AttentionGAN: Unpaired Image-to-Image Translation

Using Attention Guided Generative Adversarial Networks

Presented by:
Akhoury Shauryam
Sampad Kumar Kar
S. Aslah Ahmad Faizi

Chennai Mathematical Institute

August 30, 2023

Table of Contents

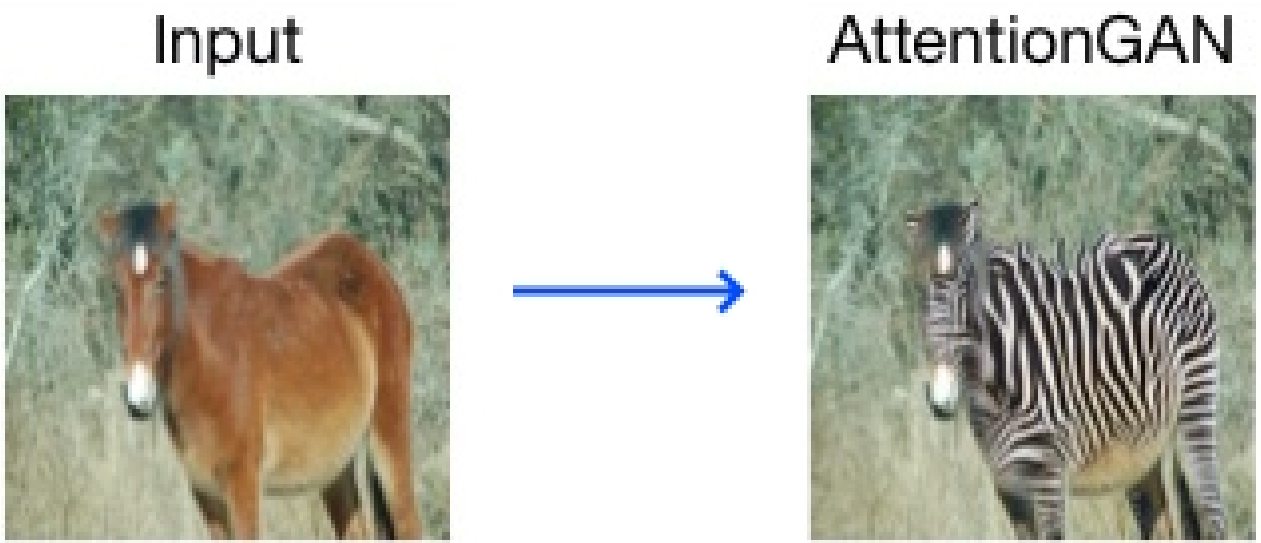
Table of Contents

Introduction

What is AttentionGAN?

- Uses Attention-Guided Generative Adversarial Networks
- Only modifies distinctive features in the foreground
- Generates masks to keep the background unchanged

- Uses Attention-Guided Generative Adversarial Networks
- Only modifies distinctive features in the foreground
- Generates masks to keep the background unchanged



It can be used to change styles of certain dominating part of an image and change facial emotions.

Introduction

How does it use Attention?

- Introduces two schemes
- Both have slightly different architecture but use the same concept of masking
- Uses attention to generate relevant masks for identifying foreground and background

Introduction

How does it use Attention?

- Introduces two schemes
- Both have slightly different architecture but use the same concept of masking
- Uses attention to generate relevant masks for identifying foreground and background

- Uses GAN to modify input x from domain X to generate G_y in domain Y
- Then uses a reverse generator to recover R_x from input G_y
- Minimizes the **Cycle-Consistency Loss** between x and R_x

