CycleGAN with Shape Color Regularization

Jingyu Liu   Ruizhen Mai

Abstract

Image-to-image translation is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs. CycleGAN[1] provides an approach to accomplish this task despite having only unpaired image sets. However, the objective of CycleGAN does not include enough information and have loose assumptions on the datasets so that the probability modeled by the generators can end up being less meaningful even though the loss function is optimized. We propose a method to constrain the probability space of the generators such that more useful information in determining the mapping functions can be extracted from the unpaired datasets. Our approach is to add a new regularization term called shape-color-regularization (SCR) in hope that when the source image X has very similar shape with the target image Y, the similar pixel areas between the generated image and the target image will have very close color distributions. Some experiments have shown that our approach has the potential of improving the original CycleGAN on datasets with more similar paired images and will generally not perform worse.

1. Introduction

From both the results in the CycleGAN paper and our own experiments, CycleGAN performs very differently across different datasets and even training. When the datasets contain images mostly about landscape or paintings, CycleGAN can produce superior results and the transformation is realistic and obvious. When trained on some datasets that contain images about objects, however, CycleGAN performs poorly and the quality of the generated images is not every consistent. We presume the cause of this problem to be that natural relationships from unpaired images can be hard to extract. For example, in the dog to cat transformation example, the task of mapping a dog to a cat can be hard if the generators can not understand whether we want to transform the whole image or just the objects in the pictures. If the two datasets contain similar backgrounds but different cats and dogs, it might be possible that the generators would think their job to be transforming the background instead of the foreground objects, which will in most cases be problematic and not what we intend to achieve. This problem becomes less serious if our job is to transform the whole image to another style, so we will focus on improving the quality of our model trained with the datasets containing images with objects.

2. Vanilla CycleGAN

2.1. CycleGAN objective

The objective of the CycleGAN discussed in the original paper is to train two generative neural networks, one for each direction of the mapping. The loss function contains two GAN losses together with their famous cycle consistent loss to ensure that when the generated image from X gets transformed back, the distance between this and the original image X should be as close as possible. The loss function will be:

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

2.2. Potential Weakness

Unfortunately, none of the loss terms in the vanilla CycleGAN specified what the generator should focus on during the training, which causes the problem we mentioned earlier. This is also a drawback of the generative adversarial networks which only cares about distinguishing between fake and real images without human visual aesthetics taken into consideration. So we will add a new loss term into the overall objective function and this loss term, or regularizer, will attempt to fix this problem by limiting the ability of the generators to change the background and encouraging them to focus mapping the useful information only.

3. Shape Color Regularization (SCR)

The definition we came up with for a meaningful and visually realistic image-to-image translation is that the gener-

