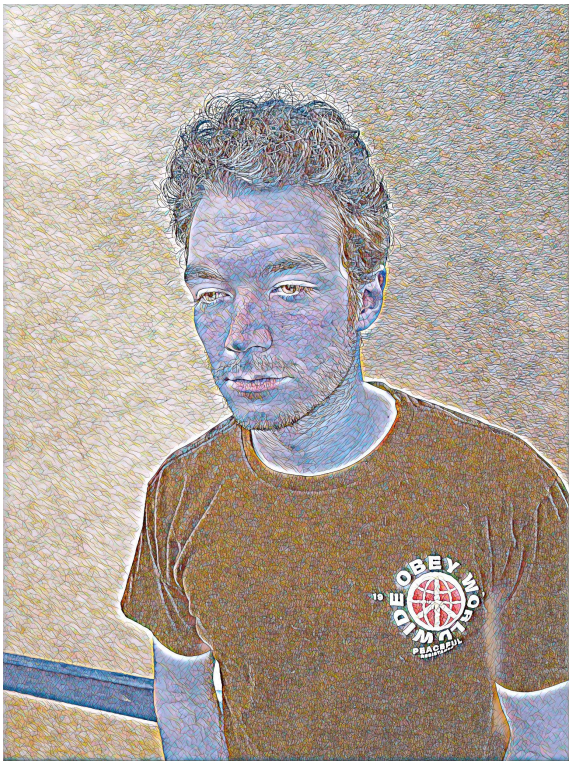
ART AND MACHINE LEARNING
CMU 2022 SPRING
PROJECT 2

Surrealist Object and Image Avatar (group 5)



Eric Menq, Ryan Gess, Serena Hou, Sachit Goyal

Team Background

Eric Menq (emenq): Electrical and Computer Engineering. No ML experience, some programming experience
Ryan Gess(rgess): Electrical and Computer Engineering. No ML experience, some programming experience
Serena Hou(shou): Fine Arts. No ML Experience, but plenty of arts experience
Sachit Goyal(smgoyal): Electrical and Computer Engineering. No ML experience, some programming experience

DESCRIPTION

Our concept involved creating surreal objects with texture transfer. We looked to give objects and people a new meaning by recreating them with unexpected textures. We used Stross style transfer, fast style transfer, and attempted background removal and fast NERF. We successfully implemented style transfer and background removal, but were unable to incorporate that work into NERF. Our results met our artistic goals and our learning goals with the technical aspects of style transfer. However, we hoped to bring out greater surreality by making our art 3D, and hope to explore and learn more about NERF so that we can successfully implement it in the future.

Concept

To begin, we explored the creation of surreal objects with tactile extension and texture manipulation. We aimed to create metaphors and manipulate symbols without assigning a rigid meaning to the image. Our major source of inspiration is the Fur-covered tea set made by Meret Oppenheim in response to the surrealist movement. The core of the movement is to challenge excessive rationality imposed on postwar society and explore the unconscious part of the human psyche. To replicate a similar style of art, we used style transfer to manipulate the materiality of the object.

While choosing the image, we've taken into consideration the perspective of the image and its potential effect on the viewer. Our traditional sense of spatial orientation has been significantly altered by the development of new technologies. AI generated art specifically uses multiple perspectives that cannot be achieved in real life. We intentionally chose to use a linear perspective to combat the inherent sense of artificiality in code generated images and further emphasizes the absurdity of the object

At the very beginning, photography was seen as a signifier of trivial realism, the direct assessment of perceived reality and universal truth. Many performers and photographers in the 20th century have tried to challenge these presumptions by modifying the photos in the darkroom. Yves Klein's famous balck and white photograph “leap into the void”

captures him jumping out of a two story building without any protection. It is criticized for blurring the line between authenticity and fakery for he modified the image and did not perform the act depicted in the photo.

The invention of photoshop and the development of AI generated images over the years shifted how the world views photography. Image nowadays no longer serves as a documentation of reality but the illustration of preferred or imagined reality.

As we will mention later, due to our results, we decided to pivot from our original artistic idea.

Transformer generation was led by Cindy Sherman, and was performed for the camera instead of the live audience, combining the concept of documentary photography and performance art. She created fictional characters that exist solely in the realm of photography and essentially created the pre-photoshop version of image avatar.

We wanted to create an alter ego that's part real and part fantasy, an image avatar that illustrates our preferred reality or sense of self.

Technique

Our technique was to use STROSS style transfer[1] and fast style transfer[5] to change the style of people. We then attempted to use OpenCV background to isolate their image from the background and then create a 3D model with NERF [2], [3], [4].

Process

In this project, we developed a four stage pipeline which consisted of

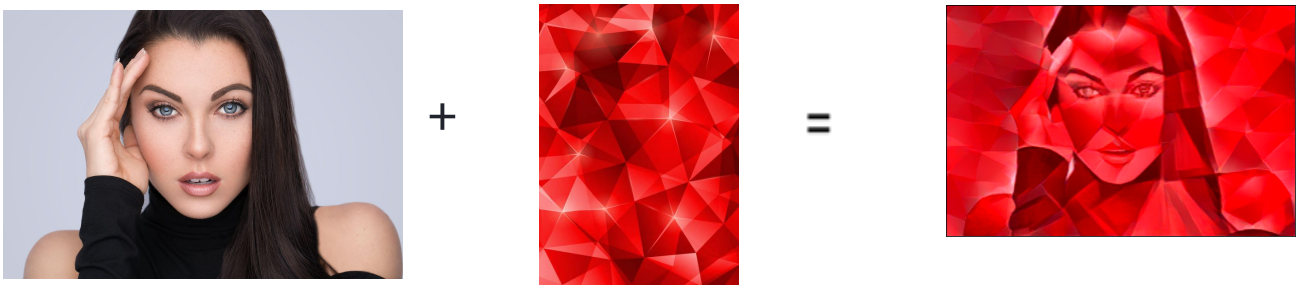
1. Testing and experimenting with style transfer on various images with strotss notebook and Fast Style Transfer
2. Performing style transfer on dataset
3. Removing background surrounding the target object
4. Running Nerf on the stylized dataset

Our initial approach was to use STROTSS to see how the texture of objects would change under style transfer. We were not particularly happy with the results, as the edges did not look great, and some of the 3D shape was lost after style transfer. We did notice some textures looked good on images with a lot of structure, such as living things (animals, humans, etc.) However, we did not feel as though this fit our original artistic goals to change texture in a meaningful way. As a result, we tried a different

strategy. Next, we used fast style transfer, which we felt had more meaningful results. We used OpenCV to remove the background. Finally, we attempted to project our results into a 3D space with NERF, trying both fast NERF and PlenOctree Volume Rendering, but we ran into many errors and were unable to achieve this.

STROTSS RESULTS

We were successful in using STROTSS style transfer with a slight modification that allowed us to use our own images uploaded to Google Drive. We also modified our code to loop through a folder of images and apply the style to a set of images.



https://colab.research.google.com/drive/1-Bk1XfuXKFBQ2KHyb7cTOKlLFj35Fb-z?usp=sharing

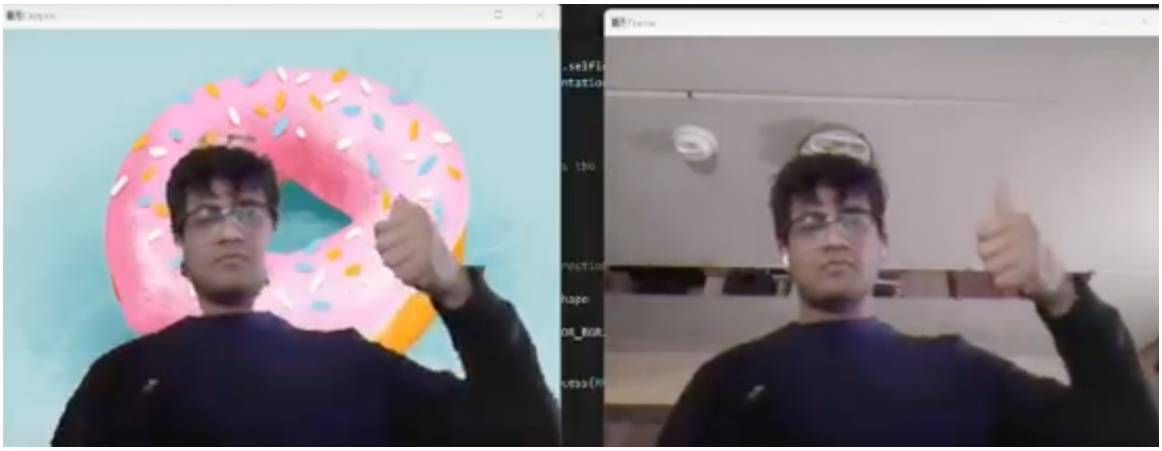
FAST STYLE TRANSFER RESULTS

We really liked the results with fast style transfer as most of the resulting images left us with nice edge detectable objects that we could apply object detection to and remove the backgrounds. We used Fast Neural Style Transfer [5] repository for this and did not modify the code at all.