Introduction

Comparision with other GANs

- Some other models for this task are CycleGAN, GANimorph, DiscoGAN, and DualGAN

Introduction

Comparision with other GANs

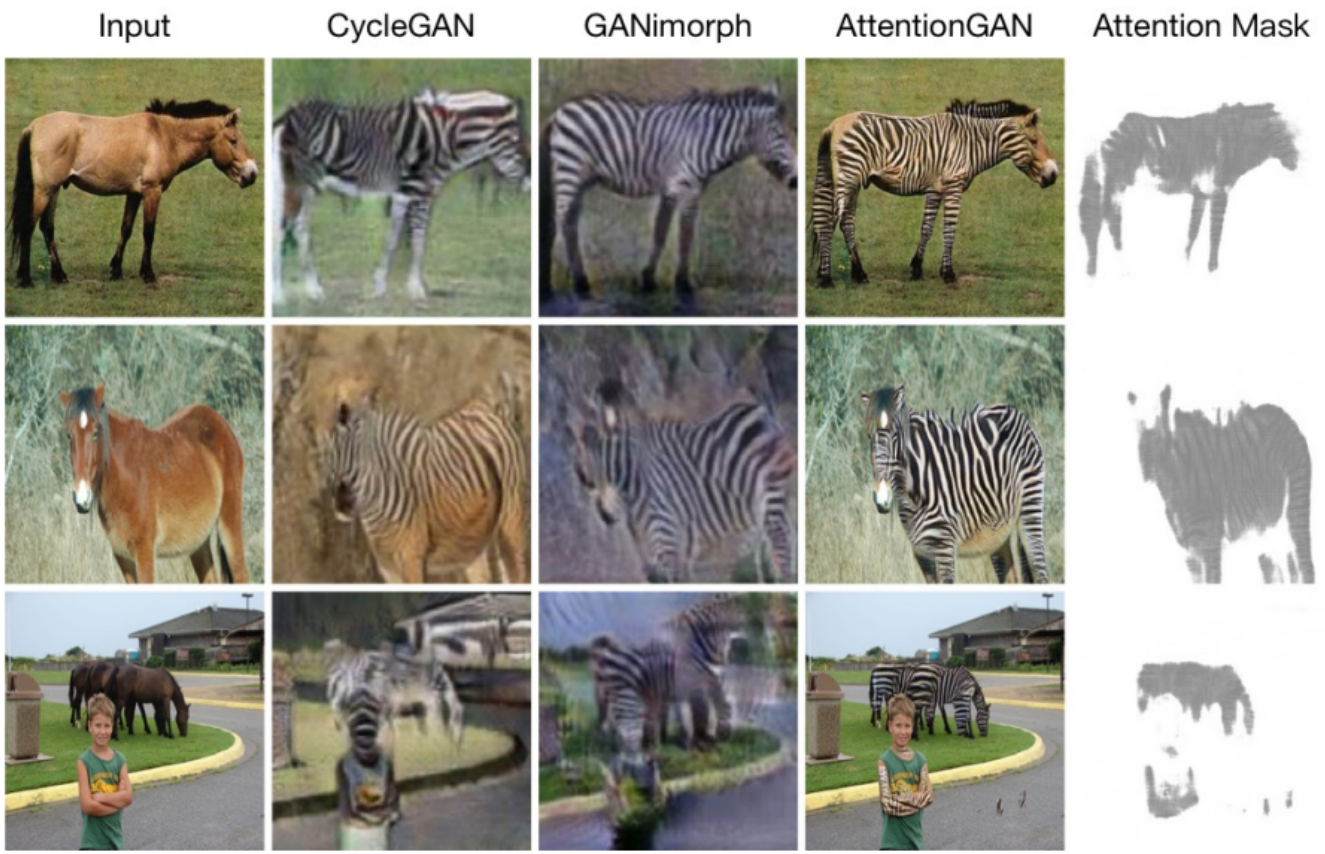
- Some other models for this task are CycleGAN, GANimorph, DiscoGAN, and DualGAN
- AttentionGAN gives much better results than most
- Attention mask helps in identifying complex foregrounds, giving edge over other GANs

TABLE IX: KID $\times$ 100 $\pm$ std. $\times$ 100 for different methods. For this metric, lower is better. Abbreviations: (H)orse, (Z)ebra, (A)pple, (O)range.

Method	H $\rightarrow$ Z	Z $\rightarrow$ H	A $\rightarrow$ O	O $\rightarrow$ A
DiscoGAN [5]	13.68 $\pm$ 0.28	16.60 $\pm$ 0.50	18.34 $\pm$ 0.75	21.56 $\pm$ 0.80
RA [48]	10.16 $\pm$ 0.12	10.97 $\pm$ 0.26	12.75 $\pm$ 0.49	13.84 $\pm$ 0.78
DualGAN [4]	10.38 $\pm$ 0.31	12.86 $\pm$ 0.50	13.04 $\pm$ 0.72	12.42 $\pm$ 0.88
UNIT [44]	11.22 $\pm$ 0.24	13.63 $\pm$ 0.34	11.68 $\pm$ 0.43	11.76 $\pm$ 0.51
CycleGAN [3]	10.25 $\pm$ 0.25	11.44 $\pm$ 0.38	8.48 $\pm$ 0.53	9.82 $\pm$ 0.51
UAIT [11]	6.93 $\pm$ 0.27	8.87 $\pm$ 0.26	**6.44 $\pm$ 0.69**	5.32 $\pm$ 0.48
AttentionGAN	**2.03 $\pm$ 0.64**	**6.48 $\pm$ 0.51**	10.03 $\pm$ 0.66	**4.38 $\pm$ 0.42**

Introduction

Visual Examples

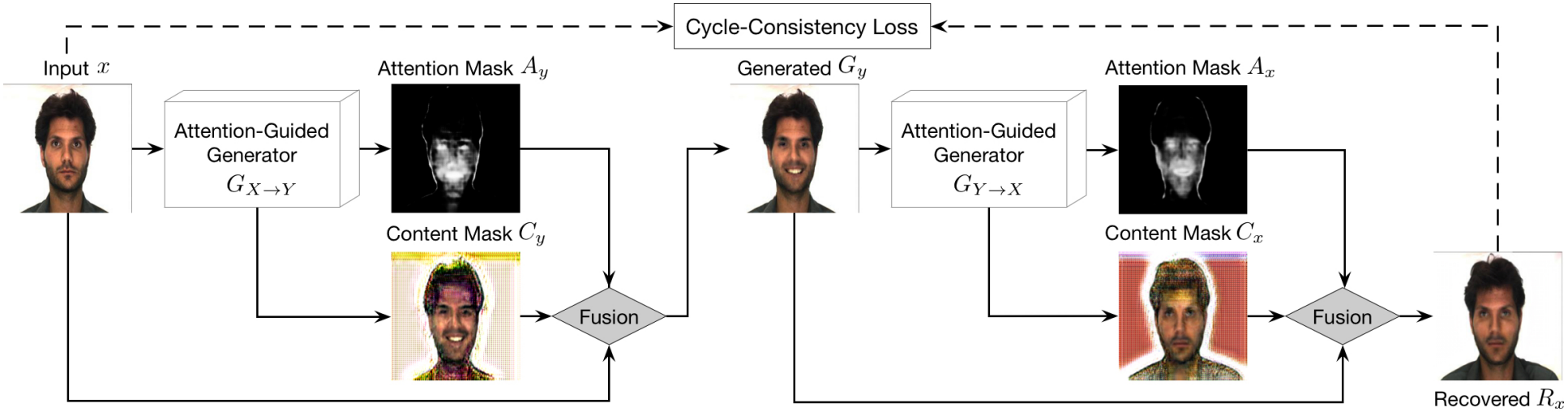


Even without the quantitative comparison, we can see what works better visually!

