

Neural Networks for Images 2023 - Exercise #4

Ron Yosef and Bar Rousso

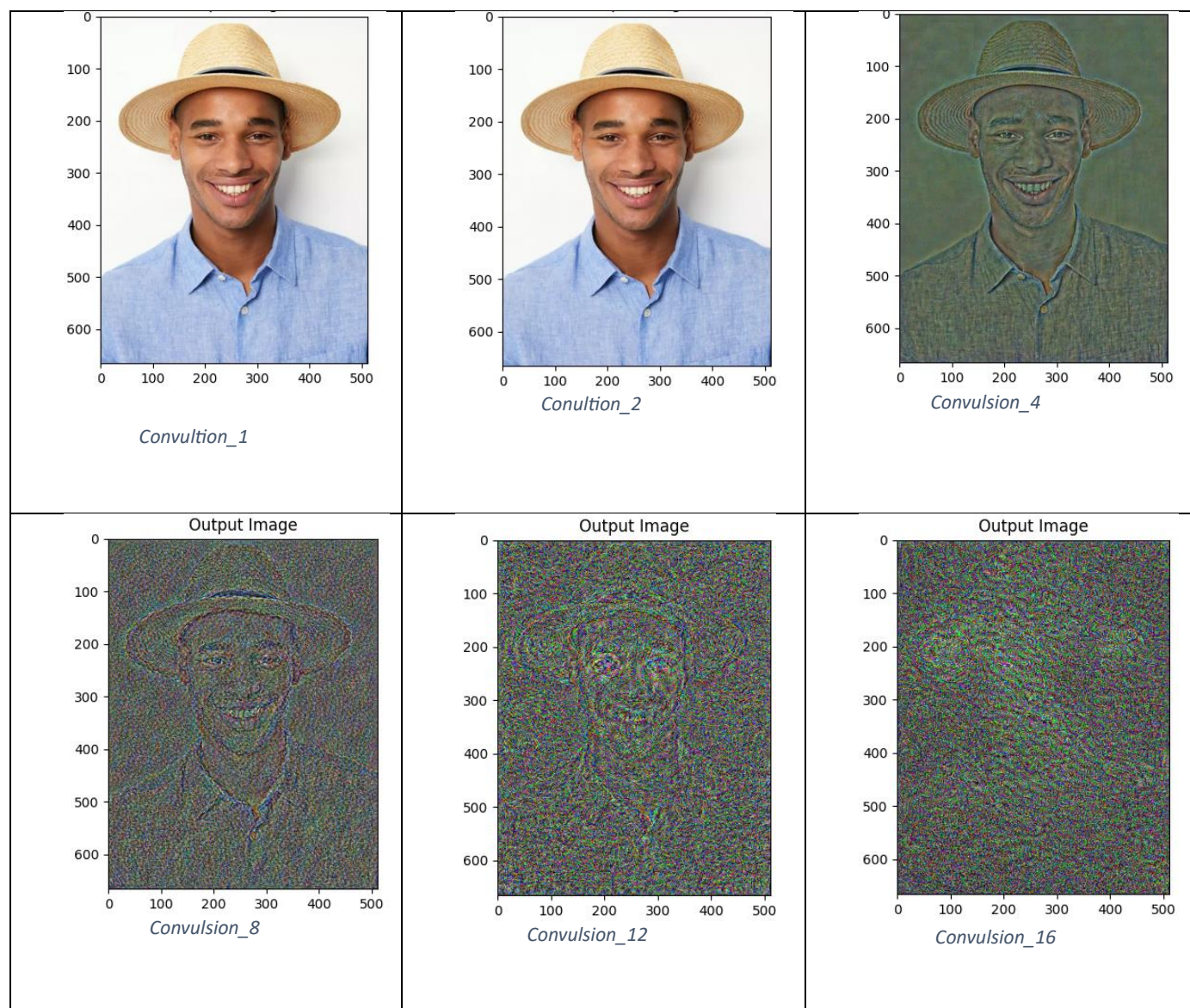
Part 1

First question

The optimization of the input image (noise image) with the content loss resulted in an image that looks very similar to the target image when using shallow layers. For example, taking the first convolution layer resulted in an image that looks exactly the same as the target image to the naked eye. As we increase the layer depth, the image becomes less similar to the target image. Using the last convolutional layer the resulting image appears like a noisy image, with only weak contours seen in the image.