\+ tokyo_ghoul.pth =

EDGE DETECTION RESULTS

Here, we experimented with edge detection using DataFlair. We successfully removed the background. However, since our NERF was never successfully implemented, we decided not to run background removal on our images as we liked having the background for our 2D results.



NERF RESULTS

In this step, we encountered many errors and tested various open source code to perform NERF on our datasets. One error we encountered was, 'file not found "poses_bounds.npy"'. Upon looking into this, we realized that this file contained the camera positions of the images captured. Therefore, we decided to use an existing dataset that had camera positions. This alleviated the error however the code stops running with output "Killed." We are unsure why this is at the moment but we plan to look into this further.

```
Data:
(55, 3, 5) (55, 3024, 4032, 3) (55, 2)
HOLDOUT view is 20
Loaded llff (55, 3024, 4032, 3) (120, 3, 5) [3024.      4032.       416.23373] ../trex
Auto LLFF holdout, 8
DEFINING BOUNDS
NEAR FAR 0.0 1.0
Found ckpts []
get rays
done, concats
Killed
(pytorch_p38) ubuntu@ip-172-31-2-224:~/nerf-pytorch$ python run_nerf.py --config configs/trex.txt
Loaded image data (3024, 4032, 3, 55) [3024.         4032.          416.23374465]
Loaded ../trex 14.611281520672364 128.97241851887296
recentered (3, 5)
[[ 1.0000000e+00 -8.9834629e-10 -9.6571471e-09  6.5023249e-09]
 [ 8.9834634e-10  1.0000000e+00  7.2768644e-09 -6.6377899e-09]
 [ 9.6571471e-09 -7.2768644e-09  1.0000000e+00  4.0639531e-10]]
Data:
(55, 3, 5) (55, 3024, 4032, 3) (55, 2)
HOLDOUT view is 20
Loaded llff (55, 3024, 4032, 3) (120, 3, 5) [3024.      4032.       416.23373] ../trex
Auto LLFF holdout, 8
DEFINING BOUNDS
NEAR FAR 0.0 1.0
Found ckpts []
```

Repositories we referenced: [1], [2], [3]

Reflection
Although we did not achieve the results we were initially hoping for, we felt we were still able to try many techniques and learn how they work. We were able to explore two different artistic approaches and apply the techniques we learned in class to try to achieve our artistic goals. Looking back, our initial idea may have been overambitious, as this is the first project in the class and all of our members' first experience with machine learning. Although some of us have had programming experience, I think we underestimated the challenges we would face with AWS and running algorithms on large datasets due to our lack of experience. Moving forward, we are excited to build upon the experience we gained in this project and apply it to future projects.

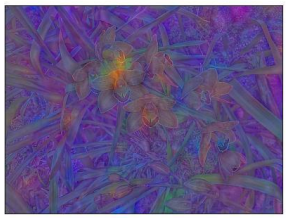
RESULT

Below are the results we liked the most.

The style transfer radically changed the tone of the original photo, transforming a photo of a hot dog to what we imagine it would look like in 10 years. The style transfer warps the sense of time that photographs preserve.

The 3rd picture is reminiscent of a dreamlike hysteria. While none of us have used LSD, the colors created remind us of the way people describe the effects of the drug. LSD is a classic example of a surreal reality humans can experience, and this photograph is our attempt at recreating that.

Finally, the last image was one of the style transfers on our portraits that we felt created an alter ego. Blue often symbolizes sadness, and the segmentations of the mosaic portray an almost ghostlike or dreamlike state.


CODE

https://github.com/ryanrmg/10-615-project-2
Some images of datasets excluded to satisfy github upload limitations

REFERENCE

[1] Peter Schaldenbrand, Strotss Style Transfer , 2019
https://colab.research.google.com/drive/1pBfUg0HAiLkhqW_vqzR353TA5RGIBW98

[2] Krishna Murthy, nerf-pytorch
https://github.com/krrish94/nerf-pytorch

[3] Yen-Chen Lin, nerf-pytorch
https://github.com/yenchenlin/nerf-pytorch

[4] Tiny Nerf
https://colab.research.google.com/github/bmild/nerf/blob/master/tiny_nerf.ipynb

[5] Fast Neural Style Transfer
https://github.com/rrmina/fast-neural-style-pytorch