Table of Contents

Scheme 1

Overview



The above describes the basic structure of the Scheme 1. It creates 2 different types of mask and uses 2 different generators.

Scheme 1

Working

- Two generators, G and F. G transforms from domain X to Y, and F from Y to X.

Scheme 1

Working

- Two generators, G and F. G transforms from domain X to Y, and F from Y to X.
- Input is a 3-channel image.

Scheme 1

Working

- Two generators, G and F. G transforms from domain X to Y, and F from Y to X.
- Input is a 3-channel image.
- G returns an attention mask A_y, a gray scale 1-channel image.
- It also return a content mask C_y, which is a 3-channel image. C_y can be thought of as a normal transformation of the image.

Scheme 1

Working

- Two generators, G and F. G transforms from domain X to Y, and F from Y to X.
- Input is a 3-channel image.
- G returns an attention mask A_y, a gray scale 1-channel image.
- It also return a content mask C_y, which is a 3-channel image. C_y can be thought of as a normal transformation of the image.
- We apply $G(x) = C_y * A_y + x * (1 - A_y)$. This is done pixel-wise to get the image G_y in domain Y.

Scheme 1

Working

- Two generators, G and F. G transforms from domain X to Y, and F from Y to X.
- Input is a 3-channel image.
- G returns an attention mask A_y, a gray scale 1-channel image.
- It also return a content mask C_y, which is a 3-channel image. C_y can be thought of as a normal transformation of the image.
- We apply $G(x) = C_y * A_y + x * (1 - A_y)$. This is done pixel-wise to get the image G_y in domain Y.

- A similar process is done with generator F to get back R_x from G_y. $R_x = F(y) = C_x * A_x + y * (1 - A_x)$

Scheme 1

Flaws

There are 3 major drawbacks of following scheme 1, these are:

1. Attention Mask and Content Mask are generated by the same GAN, which has been seen in resulting in low quality images.

Scheme 1

Flaws

There are 3 major drawbacks of following scheme 1, these are:

1. Attention Mask and Content Mask are generated by the same GAN, which has been seen in resulting in low quality images.
2. It produces only one attention mask to differ foreground and background, which doesn't give a lot of information.

Scheme 1

Flaws

There are 3 major drawbacks of following scheme 1, these are:

1. Attention Mask and Content Mask are generated by the same GAN, which has been seen in resulting in low quality images.
2. It produces only one attention mask to differ foreground and background, which doesn't give a lot of information.
3. It only produces one content mask to select useful content. Therefore, Scheme 1 is not able to deal with complex tasks such as horse to zebra translation properly