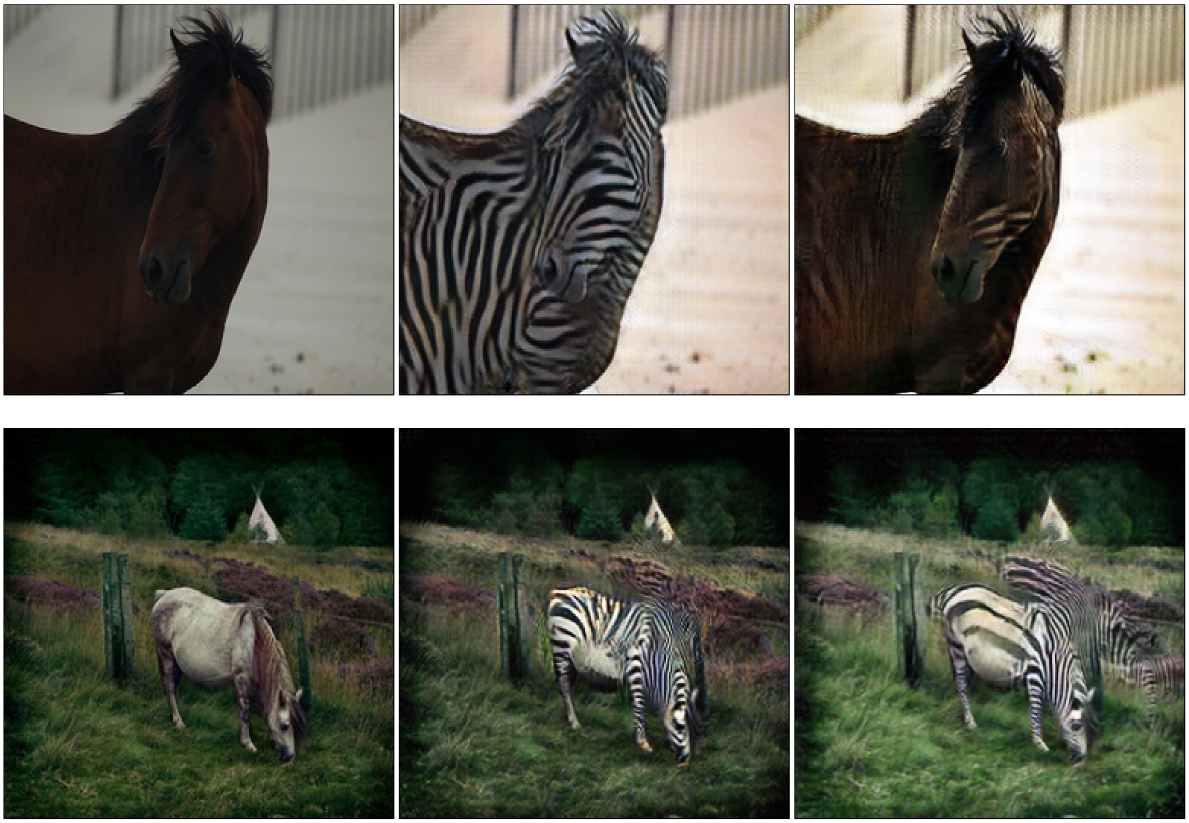
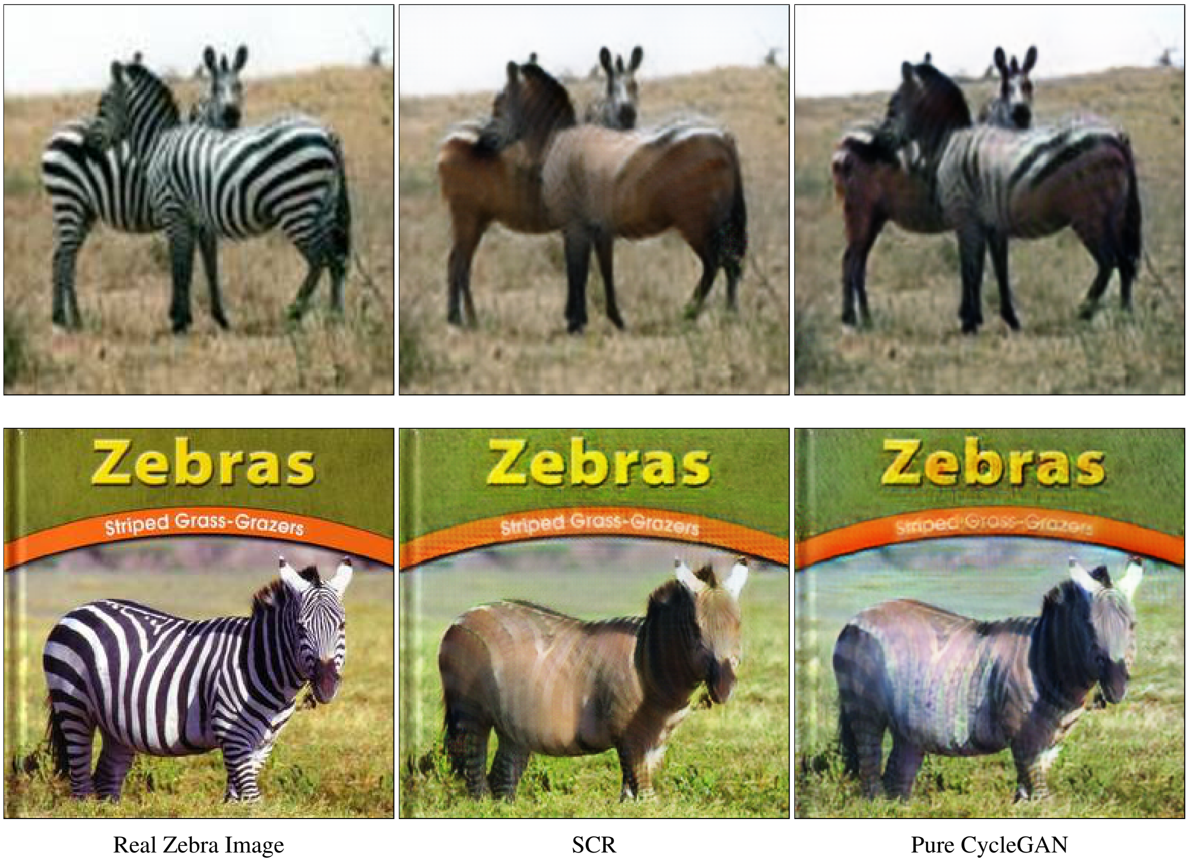