Target image



We did not see a significant difference between the different types of layers in the same depth, and example is presented below.



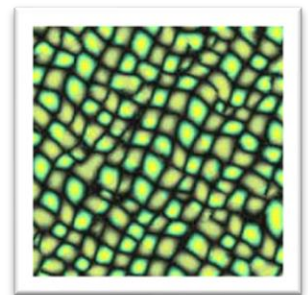
Convulsion_1



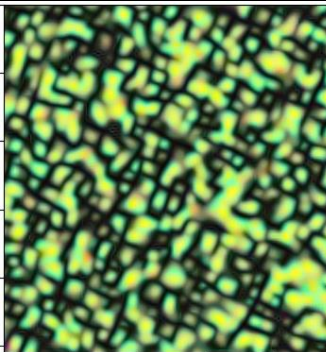
Relu_1

Second question

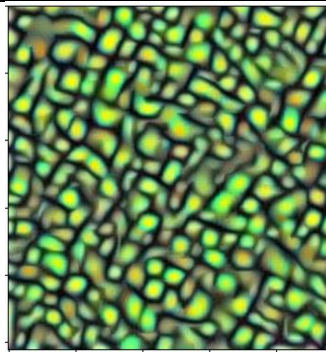
The optimization of the input image (noise image) with the style loss resulted in an image that looks very comparable to the target image when using several deep layers of the model. For example, taking convolution layers 1-7 resulted in an image that looks much more similar to the texture image, rather than taking only 1-3 layers. Using only the first layer, the image seems like some distribution of colors, without any shapes or detectable texture. As we increase the depth, we see features like better colors and shapes.



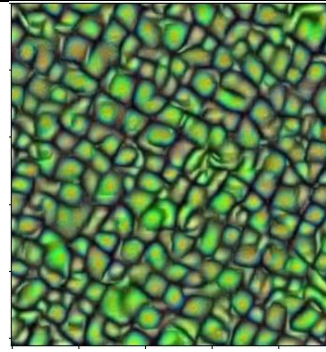
Texture image



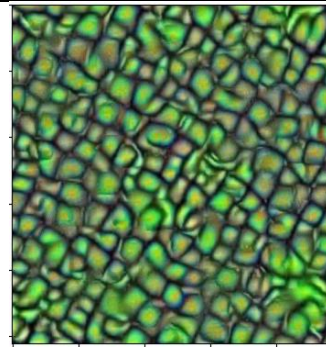
Convulsion 1-3



Convulsion 1-6

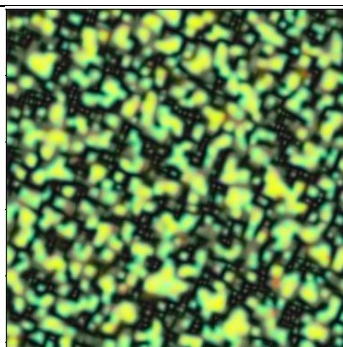


Convulsion 1-10

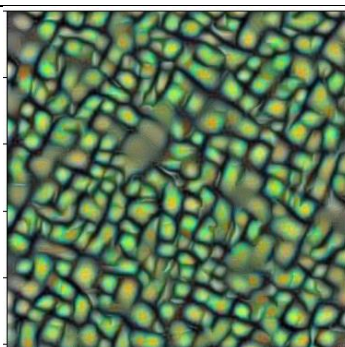


Convulsion 1-16

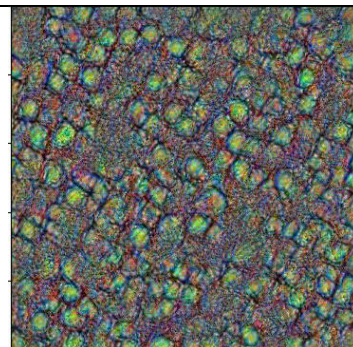
Using only a single layer resulted in a difficult learning problem for the model, the loss took more iteration and steps to converge, and the results does not look similar to the texture image.