Table of Contents

Scheme 2

Overview

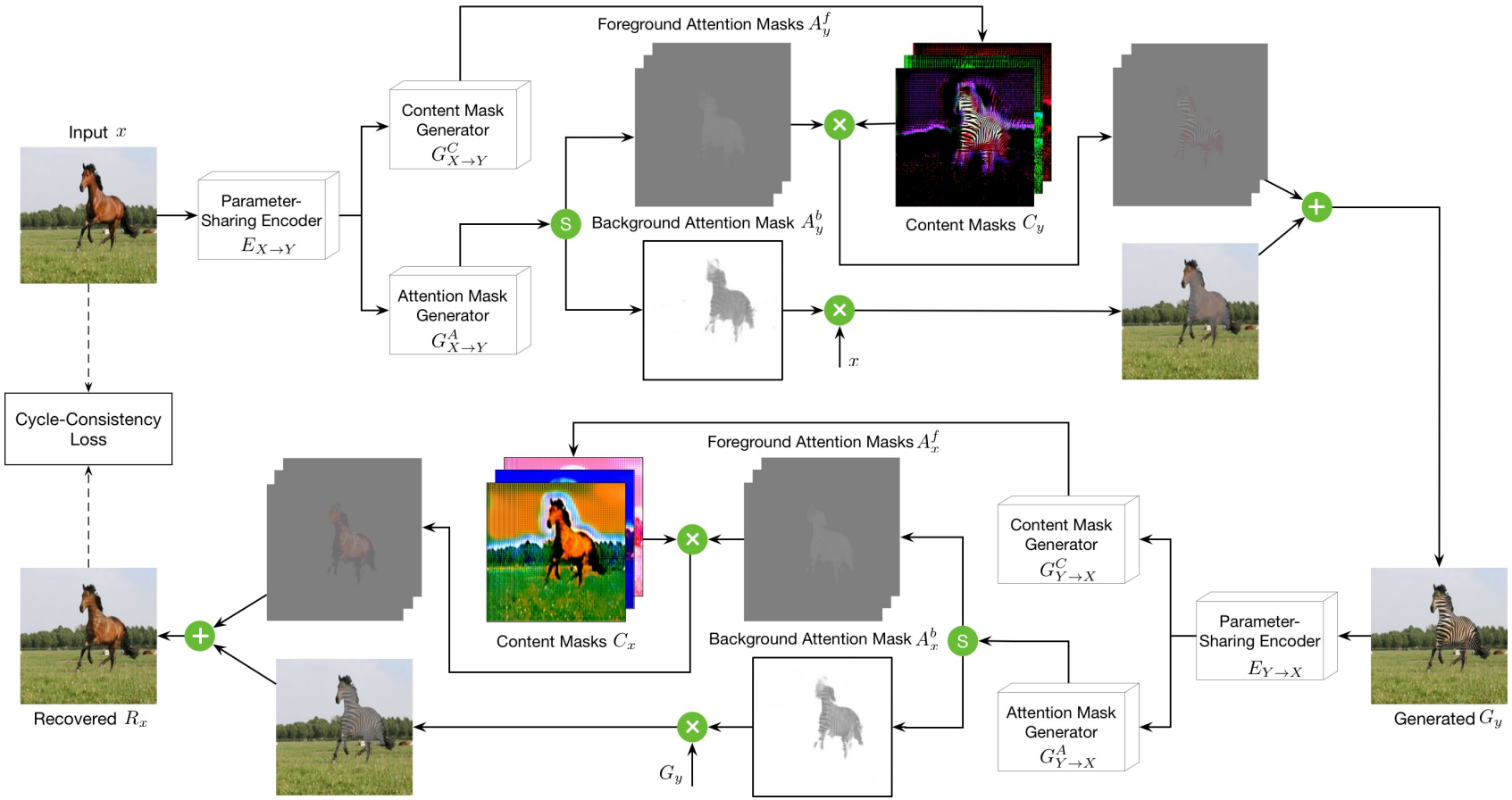


Figure above shows the architecture of Scheme 2, where it uses multiple Attention masks to fuse into needed translation.

Scheme 2

Working

- G consists of G_E, G_A, G_C

Scheme 2

Working

- G consists of G_E, G_A, G_C
- G_E encodes the input x, and sends it to G_A and G_C

Scheme 2

Working

- G consists of G_E, G_A, G_C
- G_E encodes the input x, and sends it to G_A and G_C
- G_A is a generator. It returns $n-1$ foreground attention masks A_y^f, and one background attention mask A_y^b.
  This is in contrast to Scheme 1, which only generated one foreground mask and used it for the background as well.

Scheme 2

Working

- G consists of G_E, G_A, G_C
- G_E encodes the input x, and sends it to G_A and G_C
- G_A is a generator. It returns $n-1$ foreground attention masks A_y^f, and one background attention mask A_y^b.
  This is in contrast to Scheme 1, which only generated one foreground mask and used it for the background as well.
- Similarly, G_C returns $n-1$ content masks.

$$C_y^f = \mathbf{tanh}(mW_C^f + b_C^f)$$

$$A_y^f = \mathbf{Softmax}(mW_A^f + b_A^f)$$

m is the feature map extracted through G_E

Scheme 2

Working

- The masks and the original image are fused together to get the modified image
- Each mask focuses on a different feature of the foreground, making it easier to modify complex images.

$$G_y = \sum_{1}^{n-1}(C_y^f * A_y^f) + x * A_y^b$$

Scheme 2

Improvements over Scheme 1

~~Attention Mask and Content Mask are generated by the same GAN, which has been seen in resulting in low quality images.~~

1. Different GANs used in scheme 2.

Scheme 2

Improvements over Scheme 1

~~Attention Mask and Content Mask are generated by the same GAN, which has been seen in resulting in low quality images.~~

1. Different GANs used in scheme 2.

~~It produces only one attention mask to differ foreground and background, which doesn't give a lot of information.~~

2. Different masks for foreground and background with multiple foreground masks.

Scheme 2

Improvements over Scheme 1

~~Attention Mask and Content Mask are generated by the same GAN, which has been seen in resulting in low quality images.~~

1. Different GANs used in scheme 2.

~~It produces only one attention mask to differ foreground and background, which doesn't give a lot of information.~~

2. Different masks for foreground and background with multiple foreground masks.

~~It only produces one content mask to select useful content. This results in Scheme 1 not being able to deal with complex tasks such as horse to zebra translation flawlessly~~

3. Scheme 2 Produces multiple content masks to build up more information for better translation.

The Two Schemes

Differences: Example

Table of Contents

Discriminator

- Vanilla discriminators consider the whole image, even though generators only act on the attended part.

Discriminator

- Vanilla discriminators consider the whole image, even though generators only act on the attended part.
- Additionally, an attention-guided discriminator is used. It is structurally the same, but also takes the attention mask as input.
- Discriminator D_{YA} takes $[A_y, G_y]$ and $[y, G_y]$ as input. This allows it to only consider the relevant part inside the mask and ignore unrelated content.

Discriminator

- Vanilla discriminators consider the whole image, even though generators only act on the attended part.
- Additionally, an attention-guided discriminator is used. It is structurally the same, but also takes the attention mask as input.
- Discriminator D_{YA} takes $[A_y, G_y]$ and $[y, G_y]$ as input. This allows it to only consider the relevant part inside the mask and ignore unrelated content.
- Similarly for the reverse discriminator D_{XA}.

Table of Contents

Optimization Objective

Loss

We will be working on two types of losses.

1. Reconstruction Loss: We will have objectives that tries to minimize the the difference between the original image and the reconstructed image.

Optimization Objective

Loss

We will be working on two types of losses.

1. Reconstruction Loss: We will have objectives that tries to minimize the the difference between the original image and the reconstructed image.
2. GAN Loss: We will also have loss functions that take care of the intermediate output of the generators and the working of the discriminators.

