Machine Vision Project

Generative Adversarial Networks for improving classification and Neural style transfer

Yujin CHO, University Paris-Saclay

SUMMARY

PART1 : CLASSIFICATION

PART2 : DATA AUGMENTATION (GANs)

PART3 : UNPAIRED NEURAL STYLE TRANSFER

Conclusion

PART1 : CLASSIFICATION

GOAL : Design a COW and HORSE classifier based on CNN architecture

INPUTS :

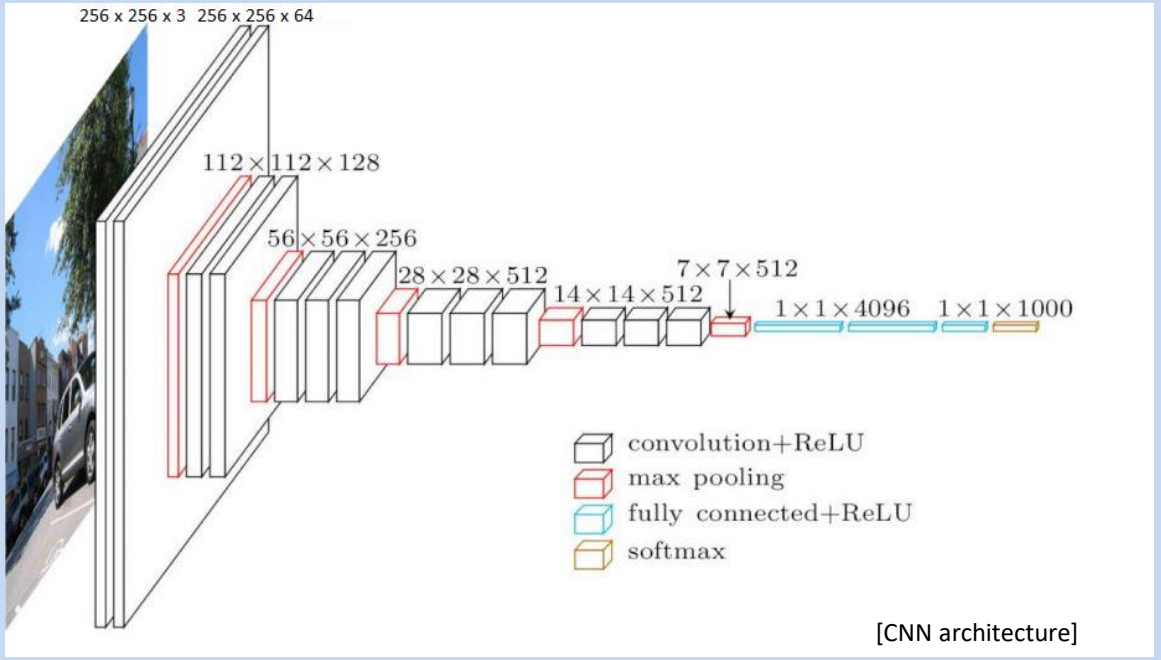
Classes : “COW” and “HORSE”
Dataset size : 41 images of each class
Image dimension : 256 x 256 x 3 pixels
1 training set and 1 test set

TRAINING DETAILS :

- 50 epochs
- Learning rate : 0.005
- Loss function : binary cross entropy loss(BCE loss)
- Optimizer : SGD(Stochastic Gradient Descent)

SOLUTION : CNN architecture based on VGG 16 architecture
Only 16 layers (13 CONV + 3 FC)

Custom CNN :
Modification of first convolution
Adding a last fully connected



[CNN architecture]

PART1 : CLASSIFICATION

GOAL : Design a COW and HORSE classifier based on CNN architecture

INPUTS :

