EE898: PA2
CycleGAN: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Geeta Kumari (20194155) **Dated: 28th June 2020**

Introduction

In this assignment, we implement image-to-image translation using cyclegan, whereby given two images, the source domain is translated to the target domain using GAN architecture. Earlier work in this field mostly uses paired data for learning the translation. However, cycleGAN makes use of unpaired data, since the unpaired data are rather easy to collect than paired data. Thus, the goal is to learn a mapping G : X → Y such that the distribution of images G(X) is close to the distribution of images Y using adversarial loss. Since, this mapping is highly underconstrained, it could learn any mapping. So to induce more structure in the objective, it also learns inverse mapping F : Y → X and introduces cycle consistency loss, such that F(G(X)) ≈ X. The diagram below shows the described concept. This technique could be applied to many tasks, such as style transfer, object transfiguration, season transfer, photo enhancement, segmentation etc.

Concept Diagram

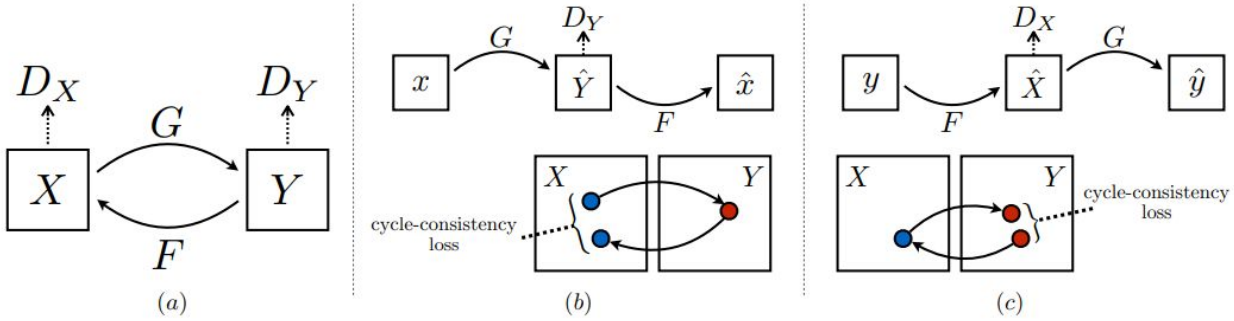


Figure 1. (a) The model contains two mapping functions G : X → Y and F : Y → X, and the associated adversarial discriminators Dy and Dx. The two cycle consistency losses capture the intuition that if one domain is translated to the other and back again, the final image should be same as the original at the start: (b) forward cycle-consistency loss: x → G(x) → F(G(x)) ≈ x, and (c) backward cycle-consistency loss: y → F(y) → G(F(y)) ≈ y

Implementation Steps

- **Step 1-1: CycleGAN Design**
  - The architecture for the generator and discrimantor are the same as used by the original paper.
  - The generator contains two stride-2 convolutions, 9 residual blocks (for 256 x 256 or greater sized images) and two fractionally strided convolutions with stride-½,

and one convolution at the start and end each with kernel size 7. It uses instance normalization.

- The discriminator network uses 70x70 patchGANs as used in pix2pix paper [1]. It aims to classify whether 70x70 overlapping image patches are real or fake. This type of discriminator has fewer parameters than a full-image discriminator and it can work on different-sized images in a fully-convolutional maner.

- **Step 1-2: CycleGAN Loss**
  - Two mapping functions: $G : X \rightarrow Y$ and $F : Y \rightarrow X$ with two adversarial discriminators Dx and Dy
  - Dx distinguishes images {x} from F(y) and Dy distinguishes from images {y} from G(x)
  - The final objective contains two losses: **adversarial loss** to match the distributions and **cycle consistency loss** to prevent the learnt translations G and F from contradicting each other
  - Adversarial loss is computed for both mapping functions
  - For $G : X \rightarrow Y$ and its discriminator Dy:

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

    G tries to generate images G(x) that look similar to images from domain Y , while Dy aims to distinguish between translated samples G(x) and real samples y. G aims to minimize this objective against an adversary D that tries to maximize it,

    $\min_G \max_{Dy} L_{GAN}(G, Dy , X, Y)$

    Similarly for the other mapping, $F : Y \rightarrow X$ and its discriminator Dx,

    $\min_F \max_{Dx} L_{GAN}(F, Dx , X, Y)$

  - Cycle consistency loss makes sure that the translated image can be translated back to its original form i.e. $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ and $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. The former is called forward cycle consistency and the later is called backward cycle consistency.
  - This can be induced by using a cycling consistency loss:

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

  - The full objective is:

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

    λ controls the relative importance of the two objectives. It is set to 10 in this implementation

  - The aim is to solve:

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y)$$

- However, in this implementation, for GAN loss, mean squared error loss is used instead of log likelihood objective, as it makes training more stable with better performance. In particular, for a GAN loss $L_{GAN}(G, D, X, Y)$, the G is trained to minimize

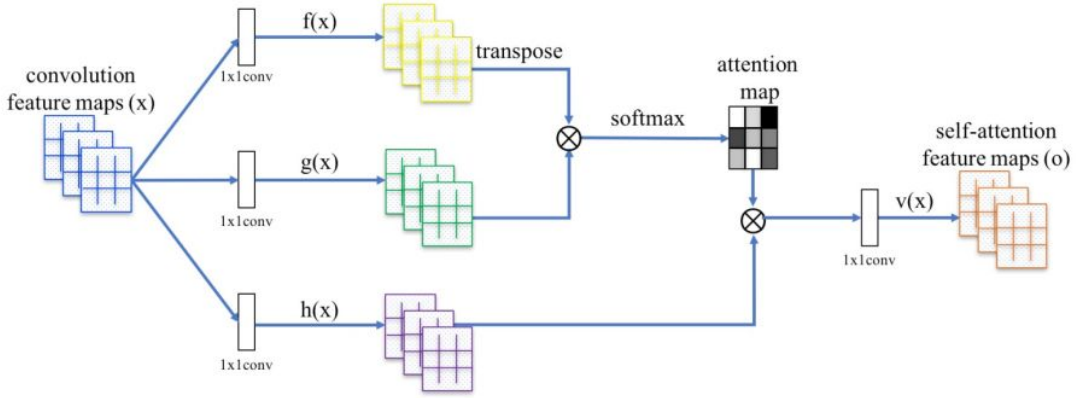
  $$E_{x \sim pdata(x)} [(D(G(x)) - 1)^2]$$

  and the D is trained to minimize

  $$E_{y \sim pdata(y)} [(D(y) - 1)^2] + E_{x \sim pdata(x)} [D(G(x))^2]$$

- Also, the discriminator is updated using a history of generated images rather than the ones produced by the latest generators. An image buffer is, thus, maintained for 50 previously generated images. This means the generator is trained first and then the discriminator iteratively

- **Step 2: Self-Attention for CycleGAN**
  - In this implementation, attention module is used to further assess the performance of the cycleGAN
  - The Attention module introduced in the Self-Attention GAN [2] paper shows that with a self-attention layer, the generator can generate images in which fine details at every location are carefully coordinated with fine details in distant portions of the image.
  - Generally, GAN-based models for image generation are built using convolutional layers, which process information in a local neighborhood. This, hence, does not allow for long-range dependencies in images
  - The self-attention module is adapted for non-local dependencies, enabling both generator and discriminator to efficiently model relationships between widely separated spatial regions
  - The attention module implemented is as follows,



  - The image features from the previous hidden layer $x \in R^{C \times N}$ are first transformed into two feature spaces f, g to calculate the attention, where $f(x) = W_f x$, $g(x) = W_g x$

$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{N} \exp(s_{ij})}, \text{ where } s_{ij} = \boldsymbol{f}(\boldsymbol{x_i})^T \boldsymbol{g}(\boldsymbol{x_j})$$

and $\beta_{j,i}$ indicates the extent to which the model attends to the ith location when synthesizing the jth region. C is the number of channels and N is the number of feature locations of features from the previous hidden layer. The output of the attention layer is o = (o1, o2, ..., oj , ..., oN) ∈ $R^{C \times N}$, where,

$$\boldsymbol{o_j} = \boldsymbol{v}\left(\sum_{i=1}^{N} \beta_{j,i} \boldsymbol{h}(\boldsymbol{x_i})\right), \ \boldsymbol{h}(\boldsymbol{x_i}) = \boldsymbol{W_h x_i}, \ \boldsymbol{v}(\boldsymbol{x_i}) = \boldsymbol{W_v x_i}$$

W_g ∈ $R^{\bar{C} \times C}$, W_f ∈ $R^{\bar{C} \times C}$, W_h ∈ $R^{\bar{C} \times C}$, and W_v ∈ $R^{C \times \bar{C}}$ are the learned weight matrices, which are implemented as 1×1 convolutions. $\bar{C}$ = C/k, where k = 8. The output of the attention module is further multiplied with a learnable scale parameter and added back to input feature map, as follows:

$$\boldsymbol{y_i} = \gamma \boldsymbol{o_i} + \boldsymbol{x_i},$$

- The module was inserted in the baseline generator model at two locations, one before resnet block and one after that and in discriminator too, it was added at two locations - one before the last conv layer and one after first 4 conv layers
- In this implementation, the attention module is intended to produce better translations i.e. only relevant parts are translated. Such as in horse2zebra dataset, only horses or zebras should be translated from one to other, leaving background unaffected

- **Step 3: FID Score [3]**
  - FID (Frechet Inception Distance) is a measure of similarity between two datasets of images. It is shown to correlate with human judgement of visual quality and hence, is often used to evaluate the quality of generated images using GANs.
  - FID basically computes the Frechet distance between two Gaussians fitted to feature representations of the Inception network
  - FID equation:

$$FID^2 = ||m_f - m_r||_2^2 + \mathrm{Tr}(C_f + C_r - 2(C_f C_r)^{1/2})$$

  - In this implementation, we measure the distance between the real images and the fake images (the translated ones) in feature space by using pretrained InceptionV3 network on IMAGENET.
  - Different layers could be used for features extraction, in this implementation, features are extracted from the final average pooling layer
  - After extracting features of real and fake data, mean and variance of the gaussian distribution of features are computed and using above FID equation, FID score is computed

Training Details

The model was implemented four times for two different datasets (maps and horse2zebra) and two different models (baseline and baseline with attention module).
All were trained on 2 GPUs (GeForce GTX 1080 Ti), with the learning rate set to 0.0001, batch size to 4 and total epochs to 400 with the linear reduction of learning rate to zero after 200th epoch.

Results & Analysis

1. **Quantitative**

- Baseline CycleGAN indicates the original implementation of the paper
- Baseline+Self-attention indicates the baseline CycleGAN with self-attention module added
- Target like source is Fake B (horse translated to zebra, aerial views translated to maps) and Target is Real B (Zebra, maps)
- Source like target is Fake A (zebra translated to horse, maps translated to aerial views) and source is Real A (Horse, aerial views)
- FID score of baseline CycleGAN is much better than the baseline with self-attention module added for both datasets, showing the attention module does not help in the image translation tasks in this case
- Also, it is noted that one way translation is better than the opposite direction i.e. horses are better transformed to zebras than zebras to horses and aerial views are better translated to maps than maps to aerial views
- Aerial views are more detailed, hence it is understood if the maps to aerial views perform better than the other way round
- Similarly, horses are more varied than zebras in the dataset besides the background and lighting changes. Almost all zebras are just black and white stripes, while horses are in various colors. Hence, horses are more effectively transformed to zebras than zebras to horses
- Perhaps better models are needed to cater for such differences in the unpaired datasets