Optimization Objective

Reconstruction Loss

Cycle Loss: Our first optimization term will use the reconstructed image to make sure the mapped spaces are similar. This can be formulated as:

$$\mathcal{L}_{cycle}(G_{X\to Y}, G_{Y\to X}) = \mathbb{E}_{x\sim p_{data}(x)}[||R_x - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||R_y - y||_1]$$

$R_x = G_{Y\to X}(G_{X\to Y}(x))$ and $R_y = G_{X\to Y}(G_{Y\to X}(y))$

Optimization Objective

Reconstruction Loss

Cycle Loss: Our first optimization term will use the reconstructed image to make sure the mapped spaces are similar. This can be formulated as:

$$\mathcal{L}_{cycle}(G_{X\to Y}, G_{Y\to X}) = \mathbb{E}_{x\sim p_{data}(x)}[||R_x - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||R_y - y||_1]$$

$R_x = G_{Y\to X}(G_{X\to Y}(x))$ and $R_y = G_{X\to Y}(G_{Y\to X}(y))$

Pixel Loss: We also use pixel loss to reduce the difference between the generated images.

$$\mathcal{L}_{pixel}(G_{X\to Y}, G_{Y\to X}) = \mathbb{E}_{x\sim p_{data}(x)}[||G_{X\to Y}(x) - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||G_{Y\to X}(y) - y||_1]$$

Optimization Objective

GAN Loss

Adversarial Loss: We will first apply the vanilla generative adversarial loss.

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

We will have a similar loss for the other generator-discriminator pair.

Optimization Objective

GAN Loss

Adversarial Loss: We will first apply the vanilla generative adversarial loss.

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

We will have a similar loss for the other generator-discriminator pair.

Attention-Guided Adversarial Loss: We will also use the attention guided discriminator mentioned earlier. Its loss will be calculated as:

$$\begin{aligned}\mathcal{L}_{AGAN}(G_{X\to Y}, D_{YA}) =& \mathbb{E}_{y\sim p_{data}(y)}[\log D_{YA}([A_y, y])] + \\ & \mathbb{E}_{x\sim p_{data}(x)}[\log(1 - D_{YA}([A_y, G_{X\to Y}(x)]))]\end{aligned}$$

Similarly for the other generator-discriminator pair.

Attention Loss: Since we do not have a ground truth for the attention masks, they can easily saturate to 1. To prevent this, a Total Variation regularization term is introduced.

$$\mathcal{L}_{tv}(M_x) = \sum_{w,h=1}^{W,H} |A_x(w+1,h,c) - A_x(w,h,c)| + |A_x(w,h+1,c) - A_x(w,h,c)|$$

Optimization Objective

Final Loss

Our final optimization objective will be:

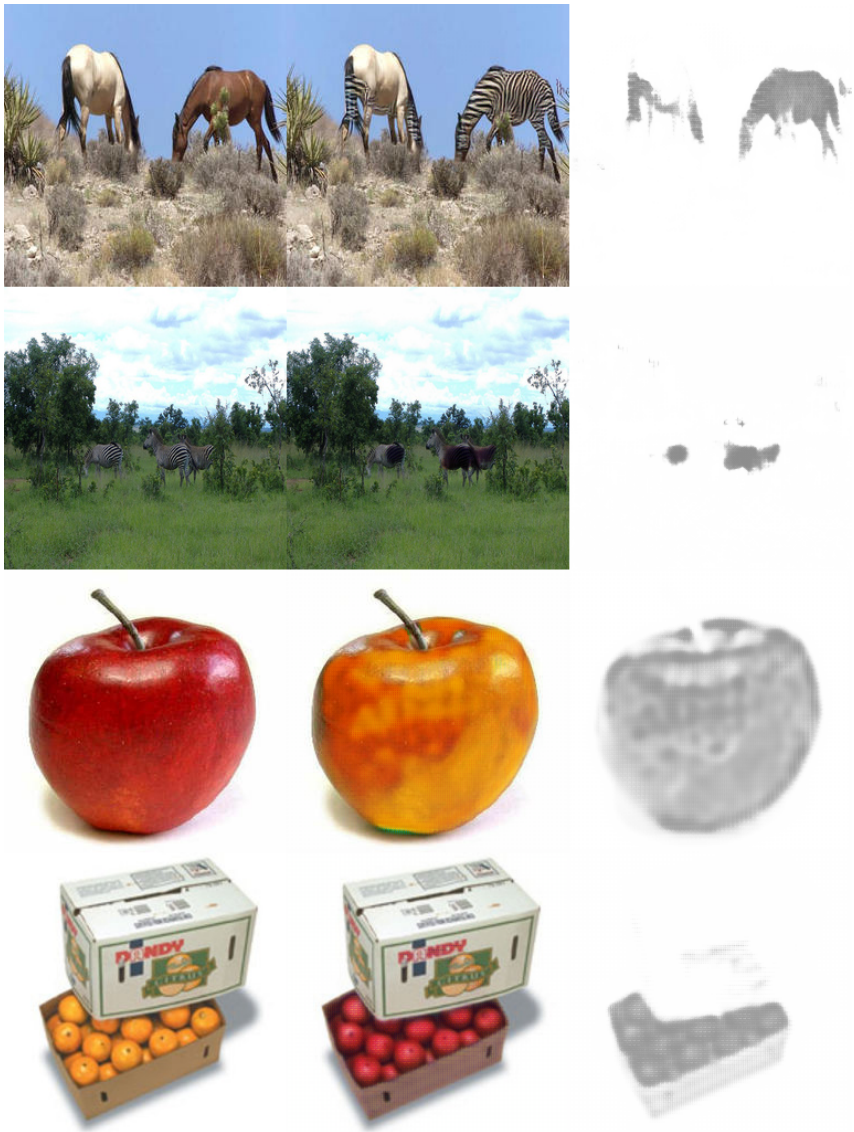
$$\begin{aligned}\mathcal{L} =&[\lambda_{cycle} * \mathcal{L}_{cycle} + \lambda_{pixel} * \mathcal{L}_{pixel}] * r + \\ &[\lambda_{gan} * (\mathcal{L}_{GAN} + \mathcal{L}_{AGAN}) + \lambda_{tv} * \mathcal{L}_{tv}] * (1 - r)\end{aligned}$$

