Image & Video Processing, ECE 588:

Domain-Cycling: An Exploration into Unpaired Image-To-Image Domain Mapping and its Effects on Various Image-Processing Techniques

By: Ryleigh Byrne, Maddie Demming, Wyatt Focht, Alok Malhotra, Paul Truitt

Background & Motivation

Significant prior research has been done in the image-processing field of "style-transfer." Most popularly, style-transfer has been utilized to transform the style in which various pieces of artwork are drawn into a completely different style by that of a different author. This can be extended to the act of isolating key objects in images and videos and transforming such objects from their original domain-style to an entirely new domain (ex. zebras transformed into horses). We will use a generalized approach, **Cycle-Consistent Adversarial Networks** (CCANs), to perform unpaired image-to-image domain mapping.

The CCAN we chose to use specifically is CycleGAN. CycleGAN is a generative adversarial network (GAN). GANs typically consist of two submodels. The first is a generative model that is perturbing the input image, and the second is a discriminator model that gives feedback on whether perturbations have been convincing. CycleGAN specifically contains two generative and two discriminator models for each pair of domains (one for zebra to horse and one for horse to zebra).

The objective function of CycleGAN has two parts. The first is termed Adversarial Loss. In this case, the Discriminator Models and Generative Models work against one another. The Generative Model tries to minimize the adversarial loss, while the Discriminator Model tries to minimize it. This Equation implies that we are training for a discriminator that will only be fooled if the generated image is hard to distinguish from the other domain's training image.

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \ & \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] \\ & + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))]\end{aligned}$$

The next portion of the objective is called the Cycle Consistency Loss. The purpose of this loss is to further encourage the generated images of one domain to be similar to those of the Training Data of the other domain. To do this, we force the input images through a 'cycle' of generative models (transformation from one domain to another, then back to the initial domain). From there, the difference between the input image and the twice transformed is measured:

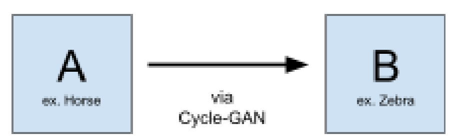
$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = \ & \mathbb{E}_{x \sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].\end{aligned}$$

Note that here the L1 norm of the difference between the images is used to encourage as many pixels as possible to remain unchanged. Both generative models then work to minimize this loss.

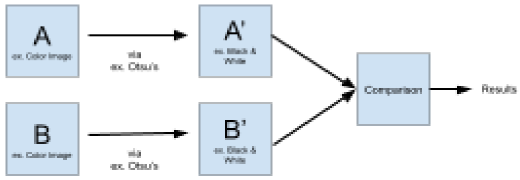
The resulting overall loss function is shown below. Both of the Generative models work to minimize this loss, while the Discriminators work to maximize it.

$$\begin{aligned}\mathcal{L}(G, F, D_X, D_Y) = \ & \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ & + \lambda \mathcal{L}_{\text{cyc}}(G, F),\end{aligned}$$

Utilizing this CycleGAN deep-learning model, we seek to process various example images whose output following input to the deep-learning model contain features that were mapped from an original domain to a new domain. For the purposes of discussion, we will refer to untransformed images as Class A images and transformed images as Class B images. Examples of the output of the CycleGAN model can be seen in the appendix.



Following this, we seek to compare the performance of various image-processing techniques upon the Class A images as well as their corresponding Class B images. For the purposes of discussion, we will refer to the post image-processing-technique Class A images as Class A' and similarly refer to the post image-processing-technique Class B images as Class B'. Finally, the Class A' images will be compared to their corresponding Class B' images in quantitative or qualitative ways, depending on the image-processing technique utilized (where A' serves as the "ground-truth" for B').



The impacts on this type of model are very significant. They allow anyone who can access the internet to easily perform these transformations for themselves by simply using CycleGAN's Colab notebook. Ideally, the output of this model would retain all of the same capabilities held by the input image. If this is truly the case, any photographer or videographer would be able to essentially change their entire environment from the perspective of a viewer, or even a computer reading in the image. This made us wonder to what extent this was actually the case. Are the outputs of CycleGAN capable of further image processing techniques?