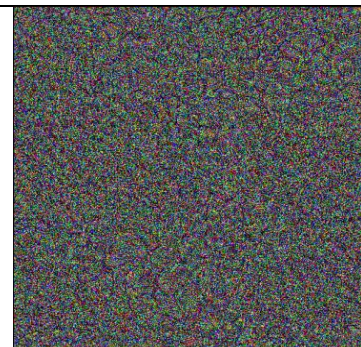
Convulsion 1



Convulsion 4



Convulsion 8

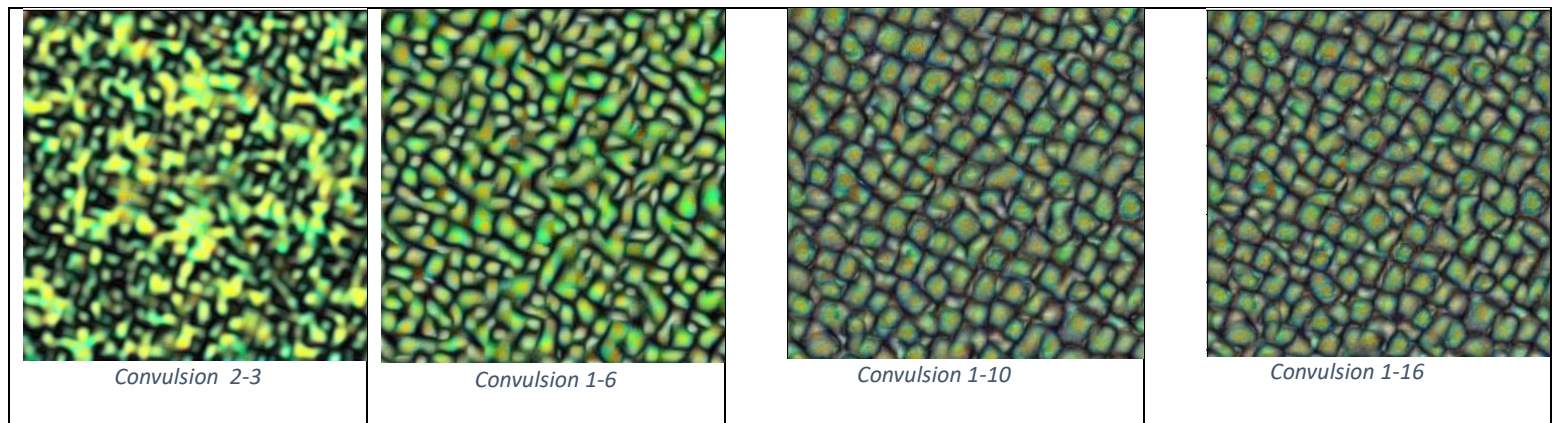


Convulsion 16

Third question

The mean and variance loss results looks very similar to the gram matrices results. A notable difference between the two is the brightness of the color. The gram matrices result has brighter colors, while the mean and variance loss have darker colors. In both cases the result looks similar to the texture image.

The mean and variance loss required significantly more time for each iteration. This is expected because of the high cost of the mean and variance computation for each channel of each layer.



Part2

In order to implement a style loss-guided diffusion, we modified the ordinary text-to-image stable diffusion in the following way:

In each timestep t :

- Using UNET model we predict the noise tensor N_t that transfers the less noised image X_{t-1} to the current noised image X_t .
- Using the scheduler model and N_t , we predicted the clean image X_0 .
- We computed the gradient (w.r.t N_t) of the style loss of X_0 compared to the target style image.
- Using that gradient, we computed a new noise tensor \widetilde{N}_t that transfers another less noised image \widetilde{X}_{t-1} to the current noised image X_t .
- Using the scheduler module and \widetilde{N}_t we get the predicted image \widetilde{X}_{t-1} .

The difference between X_{t-1} and \widetilde{X}_{t-1} is that \widetilde{X}_{t-1} has a **more** similar texture to the target style image, rather than X_{t-1} .

In other words, that gradient (multiplied by -1) points in the direction where the image becomes **more** similar to the target style image.

Thus, we eventually converge to an image \widetilde{X}_0 with a texture similar to the target style image.

First question

Comparison between the following methods:

- Style loss guided diffusion
- Neural Style Transfer using Gram-based style loss
- Neural Style Transfer using MeanVar-based style loss

In the first method, we used the following target style image ('seated nude' by Picasso):



In addition, we used the prompt *"a man wearing a straw hat"*.

