

Image-to-Image Translation with Conditional Adversarial Networks

Mansour DIA

27 août 2021

Introduction

Introduction

■ GANS

Introduction

- GANS
- DCGAN

Introduction

- GANS
- DCGAN
- cGAN

Introduction

- GANS
- DCGAN
- cGAN
- Pix2Pix

Introduction

- GANS
- DCGAN
- cGAN
- Pix2Pix

General Architecture

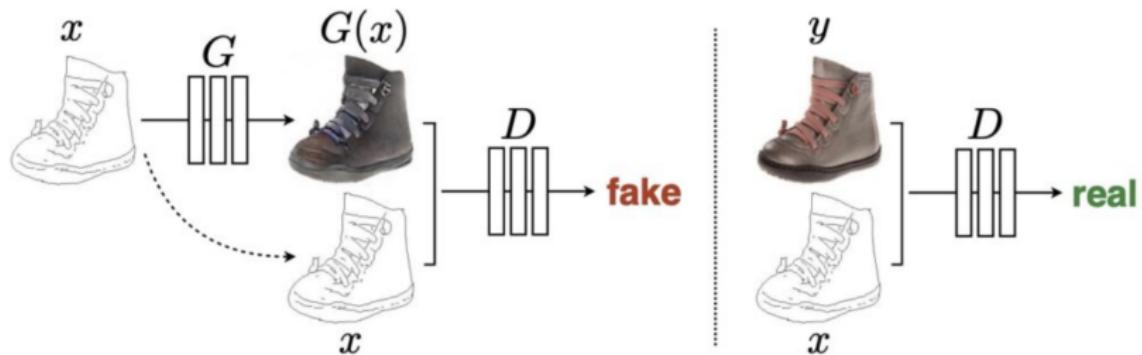


FIGURE – Pix2Pix Architecture.

PatchGAN Discriminator

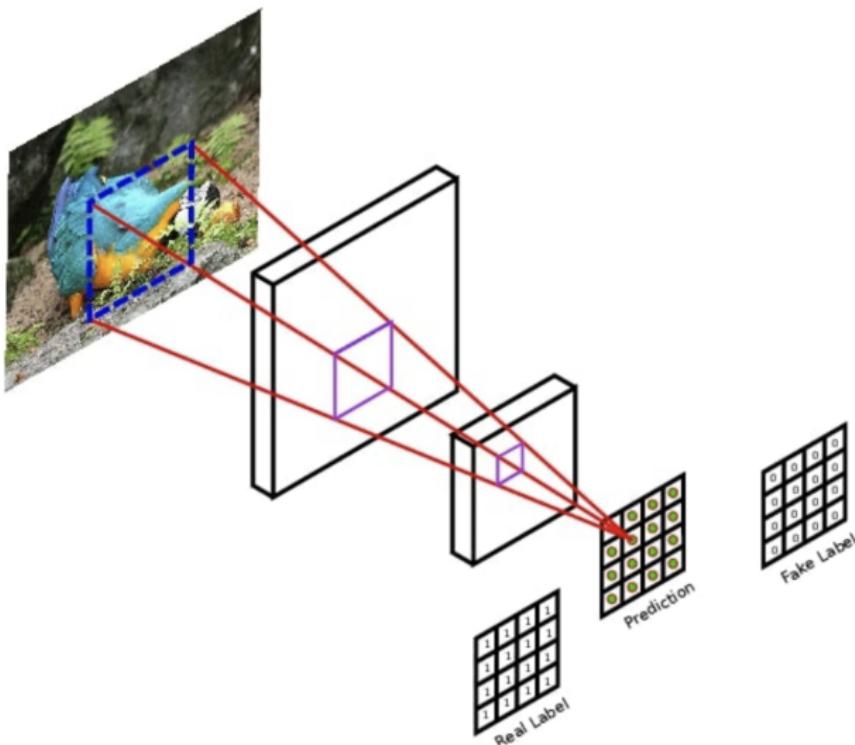


FIGURE – PatchGAN architecture.

