Deep Learning in Computer Vision - Project 3

Till Ariel Aczél (s203216), Jan Piotr Latko (s193223), Jonas Søbro Christophersen (s153232), Technical University of Denmark, Deep Learning in Computer Vision (02514), 22/06/2021, Github Repository

Introduction

The purpose of this project is to implement a network using the Cycle-Gan architecture to convincingly convert images of zebras into horses (Z2H), and vice versa (H2Z).

Data

The data consist of 1187 images of horses and 1474 images of zebras. The data set is split into training/testing with the following split: 2401 / 260. All images are scaled to a 128-by-128 pixel resolution, as the Cycle-Gan architecture is quite expensive to train.

Figure 1: .Sample data showing different pictures.

Network

The Cycle-Gan network consists of two connected General Adversarial Networks (GANs) based on residual blocks (for details see implementation). The individual GANs aim is to convincingly convert images of one class to the other, which is then checked by the discriminators. A cycle-process of the network is exemplified in Figure 2 below, where it can be seen that an inputted image of a horse is converted to a fake zebra through the generator (G_{H2Z}), which is then evaluated by a discriminator (D_Z). The fake image is fed through another generator (G_{Z2H}), with the purpose of reconstructing the original class. Analogically zebras are converted to horses (G_{Z2H}), classified (D_H) and cycled back (G_{H2Z}). Additionally an identity reconstruction is performed by passing horses and zebras through G_{Z2H} and G_{H2Z} respectively.

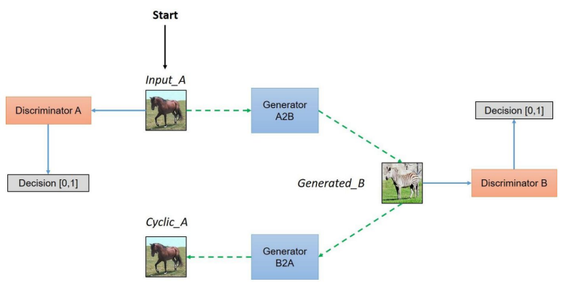

Figure 2: Cycle-GAN Architecture

Training

We train D_H and D_Z by minimizing the MSE (with target of 1 for real and 0 for fake images).
The generators are optimized to maximize the discriminator MSE and minimize the L1 reconstruction loss for both the identity and the cycle reconstructions. The loss terms are weighted (as specified in Table 1).

An array of hyperparameters were tried in an experimental fashion. The hyperparameter-setting for the results in this presentation are presented in Table 1 below.

Weight Identity	10
Weight Cycle	3
Batch Size	8
Optimizer (G&D)	Adam
Learning Rate (G&D)	0.001

Table 1: Hyperparameters used.

Additionally, when training discriminators both real and generated images were augmented by adding a random amount of color-shifts, translation and cutouts to the images.
When training D, instead of exclusively feeding it the most recently generated images, older images were taken at random, from a pool consisting of the 50 most recently faked images. This was done to improve the stability of the adversarial training, as also presented in [1].

Results

The model is evaluated by using the Frechlet Inception Distance (FID) score which evaluated the resemblance between distribution of real image, and generated images. The FID is defined as:

$$\|\boldsymbol{m} - \boldsymbol{m}_w\|_2^2 + \mathrm{Tr}\left(\boldsymbol{C} + \boldsymbol{C}_w - 2\left(\boldsymbol{C}\boldsymbol{C}_w\right)^{1/2}\right)$$

where means ($\boldsymbol{m}$) and covariances ($\boldsymbol{C}$) are calculated based on the representations extracted from the real and generated images using Inception V3 model. The FID-scores for the best performing network is seen in Table 2.
However, the main purpose of the project is to (convincingly) generate real-looking horses and zebras and thus we inspect our results qualitatively by visualising data samples in Figure 3.

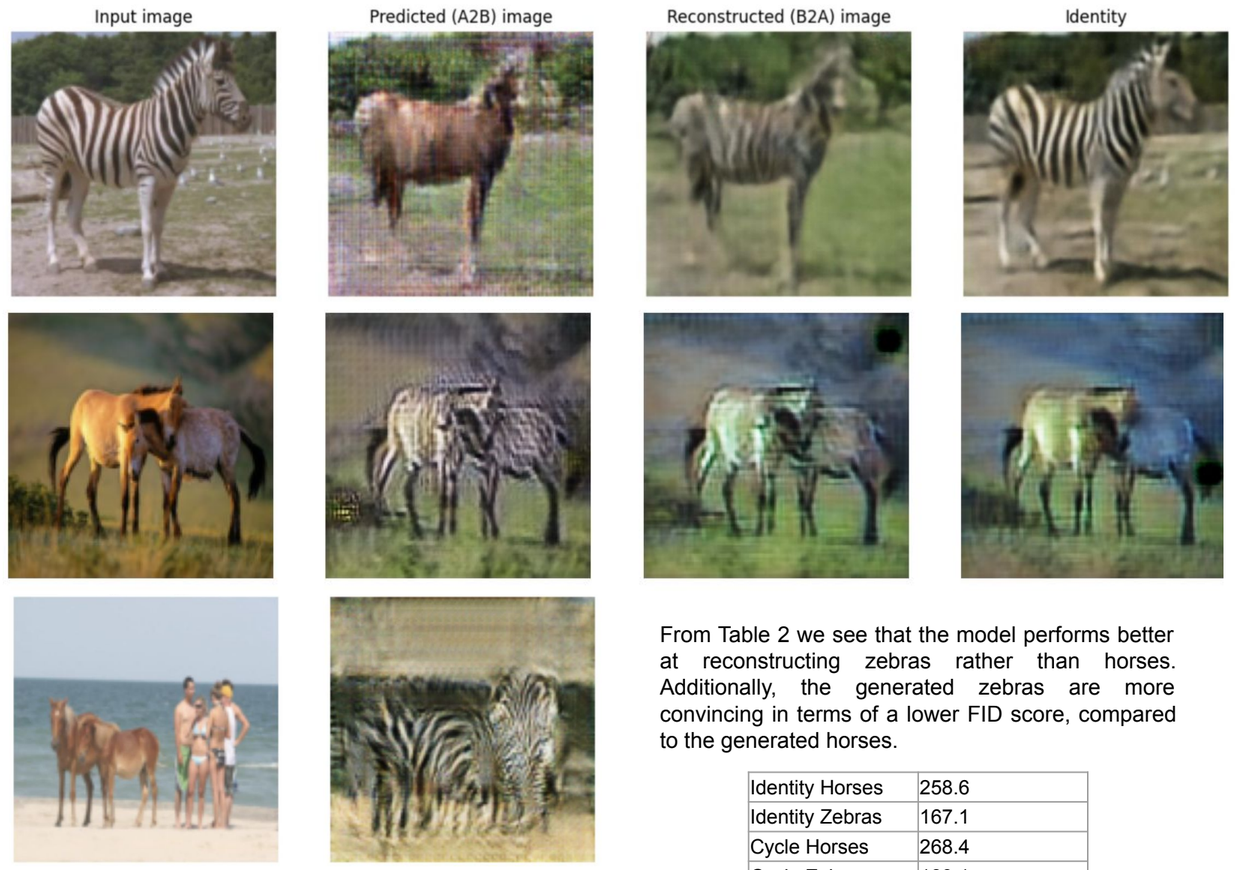

Figure 3: Converted images and reconstructions.

From the produced images it is seen that the network is not capable of converting the images convincingly, however, the network is capable of (to some extent) removing stripes from zebras, while colouring the same regions with an appropriate color (usually brown). For input images of horses the black and white stripes are added, although not always in the correct positions (sometimes everything just turns into zebra).
All generated images are subject to extensive distortions, both in color and resolution. This could be due to many things, and would likely be improved by adding more data or further augmenting the already existing data. Longer training could also alleviate this. Additionally the hyperparameters used to generate these images were not optimized - due to time constraints. The hyperparameters were selected based on what worked for others in similar situations, and as such, may not be ideal for this particular instance. We expect that tuning the hyperparameters would help maintain balance between G and D and optimally train both, resulting in better performance.

From Table 2 we see that the model performs better at reconstructing zebras rather than horses. Additionally, the generated zebras are more convincing in terms of a lower FID score, compared to the generated horses.

Identity Horses	258.6
Identity Zebras	167.1
Cycle Horses	268.4
Cycle Zebras	189.1
Horse to Zebra	179.7
Zebra to Horse	292

Table 2: FID Scores for the different functionalities of the Cycle-GAN network.

Discussion & Further Work

In addition to techniques mentioned in the results, we experimented with adding a sliding window, compared to convoluting the entire image. By evaluating only a region of the image, the model should be less subjective to differing compositions of the input image, and may thus better identify the exact boundaries and features of the horses and zebras, despite what else the image may contain. The sliding window did not improve our results, we think with the correct hyperparameter settings it should speed up training and enhance the generated images. GANs are hard to train, as the generator and discriminator influence each other during training, and thus often require rigerous hyperparameter tuning

[1] https://arxiv.org/abs/1612.07828 2016 Ashish Shrivastava et. al. *Learning from Simulated and Unsupervised Images Through Adversarial Learning.*