Figure 1. Some examples of mapping horses to zebras where the model with SCR achieved better results than without it. The left column contains ground true horse images. The middle column contains the generated images from the model with SCR (60+25). The right column contains the generated images from the pure pretrained CycleGAN model after 200 epochs. The quality of the middle and right columns are similar.

ators will focus on changing the foreground and leave the background unchanged. Our method is based on the assumption that the datasets have some pairs of images of similar shapes despite being unpaired. In this case, more information about the relationship between these two image sets can be extracted from those images of similar foreground shapes. We propose a regularization term to constrain our generators. Since the image datasets are unpaired, the more images of similar foreground shapes these two image datasets have, the better this regularization term tend to work in restricting the conditional probability modeled by the generators. The generators are not influenced too much if the assumption is not held and the result will be at least as good as the model without the regularization term. Shape-color-regularization (SCR) is a loss term added to the overall generator loss that penalizes if the target image (Y) and the generated image $G(X)$ have similar foreground shapes but the color distributions of the foregrounds are very different, or the original image X and generated image $G(X)$ have similar background shapes but the color distribution of backgrounds are very different. This loss applies to both direction and the meaning of colors being different and shapes being similar can be defined in various ways as long as the similarity is defined by a scalar value. So ideally, both generators G and F will try to make the background change as little as possible while still striving to change the foreground to finish the translation task. The shape and color similarity scores are used to construct the regularization term and the details on how to get them out of the original images are discussed in section 4 and 5. Future works. Usually we assume the foreground shape similarity is the same as the background shape similarity for the sake of both computational efficiency and ease of implementation. The SCR loss is given as:

$$\begin{aligned}\mathcal{L}_{SCR} &= ShapeSim \cdot [\alpha f(G(X), Y) + \beta b(X, G(X))]^{-1} \\ &+ ShapeSim \cdot [\alpha f(F(Y), X) + \beta b(Y, F(Y))]^{-1}\end{aligned}$$

Where α, and β are hyperparameters and f and b are foreground and background color similarity functions that takes in two images and outputs a scalar. At any given epoch and batch during the stochastic gradient descent with batch size



Real Zebra Image SCR Pure CycleGAN

Figure 2. Some examples of mapping zebras to horses where the model with SCR achieved better results than without it.

1, X and Y are images from two datasets, if they have similar foreground shapes, they will have higher shape similarity. If the foreground color distribution similarity between $G(X)$ and Y is not high enough, then the loss will be greater. If the background color similarity between X and $G(X)$ is not high enough, then the loss will again be greater. If however, the shape similarity is pretty low which covers the case where the image datasets are completely unpaired, then the loss term will be really close to 0 and have very small impact on the overall loss. And this applies for the other direction as well.The full loss of CycleGAN with SCR is then defined as:

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

4. Training Details

Our experiments were conducted on Google AI Platform with 4 CPUs and Tesla V100 and P100 GPUs. We experimented with both our own implementation of CycleGAN with SCR, and the modified original source code provided by the author. The latter achieved better results which are used in our analysis. We obtained the pretrained model (trained without our loss) and then use the weights as the starting points to train with our new regularization term.

4.1. Log Smoothing

Our color and shape similarity are all scalar floating point values between 0 and 1. And when we divide the shape similarity by the color similarity, the result will fall in the range between 0 and positive infinity. In order to avoid floating point overflow that happens when the color similarity is extremely small, we add a minimum threshold to the denominator. Then to smooth the loss and make its change less dramatic, we applied a natural logarithm to the ratio plus 1, which is $log(ratio + 1)$. In our experiments, the raw ratio will usually be less than 400, so the smoothed ratio and its rate of change will be in reasonable range.

4.2. Half and Half Splitting

During the first couple of epochs, the generators are not trained to be mature enough to generate visually realistic

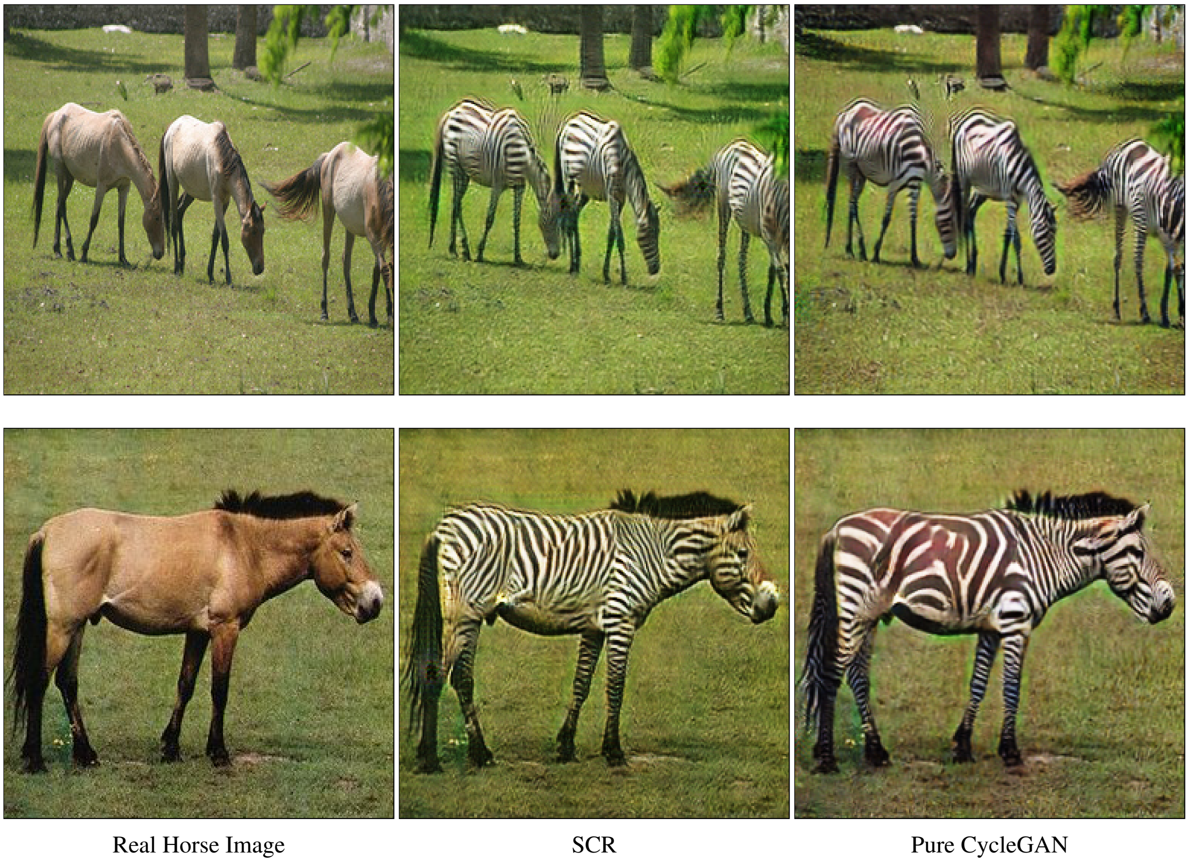


Figure 3. Some comparable horse-to-zebra results from our model to the pure CycleGAN

images. Therefore, sending these images with non recognizable objects to the mask extractor will usually result in finding no masks at all. This procedure will therefore not contribute a lot to the parameter learning of the generators. In order to make the training faster, we only add our color shape regularization term to the generator loss after half of whole training process is finished.

4.3. Mask Extraction

To obtain the masks, we first tried to use a pre-trained Mask-RCNN model implemented with Tensorflow. The output of Mask-RCNN is ideal, which contains the detection score, the category boxes and the associated silhouette masks for all recognizable objects in the image. But we only achieved 7 seconds per image inference speed on our testing machine which is too slow to iterate the experiments with. We then switched to another model called Single Shot MultiBox Detector(SSD), which only outputs a series of detection boxes but can run relatively quickly. We heuristically chose the one with the maximum masking area to be our foreground mask.