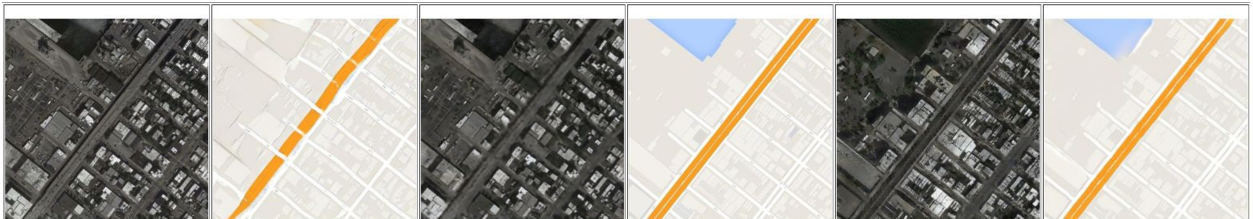
FID Score/Model	Baseline CycleGAN	Baseline+Self-attention
Maps - target like source and target	106	166
Maps - source like target and source	111	133
Horse2Zebra - target like source and target	68	94
Horse2Zebra - source like target and source	144	163

2. Qualitative

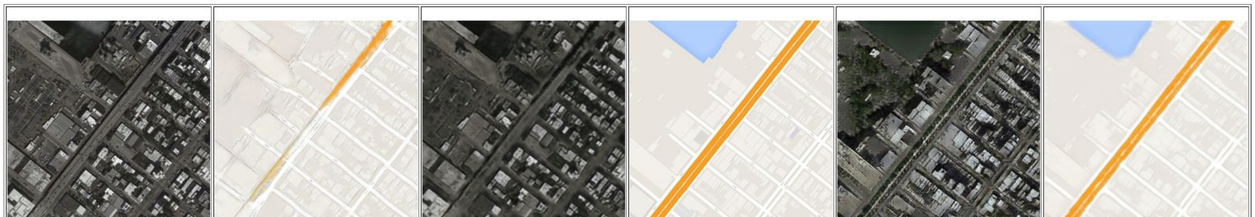
- For each 2 image rows, the first row shows the baseline results and second one shows the baseline with attention results
- In each row, real image corresponds to the original, fake to the translated and rec to the reconstructed image from the generated image using inverse mapping
- As can be seen from the qualitative results, baseline with attention module does not give better results than the baseline in both datasets
- In the generated images, self-attention modules introduces artifacts and distortions, as can be seen in the generated horse images and also in some of the maps images
- Reconstructed images are very similar to the original, showing that cycle-consistency loss makes the network cycle-consistent

real_A | fake_B | rec_A | real_B | fake_A | rec_B

real_A | fake_B | rec_A | real_B | fake_A | rec_B



real_A | fake_B | rec_A | real_B | fake_A | rec_B



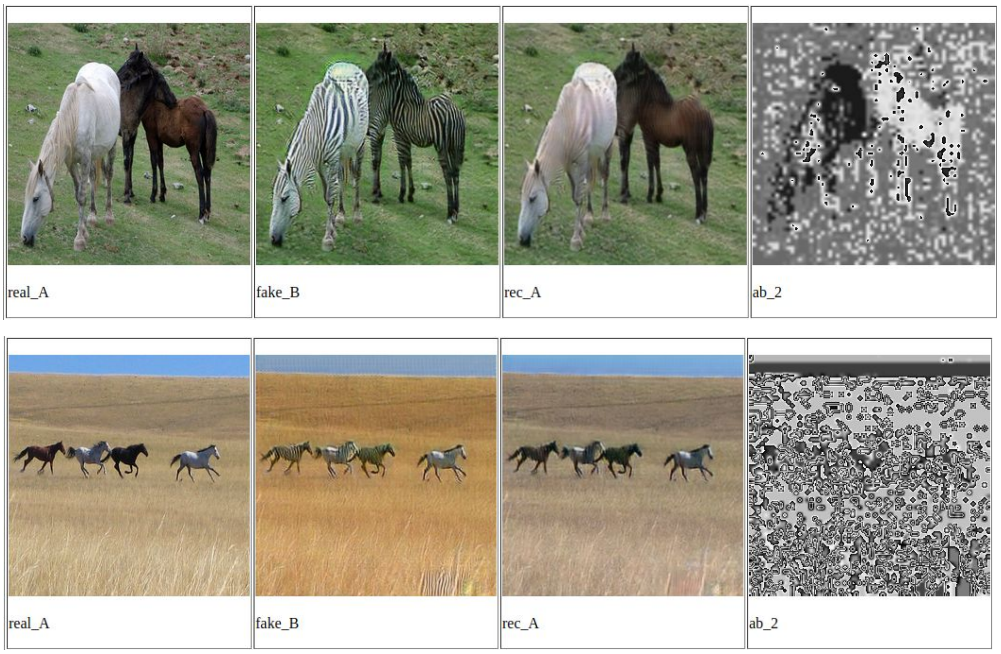
real_A | fake_B | rec_A | real_B | fake_A | rec_B

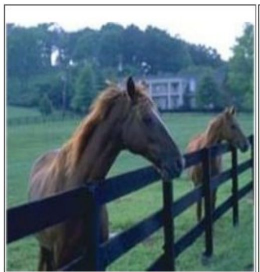
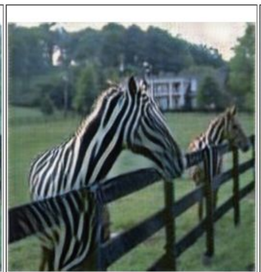
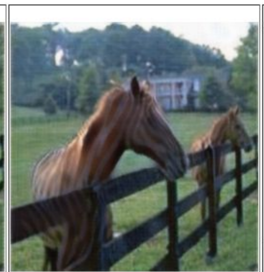
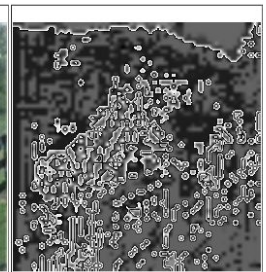
real_A | fake_B | rec_A | real_B | fake_A | rec_B

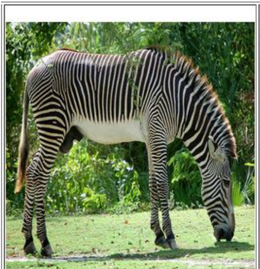
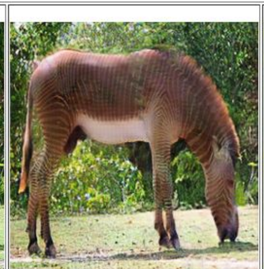
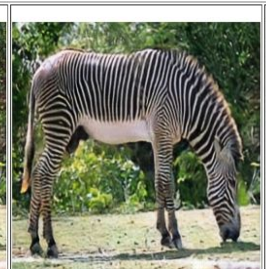
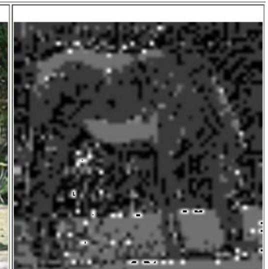
real_A | fake_B | rec_A | real_B | fake_A | rec_B

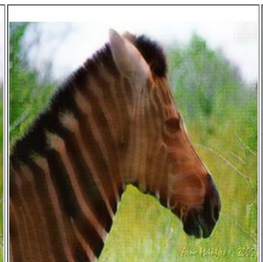
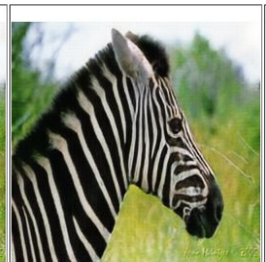
3. Attention Maps Visualization - Horse2Zebra & Maps