We decided to use the pretrained models provided in the CycleGAN Colab Notebook (linked in the Aknowledgements). This allowed to simulate being a non technical person trying to use CycleGAN. The models that they use are designed for 256x256 images, so all generated outputs will then be cropped to squares.

The various image processing techniques explored in this experimentation are Yolo, Histogram Matching, the Hough Transform, and Otsu's Method for Segmentation.

Methods Utilized

YOLO

Object detection is one of the most prominent areas of research in the field of image-processing. YOLO, or You-Only-Look-Once, is a convolutional neural network that is frequently utilized to solve this task. Most simply, YOLO is able to utilize inferences learned from an extremely large dataset to not only identify any objects in an image that fall into one of almost 80 learned-classes, but also to draw relevant bounding boxes around the feature(s) in question.

We sought to experiment with the effects of CycleGAN on the inputs to YOLO, as many of the possible transformation-domains that CycleGAN is capable of performing are classes that YOLO is also capable of recognizing. For example, CycleGAN has the ability to transform an image of a horse into an image of a zebra. Assuming a high quality of both CycleGAN and of YOLO, we would then expect YOLO to identify a zebra in the newly-generated image where it had identified a horse in the original one. Based on this experimentation, we would then be evaluating the effectiveness of CycleGAN on its main function of domain-mapping.

Histogram Matching

Histogram matching is an image processing technique which serves to transform an input (source) image by updating its pixel intensities to match the distribution of a reference image's pixel intensities. It is important to note that the source image contents do not change, only the distribution. We can then expect features like illumination and contrast to change, allowing for more aesthetic photographs as well as color corrected and consistent photographs. In order to match the histograms of images A and B, we need to first equalize the histogram of both images. Then, we need to map each pixel of A to B using the equalized histograms. Then we modify each pixel of A based on B.

We chose to experiment with the effects of Histogram matching on images transformed using CycleGAN and comparing the histogram-matched original image and histogram-matched transformed image. Note that the original and transformed images utilize identical reference images. We can observe the effectiveness of CycleGAN on histogram matching by comparing the histograms of each color channel for raw images and their transformed counterparts. The results of our experimentation will allow us to determine if a significant relationship between CycleGAN and histogram matching exists.

Circle Hough Transform (CHT)

The circle Hough Transform is a powerful feature extraction technique for detecting circles in images. It is derived from the Hough Transform and a form of model based segmentation. The CHT is nearly identical to the Hough Transform for lines, but uses a parametric form for a circle. Circles are detected by summing "votes" in the Hough space and selecting local maxima in an accumulator. The CHT can be used to determine the parameters of a circle when a number of points that fall on the perimeter of the circle are known. It is one of the best techniques for known shape detection (i.e., lines, circles, etc.), but can be computationally expensive.

We decided to experiment with the effectiveness of the circle Hough Transform on images transformed using CycleGAN. By comparing circle detection using the circle Hough Transform in raw images and their transformed CycleGAN counterparts, we can observe the impact CycleGAN may have on important image-processing techniques, like the circle Hough Transform. The results of our experimentation may indicate a relationship between the CycleGAN and the effectiveness of feature extraction on an image.

Otsu's Method

The Otsu method is an image thresholding technique that binarizes images based on pixel intensities. In other words, a binary image is created using a particular threshold value. When the intensity of a pixel is greater than the threshold, the output pixel is white. And, when the intensity of a pixel is less than or equal to the threshold, the output pixel is black. The difference of Otsu's method from other image thresholding techniques is that Otsu automatically determines a threshold based on calculations, while with other techniques a threshold is manually given (much more tedious). Therefore, Otsu's method is known as automatic global thresholding since a single threshold is automatically generated and used for an entire image.

Otsu's method determines a threshold automatically with Otsu's method with the following steps: processing an input image, obtaining an image histogram, computing a threshold value based on the histogram, and replacing the image pixels based on their intensity and the threshold. In more detail, Otsu's method processes an image histogram and segments the objects by minimization of variance on each of the classes. The image histogram displays two clear peaks that represent different ranges of intensity values. The threshold is the result of the minimization of the weighted variance of the classes.

We are experimenting with the effects of Otsu's method on images transformed using CycleGAN and comparing the Otsu applied original images to the Otsu applied transformed images. We can see the effectiveness of CycleGAN on this technique by calculating the mean square error (MSE) and Structural Similarity Index (SSIM) values between the two Otsu applied images. For this experiment, we can see if there is a relationship between CycleGAN and automatic global thresholding with images.