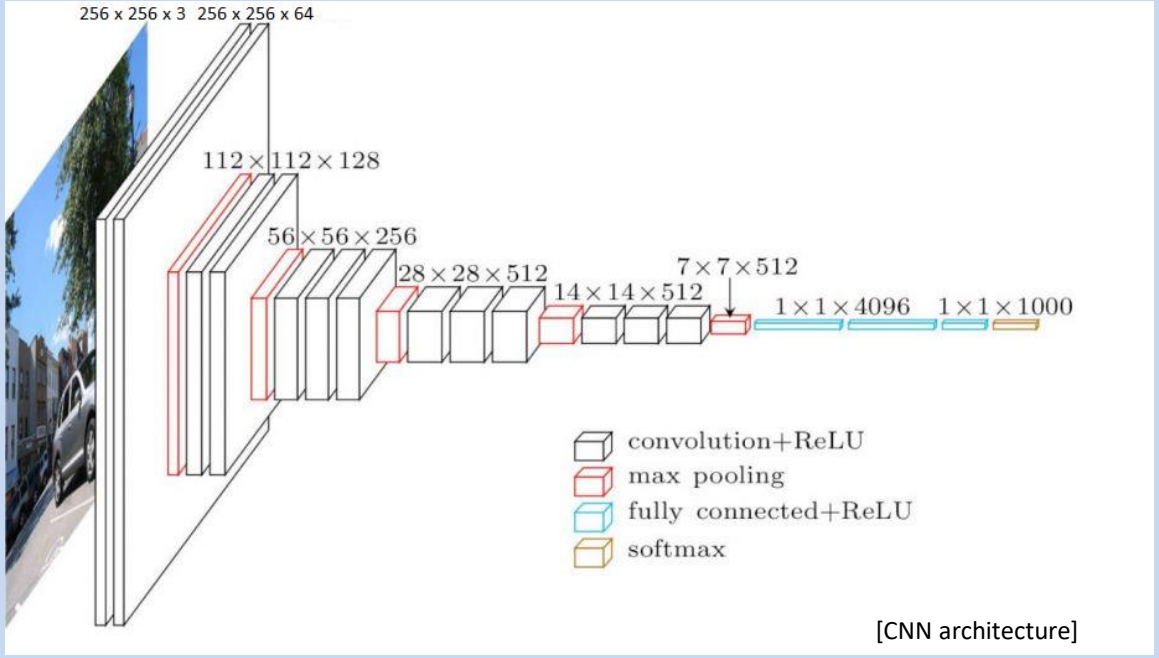
Classes : “COW” and “HORSE”
Dataset size : 41 images of each class
Image dimension : 256 x 256 x 3 pixels
1 training set and 1 test set

RESULTS :

Accuracy	Normal	with data augementation
Network	74,4%	100%
Cow	85,4%	100%
Horse	63,4%	100%

SOLUTION : CNN architecture based on VGG 16 architecture
Only 16 layers (13 CONV + 3 FC)

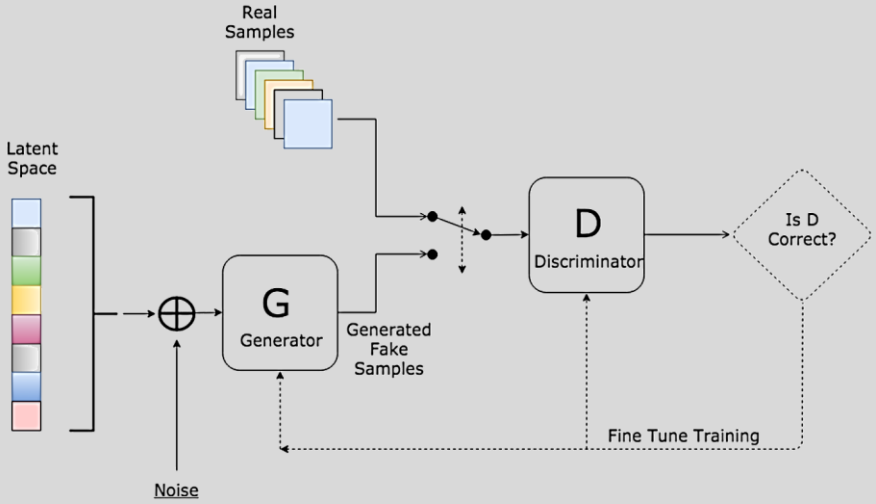
Custom CNN :
Modification of first convolution
Adding a last fully connected



[CNN architecture]

PART2 : DATA AUGMENTATION (GANs)

GOAL : Generate COW and HORSE data with GAN

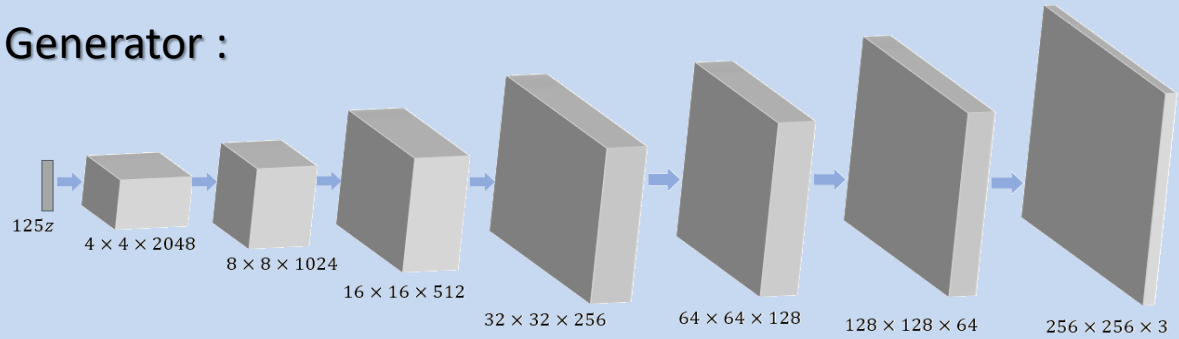


TRAINING DETAILS :

- 5000 epochs (two times)
- Learning rate : 0.0002
- Loss function : binary cross entropy loss (BCE)
- Optimizer : Adam optimizer

SOLUTION : Custom DCGAN architecture to generate 256 x 256 x 3

Generator :



7 layers (convTranspose2d+ BatchNorm2d)

Discriminator :

Resnet 18
Pretrained ImageNet
1 FC layer added

Layer name	Output size	Resnet 18
conv1	128 × 128 × 64	7 × 7, stride 2
conv2	64 × 64 × 64	3×3 mxpl, stride 2 $\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$
conv3	32 × 32 × 128	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$
conv4	16 × 16 × 256	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$
conv5	8 × 8 × 512	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$
avg pool	1 × 1 × 1	7 × 7 average pool
fully connected	1	512 × 1000 × 1

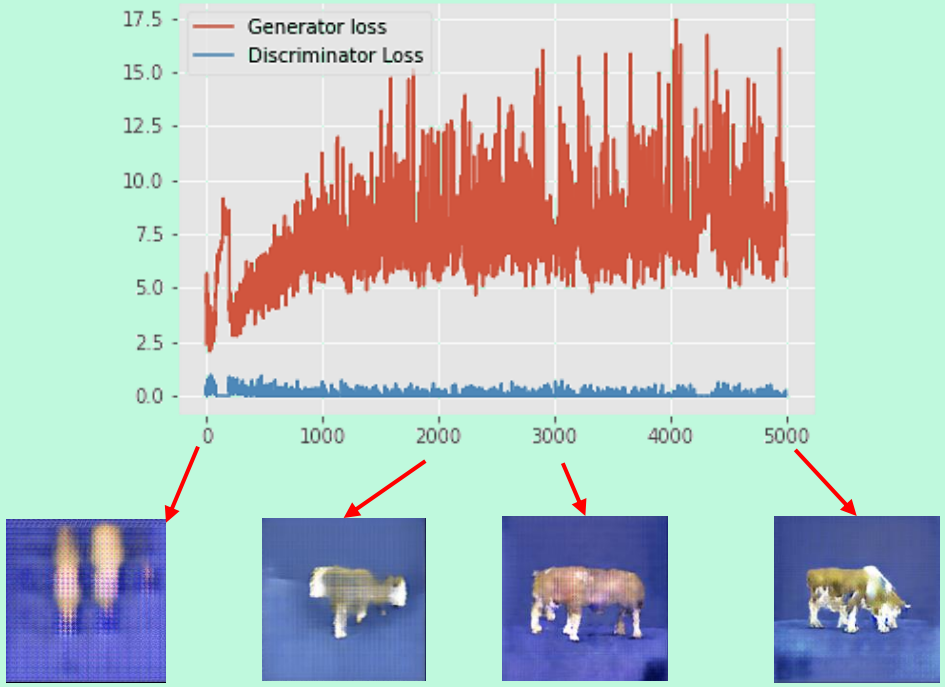
PART2 : DATA AUGMENTATION (GANs)

GOAL : Generate COW and HORSE data with GAN

RESULTS : 40 cow , 58 horses are augmented

Many epochs needed to have good results
Hard for generator to converge
Discriminator loss is too low

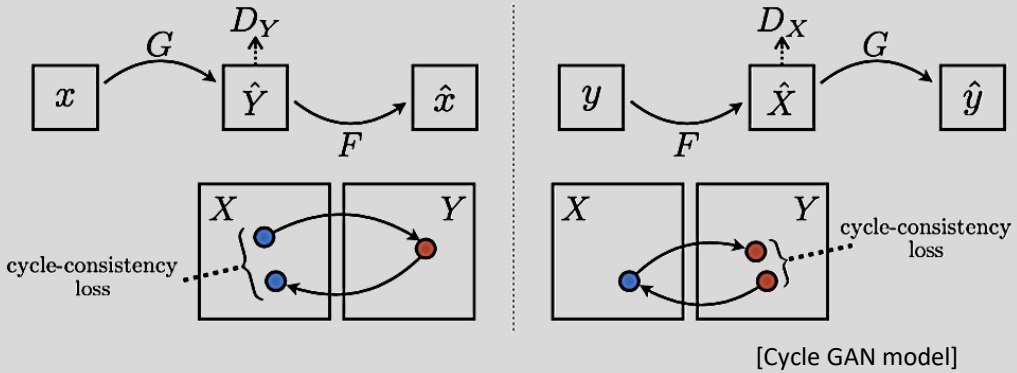
[Loss evolution during COW GAN training]



[Generated images of COW and HORSE]

PART3 : UNPAIRED NEURAL STYLE TRANSFER

GOAL : Generate COW with HORSE's feature and inverse



[Cycle GAN model]