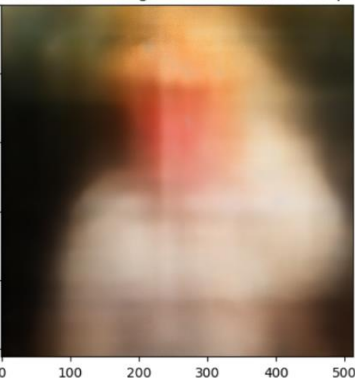



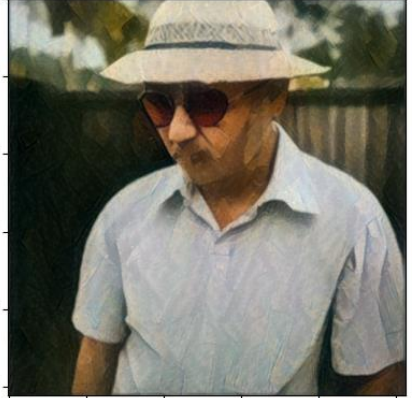


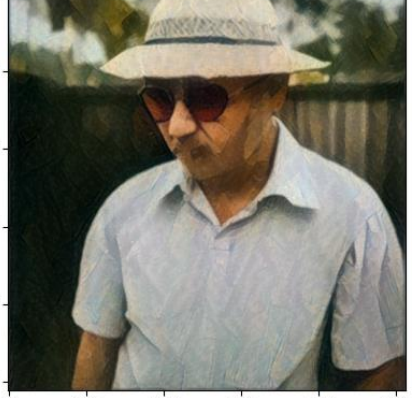



In the second and third methods, we also used the above target style image, and started our optimization on the following image:



Notes:

- Like in the paper of Neural Style Transfer, we started our optimization from a content image and not a random noised image.
- For a better comparison, the above image is actually the final image that came from ordinary text-to-image stable diffusion when we only used the prompt *"a man wearing a straw hat"*.

The following table represent the resulted images accepted on each method before and after five time equals intervals:

Style loss guided diffusion	Neural Style Transfer using Gram-based style loss	Neural Style Transfer using MeanVar-based style loss
<div>index = 0, timestep = 999, style_loss = 0.017985917627811432</div> <div>Intermediate clean image for index = 0, timestep = 999</div> 	<div>Output Image</div> 	<div>Output Image</div> 
<div>index = 100, timestep = 799, style_loss = 0.005991274490952492</div> <div>Intermediate clean image for index = 100, timestep = 799</div> 	<div>Output Image</div> 	<div>Output Image</div> 
<div>index = 200, timestep = 599, style_loss = 0.00130206230096519</div> <div>Intermediate clean image for index = 200, timestep = 599</div> 	<div>Output Image</div> 	<div>Output Image</div> 
<div>index = 300, timestep = 400, style_loss = 0.0006960308528505266</div> <div>Intermediate clean image for index = 300, timestep = 400</div> 	<div>Output Image</div> 	<div>Output Image</div> 



Conclusion:

From the table above, we can clearly see that the guided diffusion method outputs a final image with a texture that is much more similar to the target style image, compared to the two resulting final images that came from the Neural Style Transfer.

More precisely, compared to Neural Style Transfer's images, we can witness the diffusion's final image almost doesn't contain thin features of the subject (i.e., nose, eyes, shirt collar, etc.) and only maintains the general shapes.

One explanation for this phenomenon is the fact that in the diffusion process, we start from a noised image, thus we have more freedom to generate an image with a more similar texture to the target style image at the expense of some features of the subject.

Indeed, we noticed that if during the diffusion process, we get an intermediate image that contains much more of the subject features with almost no desired texture, then it will be almost impossible to converge to a final image with a similar texture to the target style image.

Second question

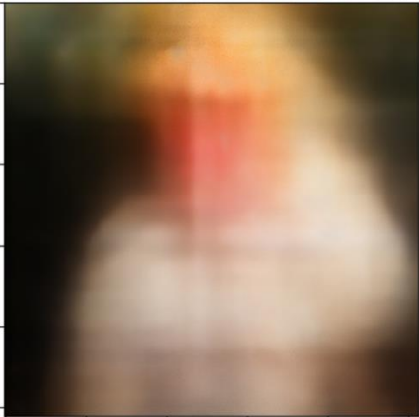
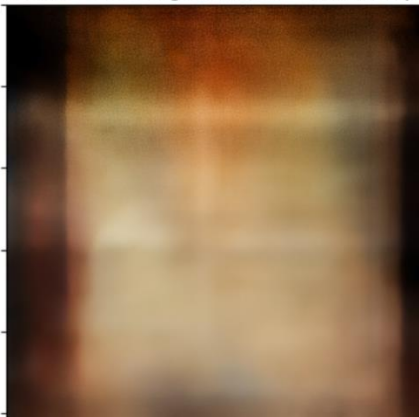