Results

YOLO

We seek to evaluate the effectiveness of the CycleGAN-transforms based on the relative accuracy that YOLO is able to achieve on un-transformed images versus transformed images. A variety of transformations with sample images were tested in this process (horse -> zebra, zebra->horse, apple->orange, orange->apple).

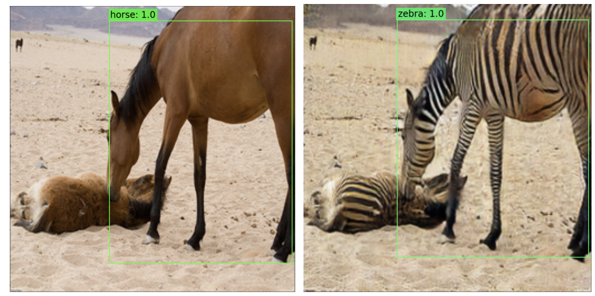


Figure 1.0: YOLO on Raw Image (left) vs YOLO on CycleGAN Transformed Image (right)

Overall, YOLO was able to classify and identify objects with a **65.02%** accuracy utilizing the original images. After transforming such images utilizing CycleGAN, a **29.92%** accuracy was achieved. As can be observed, a severe drop in accuracy occurred after the introduction of CycleGAN (**35.10%**). Though our base accuracy was not high to begin with, the relative drop in accuracy (almost half) indicates a lack of robustness in the ability of CycleGAN to accurately transform objects between relative domains.

However, it was qualitatively observed that for raw images in which the feature of interest was not particularly clear or distinct to begin with, CycleGAN produced results that were much more difficult for YOLO to identify; this can likely be attributed to the fact that the less-clear raw images were less successful at domain-transfer than clearer, crisper images.

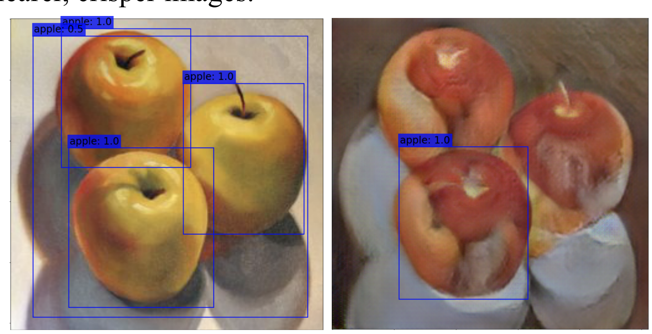


Figure 1.1: Example of Significant Decrease in Clarity from Apple to Orange Transform

Histogram Matching

We aimed to quantitatively and qualitatively evaluate the differences between original histogram-matched images and CycleGAN transformed histogram-matched images. To demonstrate our results, we've highlighted many original images and their transformed counterparts below (horse to zebra, zebra to horse, apple to orange).

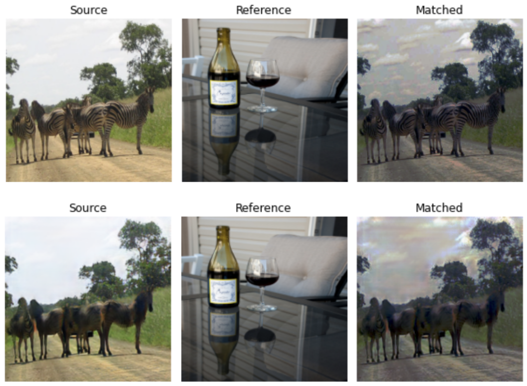


Figure 1.2: Histogram Matching Results for Zebra (Top, Orig) and Horse (Bottom, Trans)

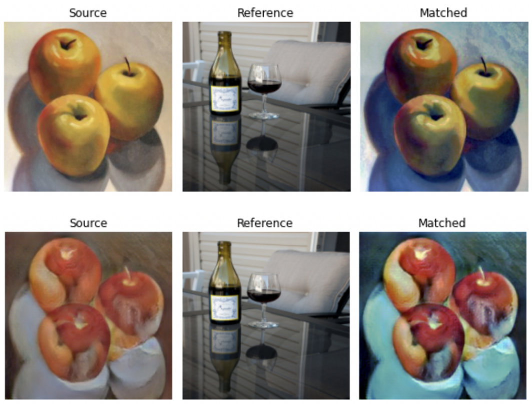


Figure 1.3: Histogram Matching Results for Apple (Top, Orig) and Orange (Bottom, Trans)