Unet Generator

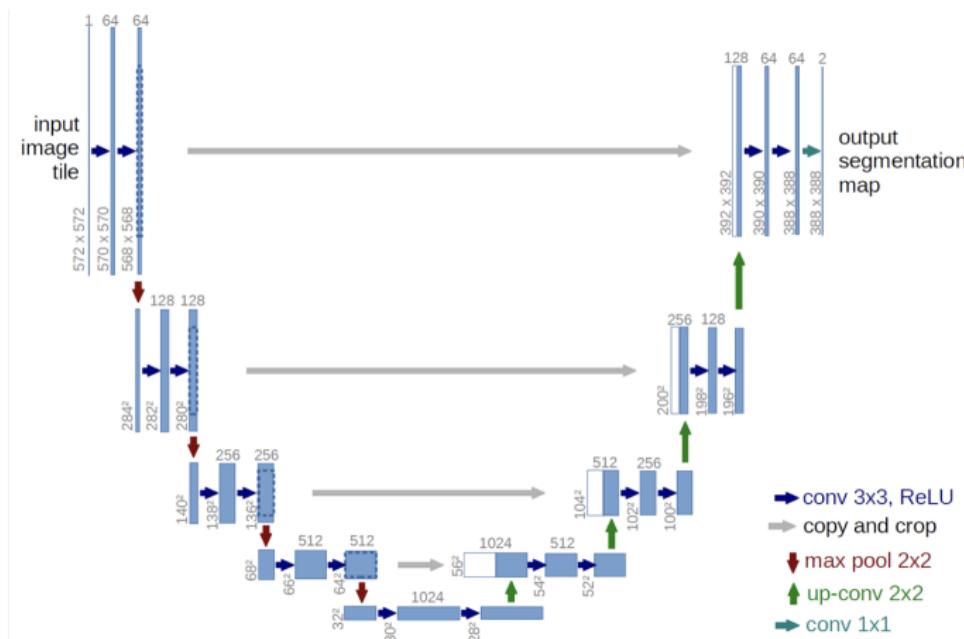


FIGURE – Unet architecture.

Pix2Pix loss function

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]$$

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1].$$

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G).$$

patGAN hyperparameters

- concat(real_input, gen_image)

patGAN hyperparameters

- concat(real_input, gen_image)
- EncoderUnit($2 \times Conv2D + BatchNorm + LeakyReLU$),

patGAN hyperparameters

- concat(real_input, gen_image)
- EncoderUnit($2 \times Conv2D + BatchNorm + LeakyReLU$),
- $PatchGAN_first_layer + 4 \times EncoderUnit + PatchGAN_last_layer$

patGAN hyperparameters

- concat(real_input, gen_image)
- EncoderUnit($2 \times Conv2D + BatchNorm + LeakyReLU$),
- $PatchGAN_first_layer + 4 \times EncoderUnit + PatchGAN_last_layer$

Unet hyperparameters

- EncoderUnit($2 \times Conv2D + BatchNorm + LeakyReLU$)

Unet hyperparameters

- EncoderUnit($2 \times Conv2D + BatchNorm + LeakyReLU$)
- DecoderUnit($3 \times Conv2D, BatchNorm + ReLU$)

Unet hyperparameters

- EncoderUnit($2 \times Conv2D + BatchNorm + LeakyReLU$)
- DecoderUnit($3 \times Conv2D, BatchNorm + ReLU$)
- Unet_first_layer + $7 \times EncoderUnit + 7 \times DecoderUnit + unet_last_layer$

Unet hyperparameters

- EncoderUnit($2 \times Conv2D + BatchNorm + LeakyReLU$)
- DecoderUnit($3 \times Conv2D, BatchNorm + ReLU$)
- Unet_first_layer + $7 \times EncoderUnit + 7 \times DecoderUnit + unet_last_layer$

The training set

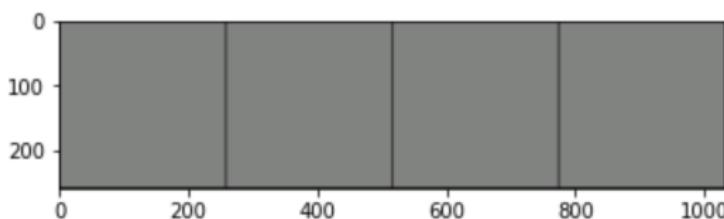
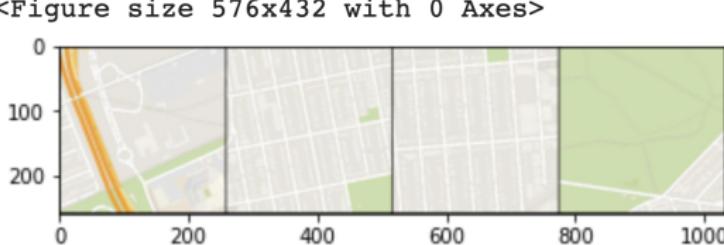
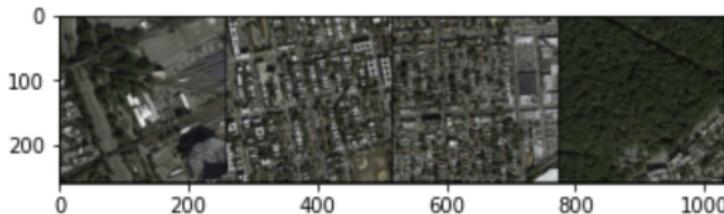
- Maps - aerial photograph ;
- around New City ;
- 2194 examples

Final hyperparameters setting

- n_epochs = 30 ;
 - ICGAN_criterion = nn.BCEWithLogitsLoss() ;
 - pix_dist_criterion = nn.L1Loss() ;
 - lambda_pix_dist = 200 ;
 - input_img_channels = 3
 - real_img_channels = 3
 - batch_size = 4
 - lr = 0.0002

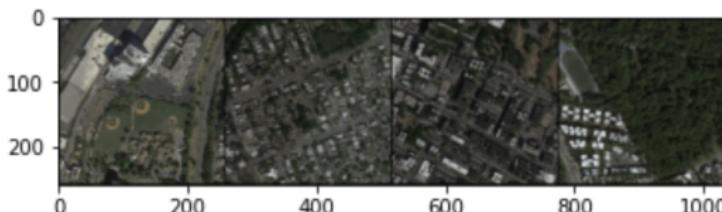
Results

epoch 1 : gen_loss = 73.94 dis_loss = 0.65

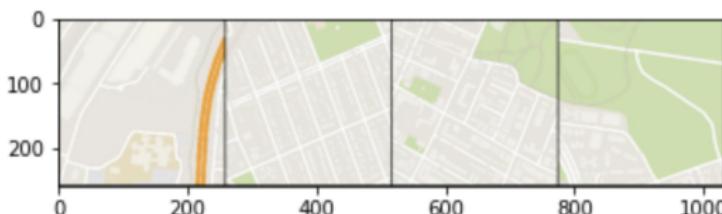


Results

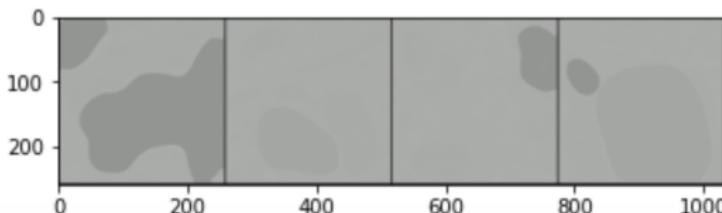
epoch 2 : gen_loss =37.39 dis_loss = 0.29



<Figure size 576x432 with 0 Axes>

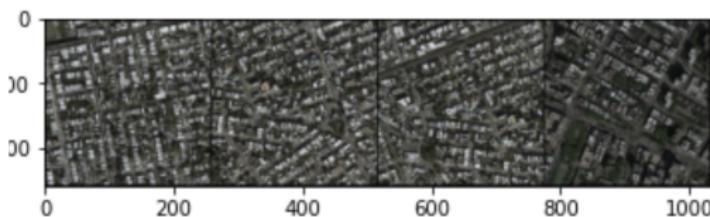


<Figure size 576x432 with 0 Axes>



Results

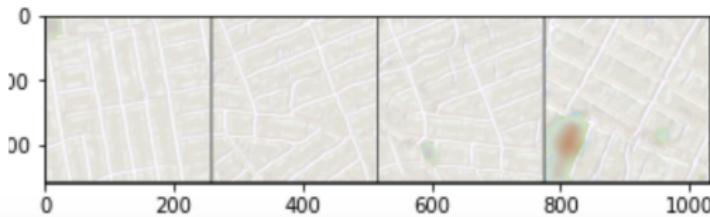
epoch 15 : gen_loss = 10.57 dis_loss = 0.27



'igure size 576x432 with 0 Axes>

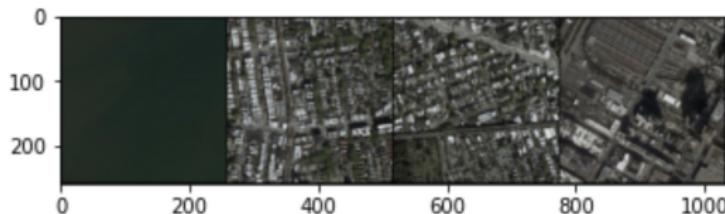


'igure size 576x432 with 0 Axes>

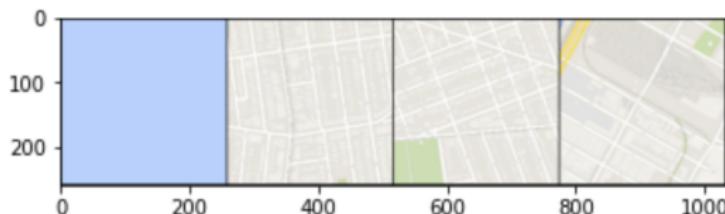


Results

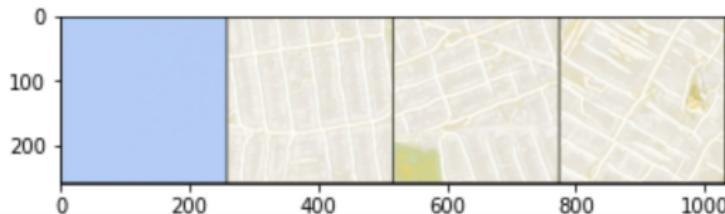
epoch 16 : gen_loss = 10.50 dis_loss = 0.29



<Figure size 576x432 with 0 Axes>

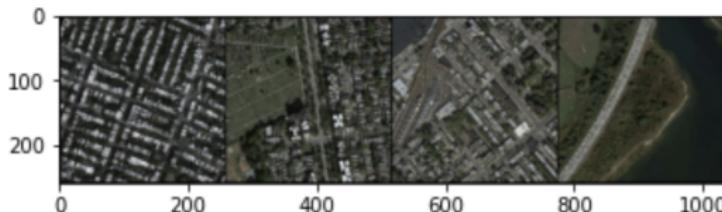


<Figure size 576x432 with 0 Axes>

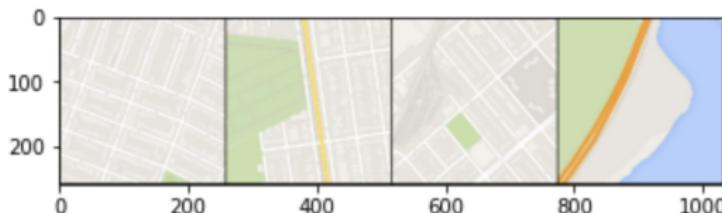


Results

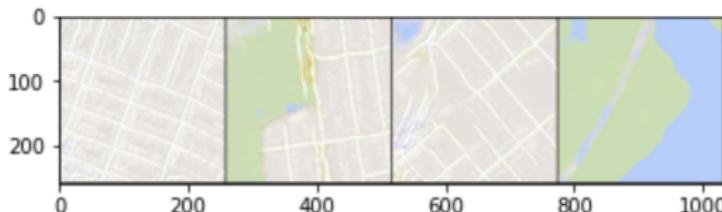
epoch 29 : gen_loss = 10.83 dis_loss = 0.205



<Figure size 576x432 with 0 Axes>

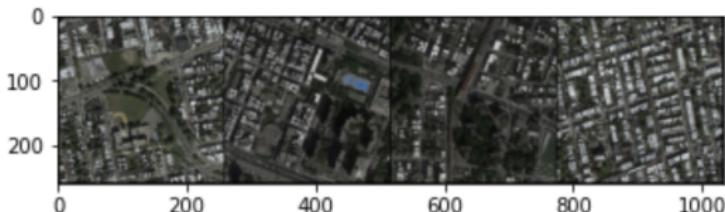


<Figure size 576x432 with 0 Axes>

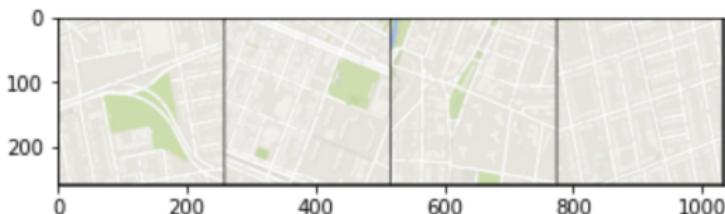


Results

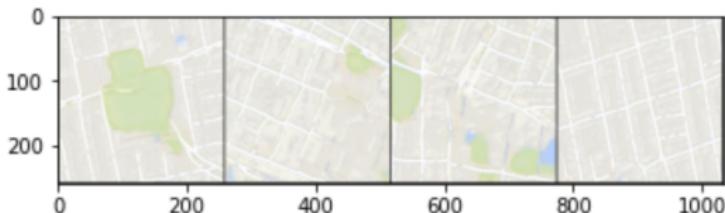
epoch 30 : gen_loss10.97 dis_loss = 0.17



<Figure size 576x432 with 0 Axes>



<Figure size 576x432 with 0 Axes>



Future work

- The other way around ;

Future work

- The other way around ;
- unsupervised image-to-image translation.

End

Thank you for your attention !