4.4. Shape Comparison Method

Our goal is to only compare the shape similarity and leave out the position differences. After obtaining the detection masks (detection boxes) from 4.3, we shift them to the upper left. Then we count the overlapping pixels and divide it by the maximum number of pixels of these two masks, the value of which is therefore clapped between 0 and 1. This method is simply and computationally cheap, but we can apply other more sophisticated methods for this step (more details in future work section).

4.5. Color Comparison Method

We extract color distribution histograms from source images and compute the color similarity value from these histograms. For each batch data including one real image from each of the two image sets, we will feed them along with their object detection masks (obtained from 4.4) to the function $calcHist$ provided by OpenCV to compute their color histograms. Depending on whether we want foreground or background color similarity, we only consider color values from either the foreground or the background, the boundary of which is defined by the masks provided in the above step. These two histograms will be used to find the color similarity via OpenCV function $compareHist$. The reason we chose correlation method from various available metrics is because its value is positively proportional to the visual color similarity and resides in the range from 0 to 1.

5. Experiment Results

We focused mainly on zebra2horse dataset because of limited time for experiments. Since we are utilizing the method of half and half splitting, we will train our model with SCR based on a pretrained model without SCR. We will give abbreviations to different experiment configurations. X+Y means we train our model with SCR for Y epochs based on the pretrained model without SCR at checkpoint X epochs. Several attempts were made: 60+25, 60+40, 60+55. We will compare the results with a pretrained models after 200 epochs provided by the author of CycleGAN. Despite testing on only one dataset, SCR does show some generalized potential in constraining our generators to focus on foreground shape and color similarities. Besides the cross differences between two models, there are some other interesting findings when the number of epochs increases for the model with SCR.

5.1. Comparison with vanilla CycleGAN

In Figure 3&4, we show that some images generated by our 60+25 SCR model have comparable quality to those generated from the pretrained models after 200 epochs. The mappings are from both directions. In Figure 1&2 we show some particular examples where our generated images have much higher quality then ones from the vanilla CycleGAN.

Besides, we found that training with fewer epochs seems to alleviate the problem of overtraining, which happens when training with too many epochs, some of the generated images become noisy, some of them start to contain shapes that mislead the detector to consider non-horse areas as horses, and some of them even looked less visually realistic. The problem might be caused by the unbalanced weight given to different loss terms and the identity loss tend to eclipse the GAN loss as the number of epochs increases. If we were able to take advantages of SCR to make the loss converge faster, then it is possible to alleviate overtraining.

5.2. Failing Examples

During training on the horse2zebra dataset, we found two extreme cases. On the one hand, our model produced black-white inverted images without the identity loss.

One hypothesis is that the without the identity constraint, the generator is trained to the exactly opposite in the probability space. Even though the generated images are not realistic visually, the loss function is optimized numerically. So there is a discrepancy between visual aesthetics and loss function representations. Adding the identity loss will direct our generators to the right direction as it provides some information that is hard to obtained from the unpaired image datasets.

On the other hand, adding an identity loss sometimes force the generator to become an identity function and the generated images merely changed from the input. We suspect that it is because the structure of our generators could capture the wrong features and the GAN itself is unstable and has high training variance.

6. Conclusion

As we have seen in the experiments, the new model with our regularizer achieved some interesting results and exhibit the tendency of constraining our generators. The difference between adding our SCR or not indicates the positive affect carried with it. By adding this regularizer to the loss function with appropriate hyperparameters, we can achieve relatively the same quality with fewer epochs as the pure CycleGAN and will perform better when the dataset contains images with objects.

7. Future Works

7.1. Different Object Detection Method

During the experiments, our object detection method is over-simplified. Having more complicated methods like silhouette extraction or shape decomposition might achieve better results.

7.2. More Robust Shape comparison Method

Our approach of comparing masks is to shift and count overlapping pixels but this is problematic and can find trivial counterexamples to prove its incorrectness. Other more interesting and robust methods include shifting to the weight center, or doing shape reconstruction. The shape comparator can be modeled using neural networks with CNN layers.

7.3. Different Color Comparison Method

Color comparison based on the distributions might be too high level to capture the important details, and adding some extra information about the texture or details might provide more substantial changes but at the same time might cause unnecessary noises, which leads to oscillating loss and overfitting. The comparison metric we chose in the experiment was based on the correlation, and other metrics like distance functions, Hellinger or Chi-square metrics might produce slightly different results.

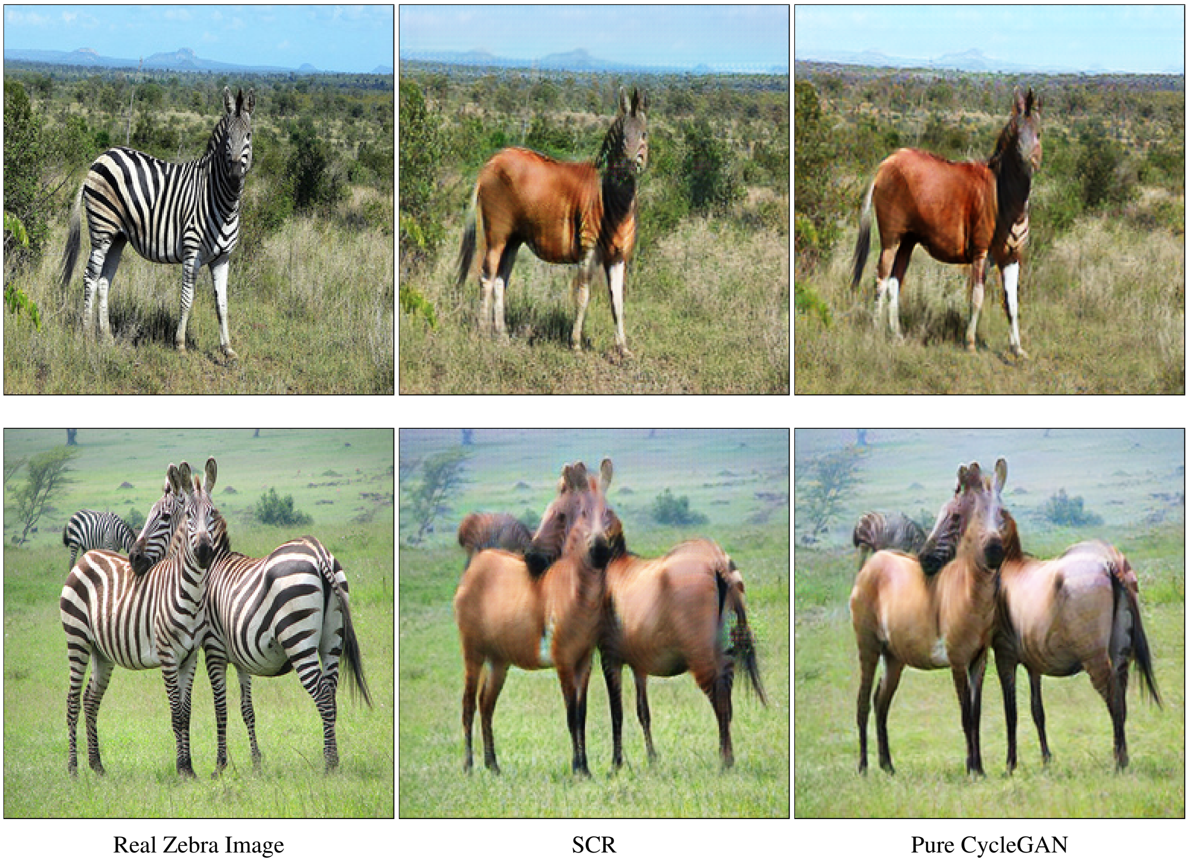


Figure 4. Another set of comparable results, from zebra to horses.

Acknowledgements

We specially thank Sudarshini Tyagi for suggesting us to work on CycleGAN, Prof. Rajesh Ranganath for giving us the clear idea, and Xintian Han and Aodong Li for the discussion and help on the coding part of the project.

References

1. Jun-Yan Zhu, et. al., *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, arXiv preprint:1703.10593

2. Wei Liu, et.al, *SSD: Single Shot MultiBox Detector*, arXiv preprint: 1512.02325