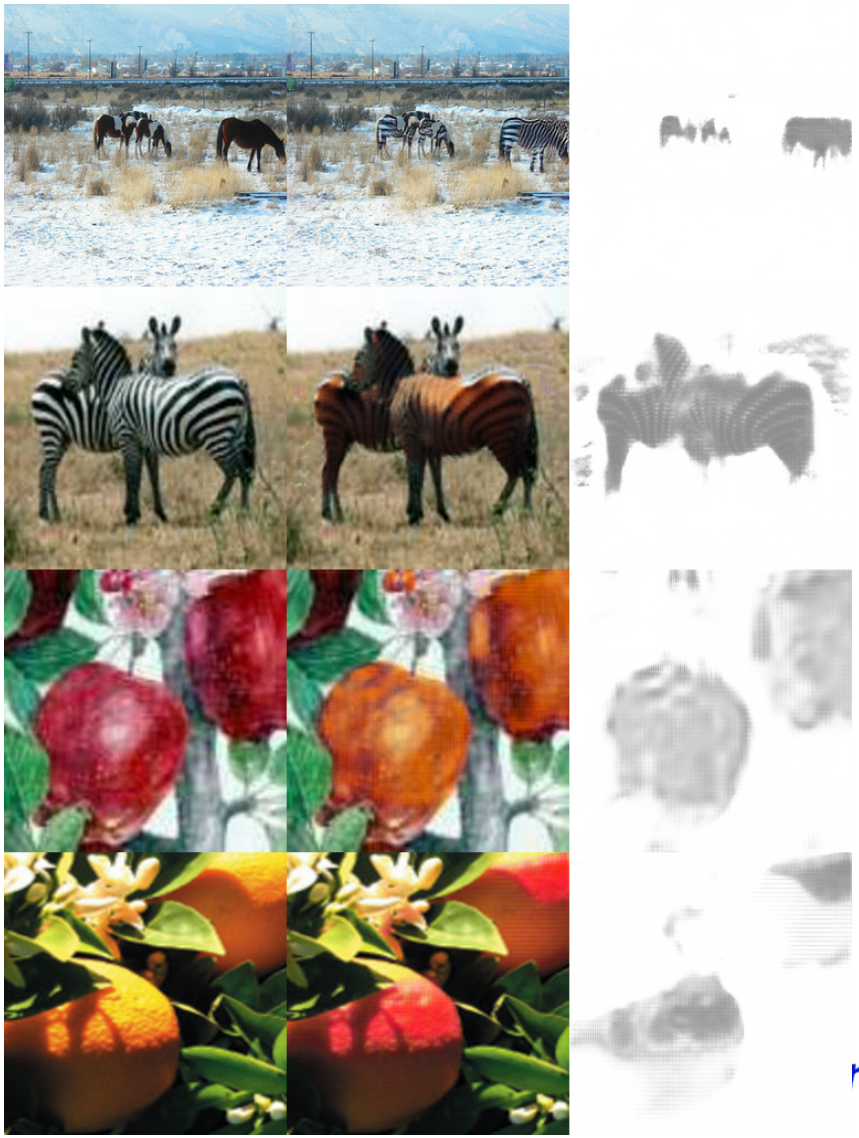
λ_{cycle}, λ_{pixel}, λ_{gan} and λ_{tv} are parameters controlling the relative relation of objectives terms.
r is a curriculum parameter to control the relation between GAN loss and reconstruction loss.