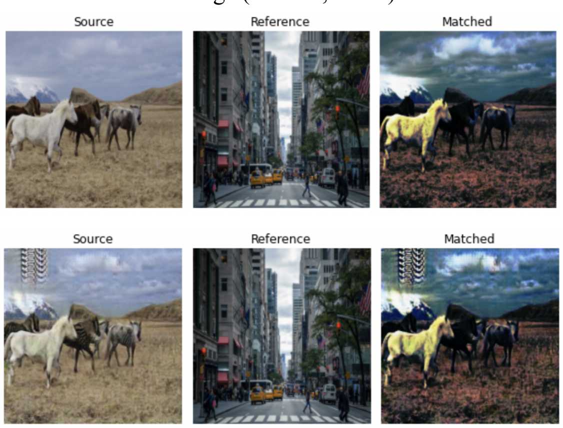


Figure 1.4: Histogram Matching Results for Horse (Top, Orig) and Zebra (Bottom, Trans)

From the above figures, it is evident from figure 1.2 that the original image and transformed image display an extremely similar matched result. Comparing the histograms for each color channel confirmed this initial result. Furthermore, it is evident from figures 1.3 and 1.4 that the matched results display lower similarity.

A mean-squared error calculation between the matched original image and the matched transformed image was performed to compare the similarities. Zebra to horse exhibited an MSE of 94.53, horse to zebra exhibited an MSE of 113.25, and apple to orange exhibited an MSE of 148.05.

From the above results, we concluded that the MSE between matched original and matched transformed images increased with respect to the qualitative assessment of the CycleGAN transformation. Poor transformations exhibited higher MSEs and good transformations exhibited lower MSEs. These results are consistent with what was expected and can also be attributed to the less-clear (i.e. artwork instead of photography) raw images being harder to transfer.

Circle Hough Transform

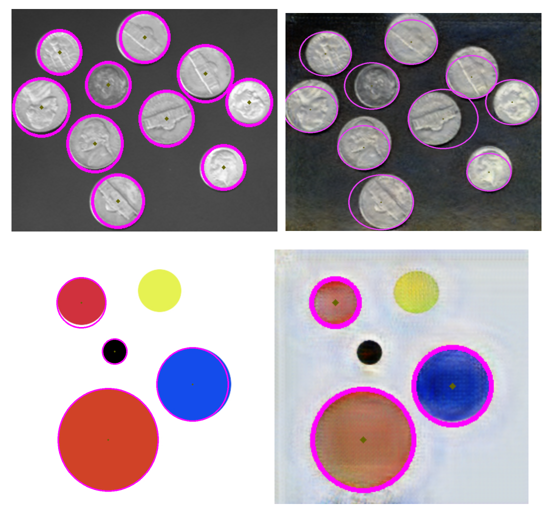


Figure 1.5: CHT on Raw Image (left) vs CHT on CycleGAN Transformed Image (right)

As seen from Figure 1.5, the circle Hough Transform was extremely effective in detecting circles on raw images. Nearly every circle is detected in the raw images, and appropriately positioned. The CHT was less effective in detecting circles on the CycleGAN transformed images. As observed in the transformed image of the coins, the CHT detected the correct number of circles, but the positioning of these circles is not as accurate as the raw image. In the transformed image of the painted circles, the CHT detected less circles, compared to the raw image of the painted circles.

These observations could imply that using CycleGAN to transform images has an impact on the performance of feature detection techniques, specifically the circle Hough Transform. Although CycleGAN successfully transforms the style of an image, it may change important qualities that degrade the performance of model based segmentation techniques, like the circle Hough Transform. More experiments using various segmentation techniques could indicate a stronger relationship between CycleGAN and the effectiveness of feature detection techniques.