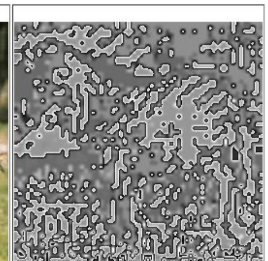
- Attentions maps/masks are generated by taking the average across all features after the application of attention module and then upsampling them to the image size
- The attention maps generated show that in most of the cases they do not quite correspond to relevant areas for translations for both datasets
- In the case of maps dataset, for maps to aerial views translation, the attention maps do not show any useful information.
- Either better attention modules need to be designed or they need to be placed at more appropriate places in network



			
real_A	fake_B	rec_A	ab_2

			
real_B	fake_A	rec_B	aa_2

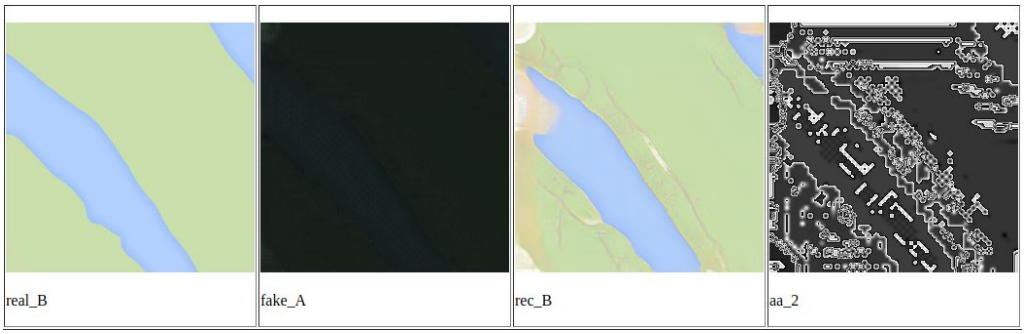
			
real_B	fake_A	rec_B	aa_2

			
real_B	fake_A	rec_B	aa_2

			
real_A	fake_B	rec_A	ab_2

real_B | fake_A | rec_B | aa_2

real_A | fake_B | rec_A | ab_2



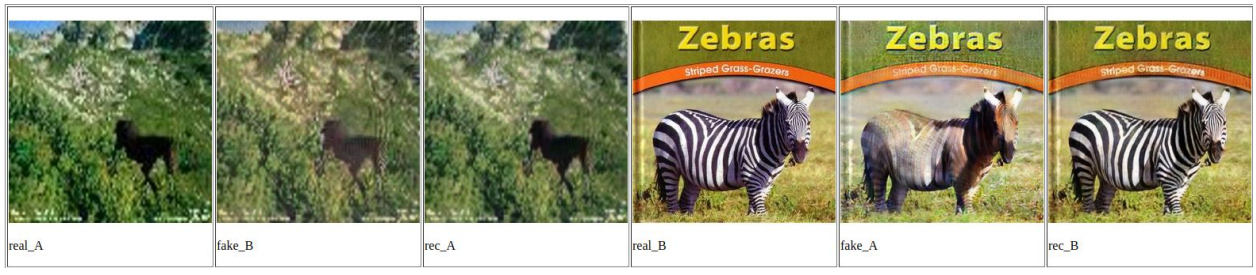
real_B | fake_A | rec_B | aa_2

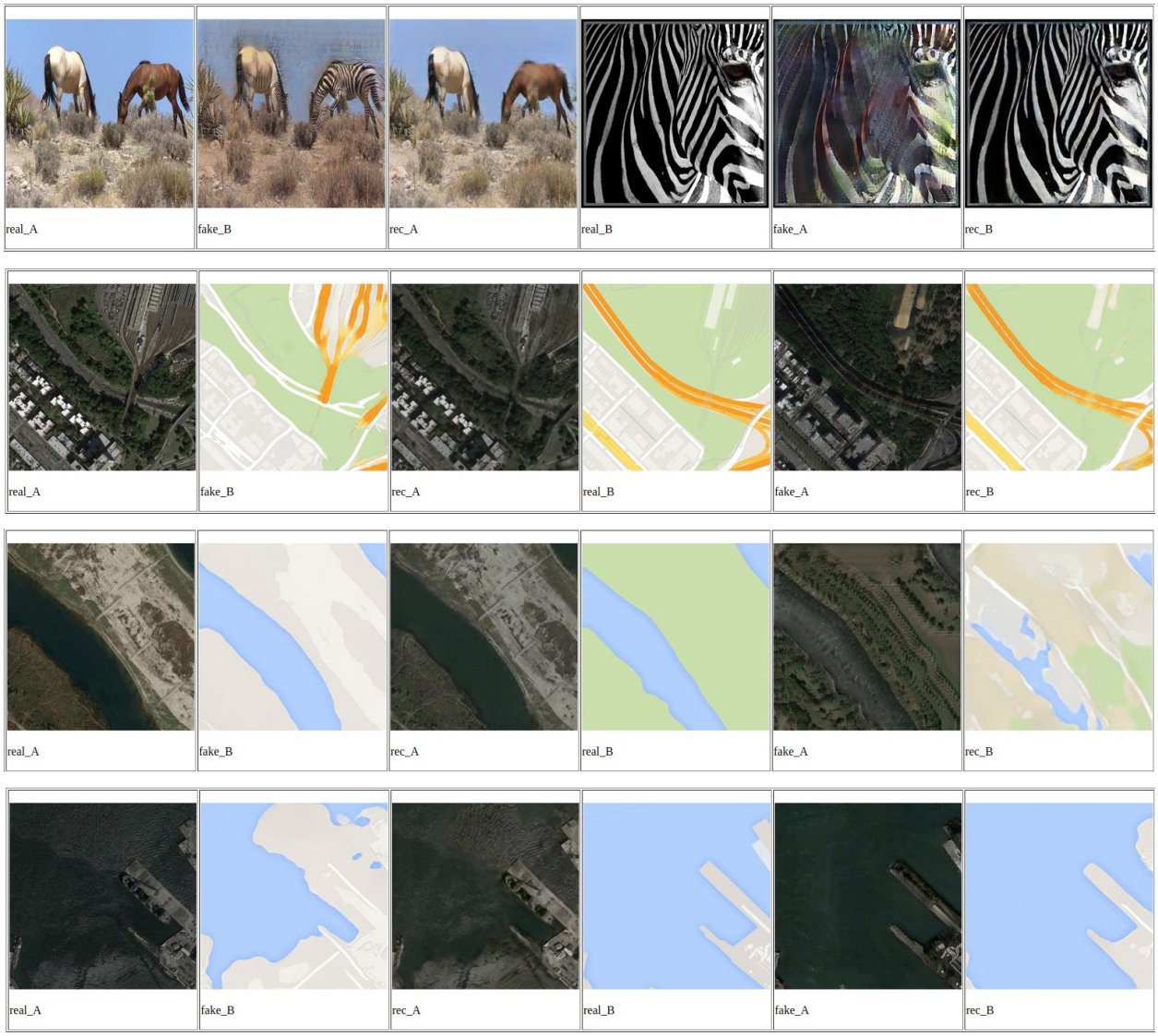
real_B | fake_A | rec_B | aa_2

real_A | fake_B | rec_A | ab_2

Discussion

- The cycleGAN model gives very compelling results on image-to-image translation tasks for both the datasets -maps and horse2zebra. However, there are many failure cases, where either an image is not translated (horse2zebra) at all or translated incorrectly (maps) as shown in images below.
- This model may not work for more varied and extreme transformations, especially where geometric changes are required
- The data distribution has to be more varied too for training, so that for test cases, more meaningful results are generated - as can be seen in the first row of images below, where a human is transformed to zebra too. Current datasets deals with horses and zebra mostly alone in the wild, where in more real cases there may be more things or different animals in the images
- This method being unsupervised with the use of unpaired data has great potential for image-to-image translation if more improved training models or methods are employed with more data training
- Current setting trains each task individually, which is very time consuming and sample inefficient too. It would be more desirable and interesting if more translation tasks are employed using single setting by using meta-learning or few-shot learning techniques





References

1. Image-to-Image Translation with Conditional Adversarial Networks, Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros
2. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Sepp Hochreiter
3. Self-Attention Generative Adversarial Networks, Han Zhang, Ian Goodfellow, Dimitris Metaxas, Augustus Odena