Table of Contents

Our Results

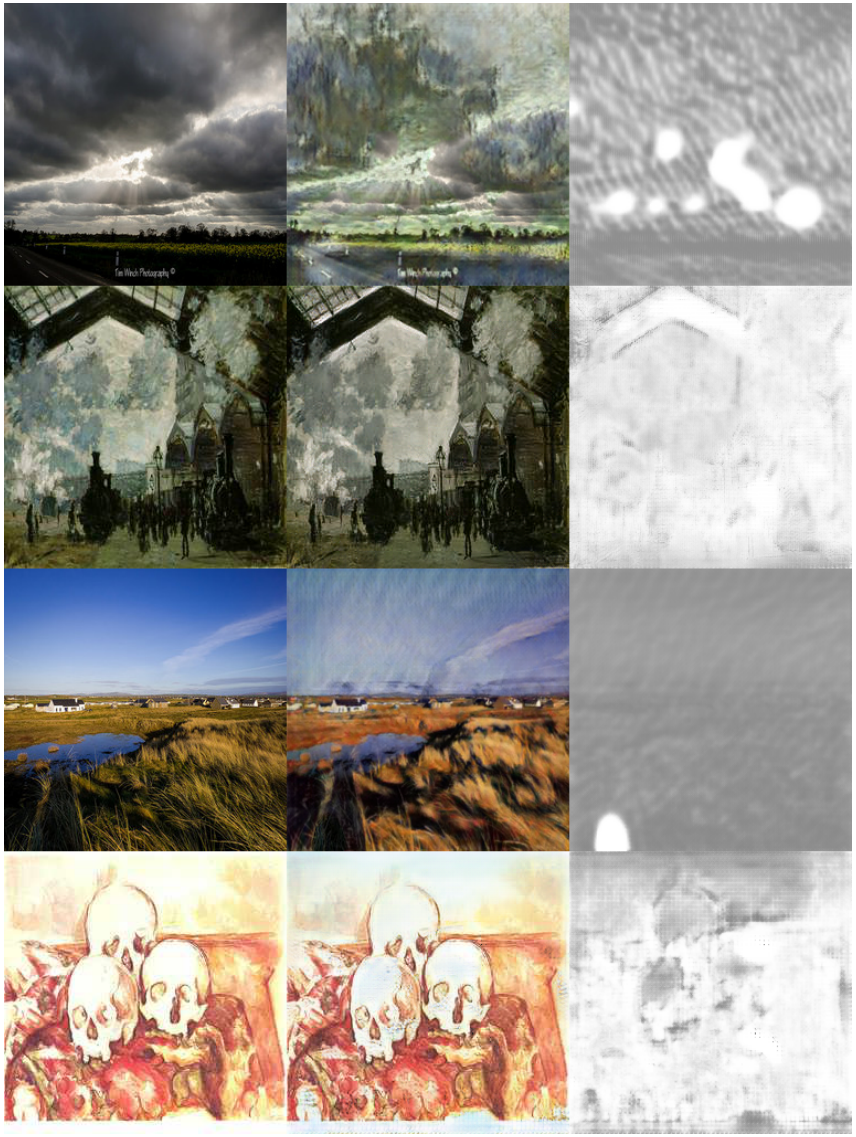
Modifications of Images

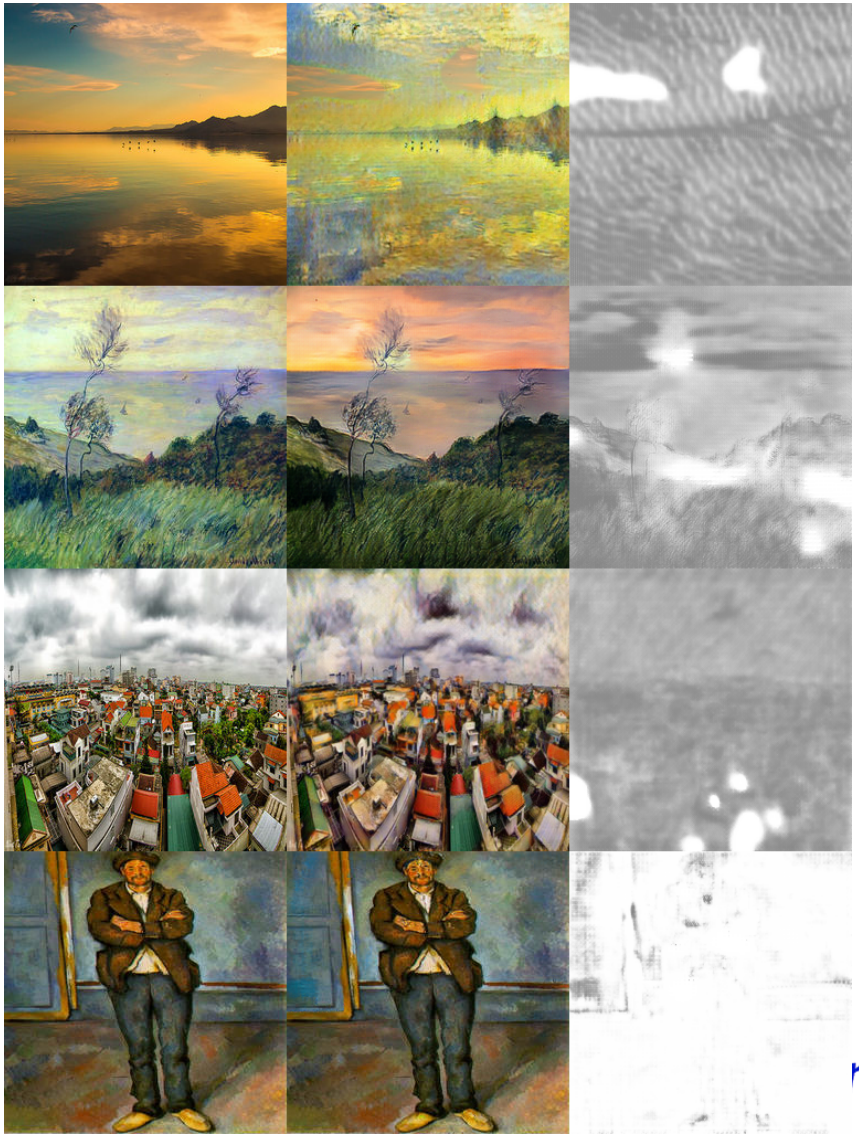




Our Results

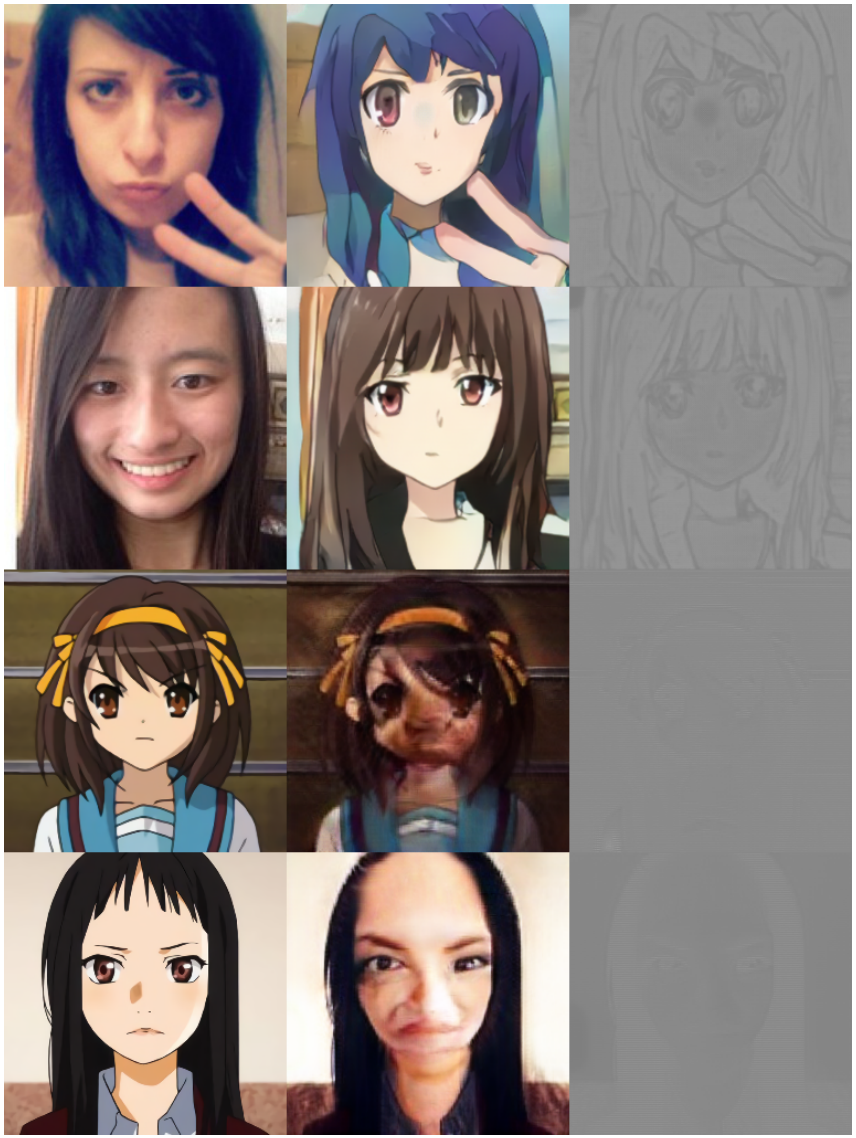
