Exploring Pix2Pix and CycleGAN for Image Alteration and Style Transfer

Jue Hou Azaan Barlas Santiago Valencia Nikhil Khandekar

May 7, 2023

1 Motivation and Impact

The motivation behind our project was to explore the power of using image-to-image and video-to-video translation alteration applications for both art and real-world use. Our goal was to investigate the effectiveness of Pix2Pix and CycleGAN for creating satellite-to-map images, segmenting facades, and artistic styles transferring on videos. Image alteration is an important area of research that has many practical applications. For example, image restoration can be used to recover damaged or distorted images, while image enhancement can improve the visual quality of images for human observers. Similarly, artistic style transfer can be used to create unique and visually stunning works of art.

We believe that our approach is novel because we re-implement Pix-to-Pix and Cycle-GAN but also apply non-ML image processing methods to help transform and stitch large images and perform a video-to-video translation. By doing so, we hope to achieve better results with fewer resources. Current methods for applying image-to-image models to videos all include changing architectures of machine learning models or using AI. Our approach is different because we utilize non-ML image processing methods to achieve results with fewer resources as opposed to vid2vid, which is one of the few existing deep learning models which can translate between videos but only perform semantic segmentation. This project repository is https://github.com/juehoujhou4/CS445-Spring-2023-FinalProject.

2 Approach

We had two tasks for our project. The first was to implement CycleGAN and Pix2Pix and run it on a dataset to see the results. The second was to investigate the effectiveness of Pix2Pix and CycleGAN for image alteration and style transfer in two different domains: large images and videos. We trained both models ourselves but utilized longer-trained pre-trained models for our final outputs for the art transformations due to time and hardware limitations. For the map images and youtube videos, we used the pre-trained models since training these would take too long.

Pix2Pix and CycleGAN are deep learning models for image-to-image translation that can be trained on paired and unpaired image datasets, respectively. Pix2Pix is a conditional GAN that learns a mapping from input images to output images, while CycleGAN is an extension of Pix2Pix that can learn mappings between two domains without the need for paired training data. To test our implementations, we ran the Facade Dataset for Pix2Pix and we tested the Grumpy Cat dataset for CycleGAN. With these models, we are able to generate new images and videos that maintain the characteristics of the original data while also introducing new features and artistic styles.

For our map outputs, we input a large satellite map image (Google Maps) and break it up into smaller patches with overlap to process in the model. After inputting the patches into the model, we used a stitching method to combine patches and blended the overlap between them. This allowed us to generate high-quality maps that maintained the original features of the satellite data while also introducing new details and styles.

For videos, we first broke up the video into individual frames, then resized all frames to 512x512. We input these into Pix2Pix and CycleGAN. For Pix2Pix output, we utilized many different techniques to improve video quality, mainly histogram matching, bulk blending, and denoising to create a better video. For CycleGAN, we only utilized one method. We then reassembled the frames to create a new video that maintains the original content while also introducing new styles and features.

3 Implementation and Results

Pix2Pix and CycleGANS are two different adversarial networks used to build Image-to-Image translations which was the component to go from one video or image style to another. Our Pix2Pix and CycleGANs models were close implementations of the original papers, which were cited below. Our team was very new to PyTorch and so the Pix2Pix model was largely based on a tutorial (reference [6]) that we watched online to get a gist of how to build and train GANS. We copied the code verbatim for dataloader and data augmentation, but we tried to understand and then implement the code involved in training the functions ourselves. Team members had experience writing neural networks in PyTorch, however, so we implemented the network itself and the loss functions from scratch.

After gaining the relevant experience needed to understand the non-neural network components of GANS, we were able to implement CycleGANs from scratch by following the linked paper. Specifically, we implemented the cycle loss, identity loss, adversarial loss, the training loops, and the Generator and Discriminator neural network ourselves. The only component we used from a source code was again for the data augmentation and the data loader. We decided to demonstrate that we were able to successfully implement these two models by running this through the Facade data set which would segment facades by features such as pillars, doors, and windows. We did the same for CycleGAN as well. The results can be shown below.

Our project was implemented in Python using PyTorch and OpenCV libraries. Due to hardware limitations, we utilized pretrained models for the map section and trained our own models for the artistic video transformation. We used Google Colab for training the models due to the high computational requirements. We trained our data on an AWS instance (g4dn.8xlarge) yet the training still took an extremely long time. All source code and preprocessed data are available on GitHub.

Our results show that CycleGAN outperformed Pix2Pix in both the map and artistic videos. Even after image processing techniques, CycleGAN was superior and did not require additional processing compared to Pix2Pix. We think this is because of how CycleGAN is structured. For the map section, both models were able to generate maps that looked close enough to a Google Maps output of a map. Meanwhile, for the art transformations, CycleGAN performed much better than Pix2Pix and maintained the structures of objects in the pictures while also having less noise than Pix2Pix.

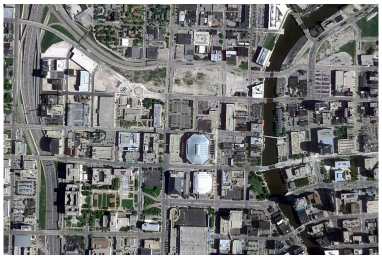

Figure 1: Original satellite input

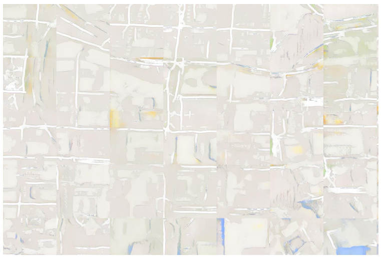

Figure 2: Pix2Pix output

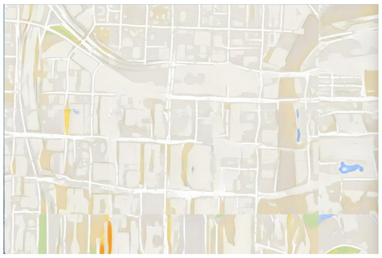

Figure 3: CycleGAN output

When it came to maps, Pix2Pix was noticeably worse at producing a map output from a satellite picture input. We broke up our satellite image into square patches with an overlap of 50% of patch size. We then input these patches into the model. With this output, we stitched the patches together by blending the overlapping sections. We did not need to do this with CycleGAN as the output was already very high quality. The examples can be seen inside the CycleGAN and Pix2Pix folders of our code.

Our artistic output, includes our original picture, the Pix2Pix output, and the CycleGAN output. Note that due to time constraints, we trained different art datasets on the Pix2Pix and CycleGAN models. Our final video products are https://www.youtube.com/shorts/hspKupwXvsM for CycleGAN, and also another one which is for Pix2Pix https://www.youtube.com/shorts/bdYzgwVDj8Y

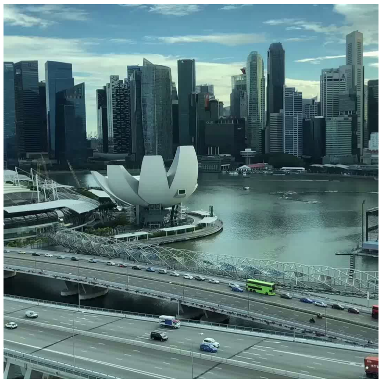

Figure 4: Original video input

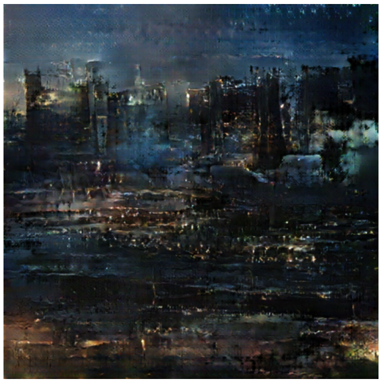

Figure 5: Pix2Pix output

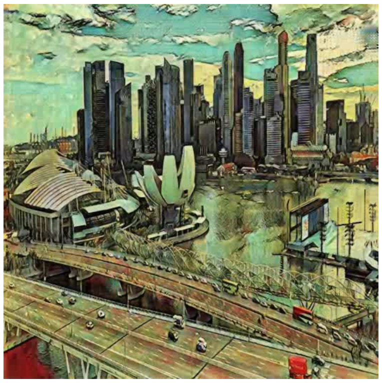

Figure 6: CycleGAN output

We attempted to improve the images in Pix2Pix with a couple of techniques. It should be noted that these methods affect multiple images, so just displaying frames will not show the full impact of these methods unless the video is seen. The first method called Laplacian Blending, involved using a Laplacian-Gaussian blending technique that blends multiple frames together by $L_i = G_i - P(G_{i+1})$ where G_i is the i-th level of a Gaussian pyramid and $P()$ denotes upsampling by a factor of 2. This is done with an overlap size of 50, which denotes that each new frame overlaps the previous one by 50 pixels.

The second method is called Image Denoising, where we take an input frame and neighboring frames and attempt to denoise these frames. This is done by passing the image as well as a block size and window size. Blocks are compared between frames in an image within a window in order to find ideal blocks to denoise the image. Specifically, it computes weights for each pixel based on the color and distance similarity, normalizes the weights so they sum to 1, applies the weights to each color channel, and merges the color channels back into the image. It then does this for neighboring frames and then median blur them as well in order to remove extra noise, and finally converges with the current frame.

The third method used is Histogram Matching, which basically chooses an ideal frame or a template, and applies its intensity values to the source (rest of the images) in the directory. It does this by obtaining pixel values and indices for both the template and source, and then the normalized cumulative distribution function (CDF) is taken for both of these by $CDF(x) = (pixel\ values\ <=\ x)/total\ pixels$ and finally, these values are linearly interpolated and are used to map the pixels from the source to the new image. These methods greatly improved the video result output of Pix2Pix, but for CycleGAN it made small but noticeable changes in the output. Below are some examples of how these methods look in our Pix2Pix and CycleGAN output images.

For these videos, we found that Pix2Pix was able to produce videos with good quality, but required additional image processing techniques to achieve the best results. In contrast, CycleGAN was able to produce high-quality videos without the need for additional processing. We believe this is because CycleGAN is better able to capture the underlying style and characteristics of the original video data.

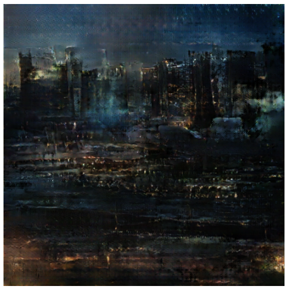

Figure 7: Laplacian

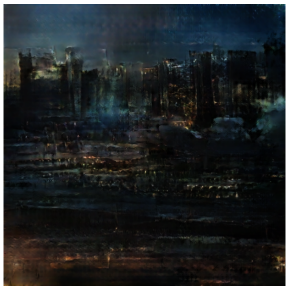

Figure 8: Denoise

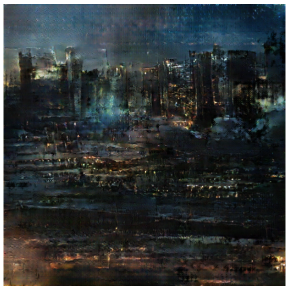

Figure 9: Histogram

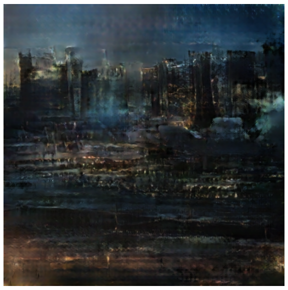

Figure 10: All methods

Overall, our results demonstrate the effectiveness of Pix2Pix and CycleGAN for image alteration and artistic style transfer. We found that CycleGAN outperformed Pix2Pix in both the map and artistic video domains, and that non-ML image processing techniques can be used to achieve a great video result with fewer resources in contrast to using ML methods across whole videos.

4 Challenges and Innovations

Our project faced several challenges that we had to overcome in order to achieve our results. First and foremost, we were not experienced in machine learning, and training the ML models took a long time. This was compounded by the fact that GANs are a newer machine learning technique, and it was difficult to find resources to understand them. Additionally, there were not many small-enough datasets available for our applications, which made it difficult to train our models effectively. We also encountered hardware limitations with our GPU, which prevented us from training on larger datasets, and hence, why we trained on smaller sets. Finally, we made many changes to our project because the outputs of some of the models were not great, so we had to think of alternative ideas multiple times throughout the project. We ended up deciding to demonstrate that we were

One specific innovation of our project was the implementation of the first GAN that we know of that takes in pictures and stitches them into video. This allowed us to generate new video content that maintains the length of frames and consistency of objects while introducing new styles and features. This has implications not only for artistic video transformation but also for real-world applications, such as satellite imaging.

Overall, our project faced several challenges, but we were able to innovate and achieve novel results by utilizing ML and non-ML image processing methods for large images and videos. We expect full points for this section because of how difficult our task was, our satisfactory output despite this, and our innovative approach.

5 Conclusion

In conclusion, our project explored the use of Pix2Pix and CycleGAN for creating satellite-to-map images and artistic style transferring on videos. We trained both models ourselves and utilized longer-trained pretrained models for our final outputs for the art transformations. For the map section, we only utilized pretrained models as these models are the most ideal. Our approach was novel because we maintained these same models, but applied non-ML image processing methods after running these images through the models. This allowed us to achieve results with fewer resources and may have implications for the development of more efficient and cost-effective image processing methods.

Our results showed that CycleGAN outperformed Pix2Pix in both the map and artistic video domains. Even after image processing techniques, CycleGAN was superior and did not require additional processing compared to Pix2Pix. We think this is because of how CycleGAN is structured, as it is better able to capture the underlying style and characteristics of the original data.

Our project faced several challenges, including our lack of experience with machine learning, limited datasets, hardware limitations, and the need to iterate on our project several times. Despite these challenges, we were able to innovate and achieve novel results.

Overall, our project demonstrates the effectiveness of Pix2Pix and CycleGAN for image alteration and artistic style transfer, and the potential of non-ML image processing methods to achieve results with fewer resources. Our results have implications for the development of more efficient and cost-effective image processing methods and for the exploration of new possibilities in the fields of art and computer vision.

References

[1] AsapSCIENCE. The science of motivation. YouTube video, 2019.

[2] Saif Gazali. Cycle gan architecture. Medium article, 2019.

[3] Roger Grosse. Csc321 assignment 4. PDF document, 2018.

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2017.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018.

[7] Aladdin Persson. Pix2pix implementation from sratch. YouTube video, 2021.

[8] Pexels. City water videos.

[9] Y. Xu and W. Hwu. Computational photograph: Video. University of Illinois at Urbana-Champaign.

[10] Jun-Yan Zhu. Pytorch implementation of pix2pix, 2017.

[11] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.