Otsu's Method

In order to analyze Otsu's method with CycleGAN, we applied Otsu to original images and CycleGAN transformed images and compared their differences. We used this technique for multiple sets of images, such as apples to oranges, horses to zebras, and photos to Monet artwork. In order to calculate the differences between the images, we use MSE (mean squared error) and SSIM (structural similar index) between the original image with Otsu and the transformed image with Otsu. The higher the MSE value, the more error and therefore the more differences between the two images. The higher the SSIM value, the higher the similarity and therefore the less differences between the two images.

Analyzing the impact of CycleGAN with the different categories of images, we can see that each set of images have different effects when Otsu is applied. For example, converting "apples" (original image) to "apples to oranges" (CycleGAN image) has a higher amount of error and lower structural similarity than "oranges" (original image) to "oranges to apples" (CycleGAN image). In addition, converting "horses" (original image) to "horses to zebras" (CycleGAN image) has a lower amount of error and higher structural similarity than "zebras" (original image) "zebras to horses" (CycleGAN image). References of two types of CycleGAN transformations with Otsu's method are shown below in Figure 1.6.

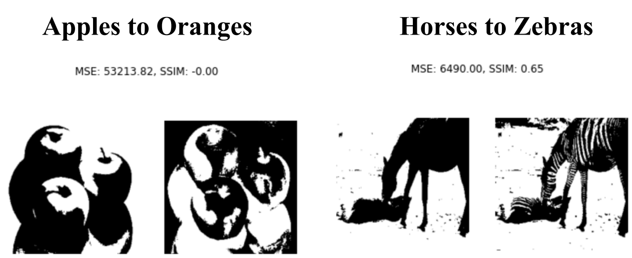


Figure 1.6: Otsu's Method on Zebra to Horse and Horse to Zebra

Discussion

Current uses of style transfer networks are elementary; online platforms that support these transformations are low stakes and entertainment-based. However, as Cycle-Consistent Adversarial Networks improve in both accuracy and speed, more useful applications have emerged. One such application is short film production to avoid the hardships associated with complex special effects. For example, an American student interested in creating a film involving African wildlife could utilize local farm horses to depict a dazzle of zebras as a substitute for an expensive trip or editing software. Producers can then also apply other image processing techniques if they desire.

The produced CycleGAN images exhibited poor consistency, suggesting a necessity for improvements in the algorithm and network prior to its utilization as special effects. This is evidenced by significant performance increases in our tested techniques; these increases occurred only for images with extremely believable CycleGAN transformations. Looking at the results of this model, the object to object transforms performed significantly worse when the tested images were significantly different from the training data. Therefore, a potential user of this editing technique will need to carefully obtain a training set that is similar to their desired transformation outputs (i.e. similar lighting, single vs. multiple objects).

Another interesting result was found in the image sizing. Images not correctly fitted to the model were distorted and directly impacted the model's ability to produce a believable transform. It is of paramount importance that the network be designed for images of the exact size of those that will be used. This will reduce the distortion of shapes that cause issues in shape dependent transforms like Hough Transform.

Overall, if someone is hoping to use a Cycle-Consistent Adversarial Network for an application where quality is important, they must be mindful of the Architecture and Training process of networks they consider. It is important to find one that is trained to perform on similar images, and designed to handle similar images. This will allow for minimum distortion on the transformed images resulting in increased flexibility of post transformation operations.

Acknowledgements

- Otsu, Nobuyuki. "A Threshold Selection Method from Gray-Level Histograms." *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, 1979, pp. 62–66., https://doi.org/10.1109/tsmc.1979.4310076.
- Zhu, Jun-Yan, et al. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks." *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, https://doi.org/10.1109/iccv.2017.244.
- Duda, Richard O., and Peter E. Hart. "Use of the Hough Transformation to Detect Lines and Curves in Pictures." *Communications of the ACM*, vol. 15, no. 1, 1972, pp. 11–15., https://doi.org/10.1145/361237.361242.
- Zhu, Jun-Yan, et al. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks." *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, https://doi.org/10.1109/iccv.2017.244.
- CycleGAN Codebase
- CycleGAN Colab
- YOLO Codebase
- Histogram Matching
- Otsu's Method
- Image Comparison

Appendix

Apple2Orange:

Orange2Apple:

Horse2Zebra:

Zebra2Horse:

Photo2Monet: