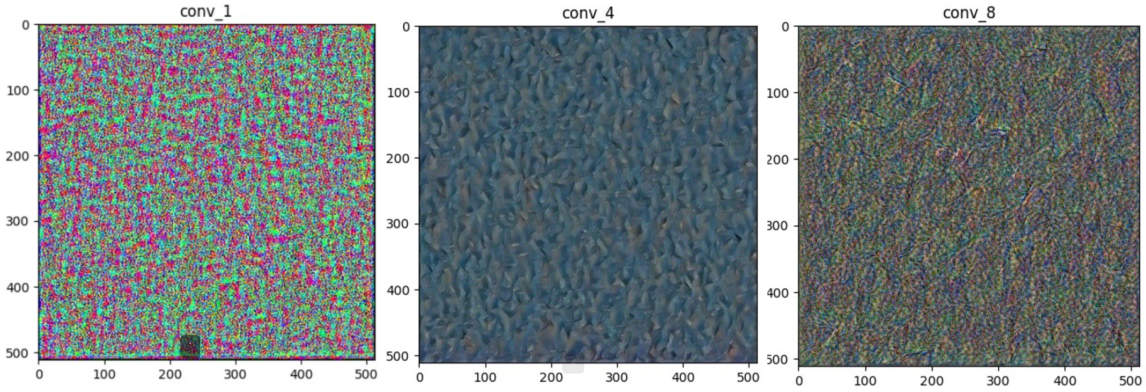

Figure 16: Texture inversion- loss over MeanVar

2 Part 2

2.1 Part A

The results of diffusion obtained with style loss guidance over convolution layers 1 to 10:

Figure 17: Style loss guidance using Gram matrix loss

Figure 18: Style loss guidance using MeanVar loss

As expected both loss guidance yields impressive results. Nevertheless, it is noticeable the Gram Matrix loss yields (as expected) higher performances then the MeanVar loss.In the aspect of the textures' pattern the MeanVar loss manage to capture the texture of the style image (the sharp curves the "triangles". On the other hand its seems like in most of the image part the model didn't capture the textures' colors.

Figure 19: Style loss guidance over pooling layers as-well

In order to compare the results of diffusion obtained with style loss guidance to those obtained by the optimization-based Neural Style Transfer we use a generated image which contain similar content "a photograph of an astronaut riding a horse ":

Figure 20: Style loss guidance base line image

Figure 21: Style loss guidance using Gram Matrix loss

Figure 22: Style loss guidance using MeanVar loss

In the same manner as mentioned before, both loss guidance yields impressive results. Nevertheless, it is noticeable that the Gram Matrix loss yields (as expected) higher performances then the MeanVar loss. Again, as before, the MeanVar loss didn't capture the textures' color (while it did capture quite well the textures' patterns).

2.2 Part B

The image that obtained by ordinary text-to-image stable diffusion, with the text-based style description added to the prompt.
The prompt: "A photograph of an astronaut riding a horse with style of Vincent Van Gogh, The Starry Night"

Figure 23: Text-to-image stable diffusion, with the text-based style description added to the prompt

Figure 24: Text-to-image stable diffusion, the prompts' style image, Vincent Van Gogh, "The starry Night"

In comparison, the classifier-guided diffusion generate the following images given the prompt "A photograph of an astronaut riding a horse" with the same image Vincent Van Gogh, "The starry Night" as a style guided loss.

Figure 25: Classifier-guided diffusion with the same style image Vincent Van Gogh, "The starry Night"