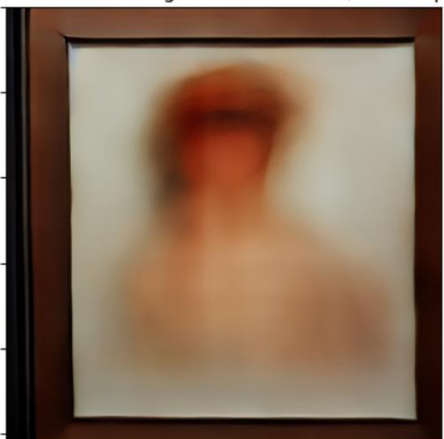
Comparison between the following methods:

- Style loss guided diffusion
- Ordinary text-to-image stable diffusion, with the text-based style description added to the prompt.

Like before, In the first method, we used the target style image ('seated nude' by Picasso), and the prompt *"a man wearing a straw hat"*.

In the second method we used the prompt: *"a man wearing a straw hat with similar texture to the painting named 'seated nude' by Picasso"*.

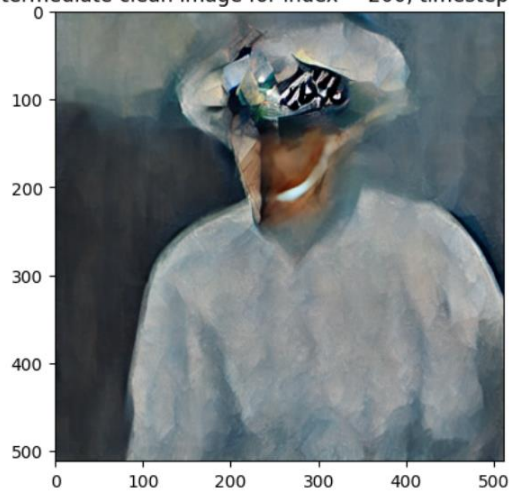
The following table represent the resulted images accepted during the diffusion process on a pre-selected timesteps:

Timeste p	Style loss guided diffusion	ordinary text-to-image stable diffusion
999	<div>index = 0, timestep = 999, style_loss = 0.017985917627811432</div> <div>Intermediate clean image for index = 0, timestep = 999</div> 	<div>index = 0, timestep = 999, style_loss = 0.036674268543720245</div> <div>Intermediate clean image for index = 0, timestep = 999</div> 
799	<div>index = 100, timestep = 799, style_loss = 0.005991274490952492</div> <div>Intermediate clean image for index = 100, timestep = 799</div> 	<div>index = 100, timestep = 799, style_loss = 0.02551405131816864</div> <div>Intermediate clean image for index = 100, timestep = 799</div> 

599

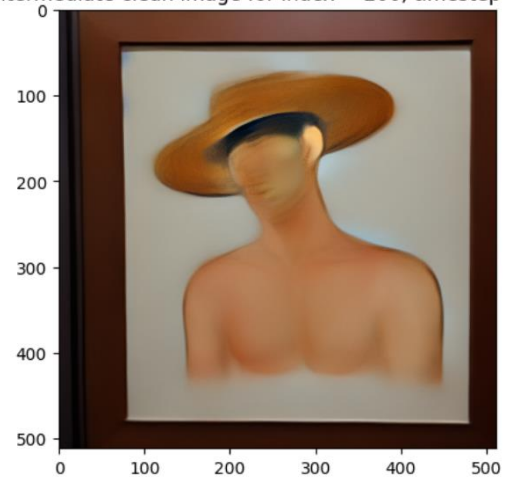
index = 200, timestep = 599, style_loss = 0.00130206230096519