Modifications of Images 2

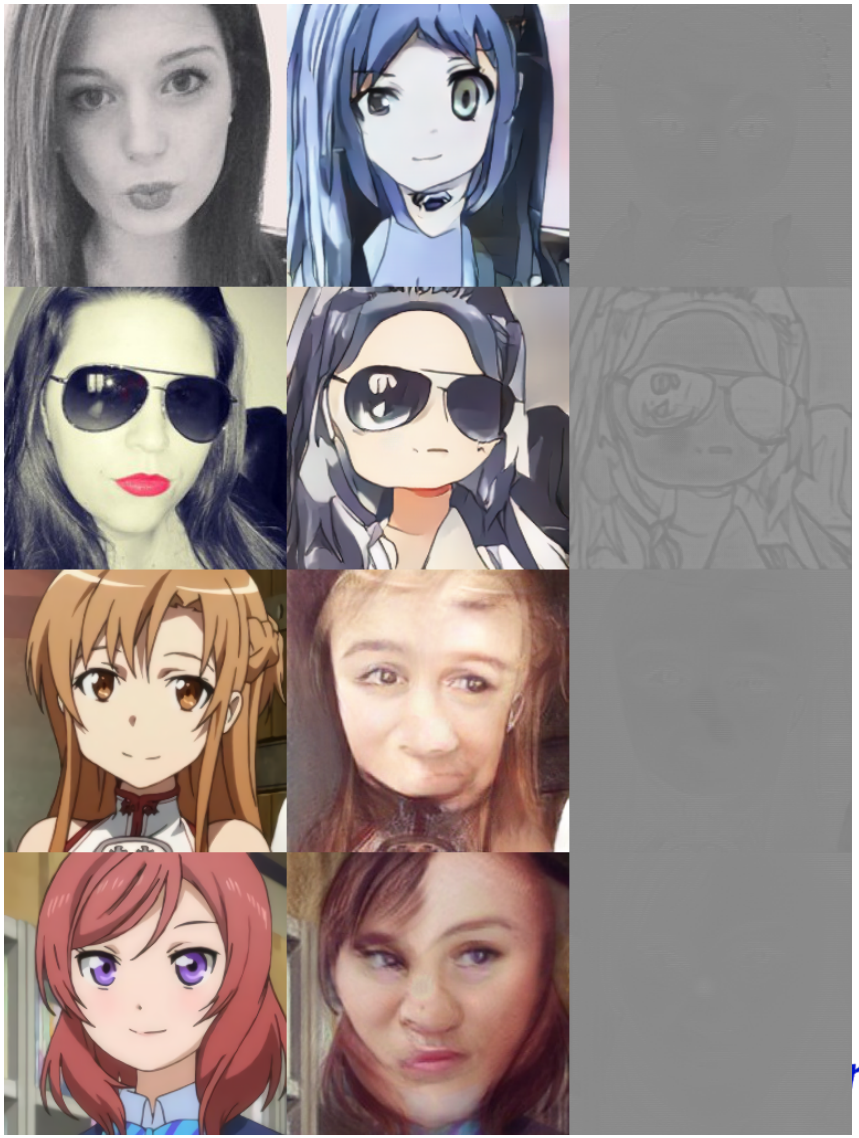




Our Results

Low Quality Success with Selfie2Anime





Thank You for Your Attention!