Intermediate clean image for index = 200, timestep = 599



index = 200, timestep = 599, style_loss = 0.027919035404920578

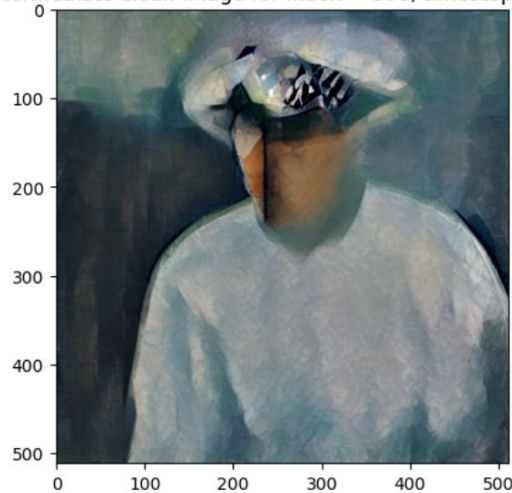
Intermediate clean image for index = 200, timestep = 599



400

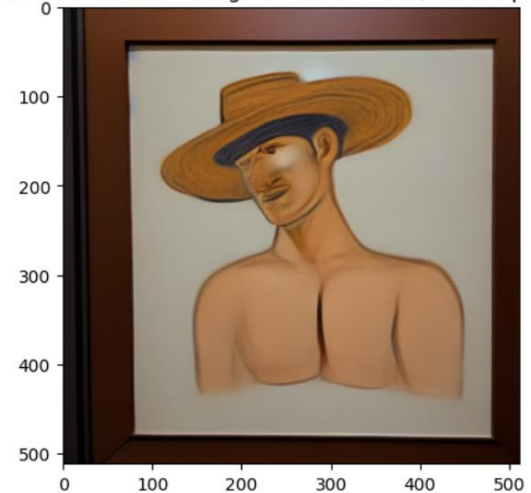
index = 300, timestep = 400, style_loss = 0.0006960308528505266

Intermediate clean image for index = 300, timestep = 400



index = 300, timestep = 400, style_loss = 0.033757179975509644

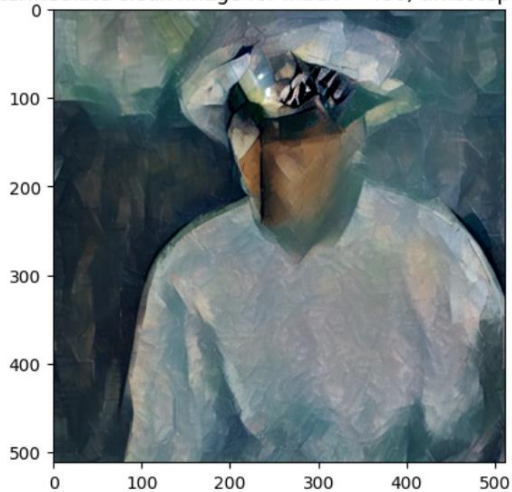
Intermediate clean image for index = 300, timestep = 400



200

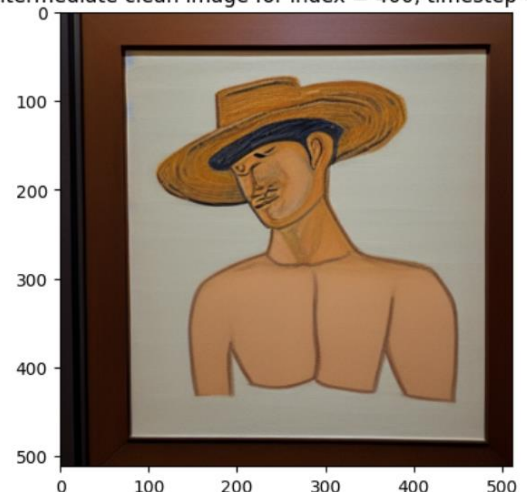
index = 400, timestep = 200, style_loss = 0.0002050118346232921

Intermediate clean image for index = 400, timestep = 200



index = 400, timestep = 200, style_loss = 0.03498648852109909

Intermediate clean image for index = 400, timestep = 200





Conclusion:

For the second method, although we tried to use several different prompts, we never succeed to receive a final image that has a similar texture to the target style image. One explanation for this phenomenon is that the prompt cannot encode enough information to capture the accurate texture of the target style image, as the common phrase "*A photo is worth a thousand words*".

On the other hand, for the first method, we can clearly see that the final image has a similar texture to the target style image. This makes sense as we actually used that target style image during our diffusion process, which of course contains the encoded information of its own texture.