No paired image needed

2 adversarial GAN

2 Generator

G : COW→ HORSE

F : HORSE → COW

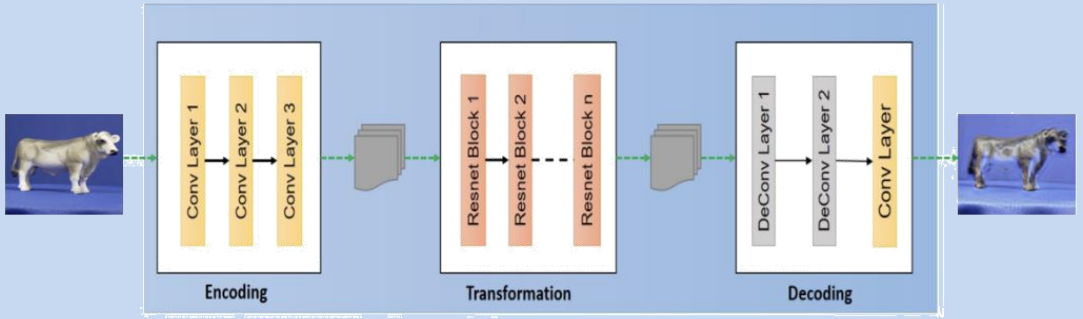
2 Discriminator

Dy : Generated HORSE / real HORSE

Dy : Generated COW / real COW

SOLUTION : CycleGAN architecture for 256 x 256 x 3

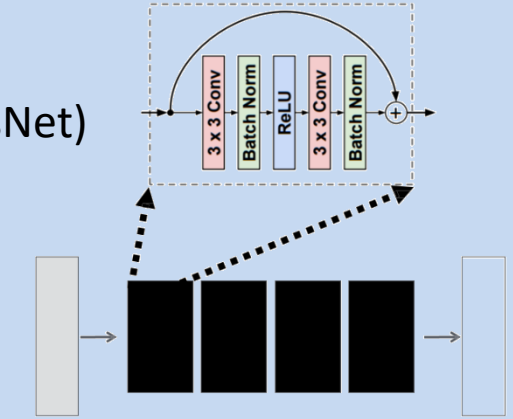
Generator :



Encoding : Downsampling

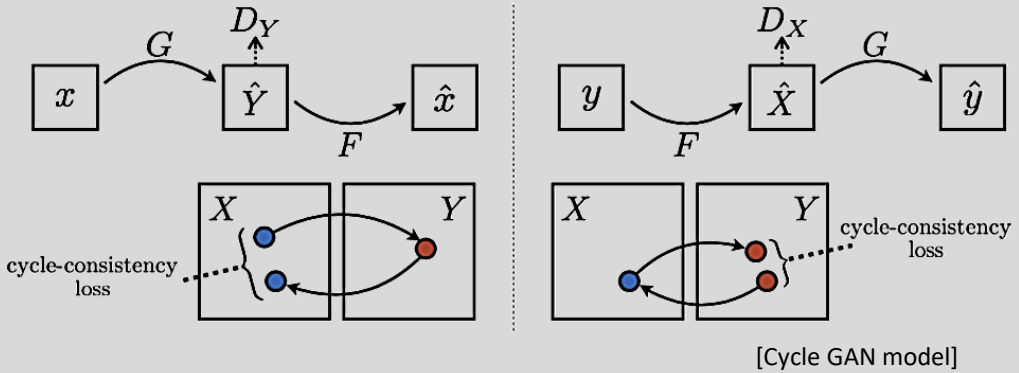
Transformation : 9 residual block (ResNet)

Decoding : Upsampling



PART3 : UNPAIRED NEURAL STYLE TRANSFER

GOAL : Generate COW with HORSE's feature and inverse



[Cycle GAN model]

No paired image needed

2 adversarial GAN

2 Generator

G : COW→ HORSE

F : HORSE → COW

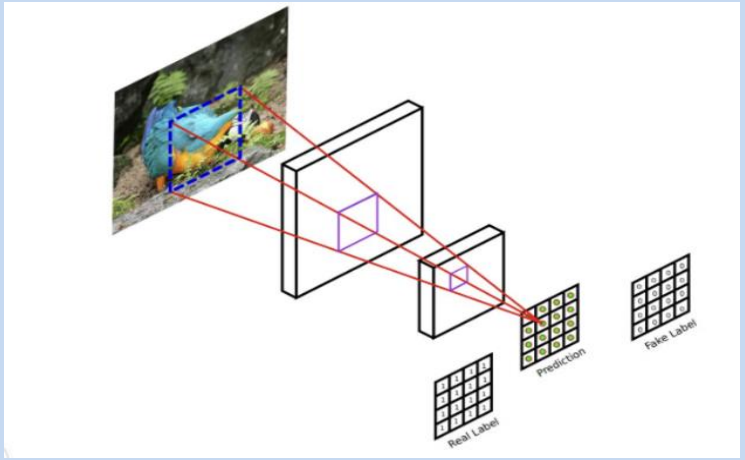
2 Discriminator

Dy : Generated HORSE / real HORSE

Dy : Generated COW / real COW

SOLUTION : CycleGAN architecture for 256 x 256 x 3

Discriminator : Patch GAN



Divide image in many blocks

Judge eachs blocks

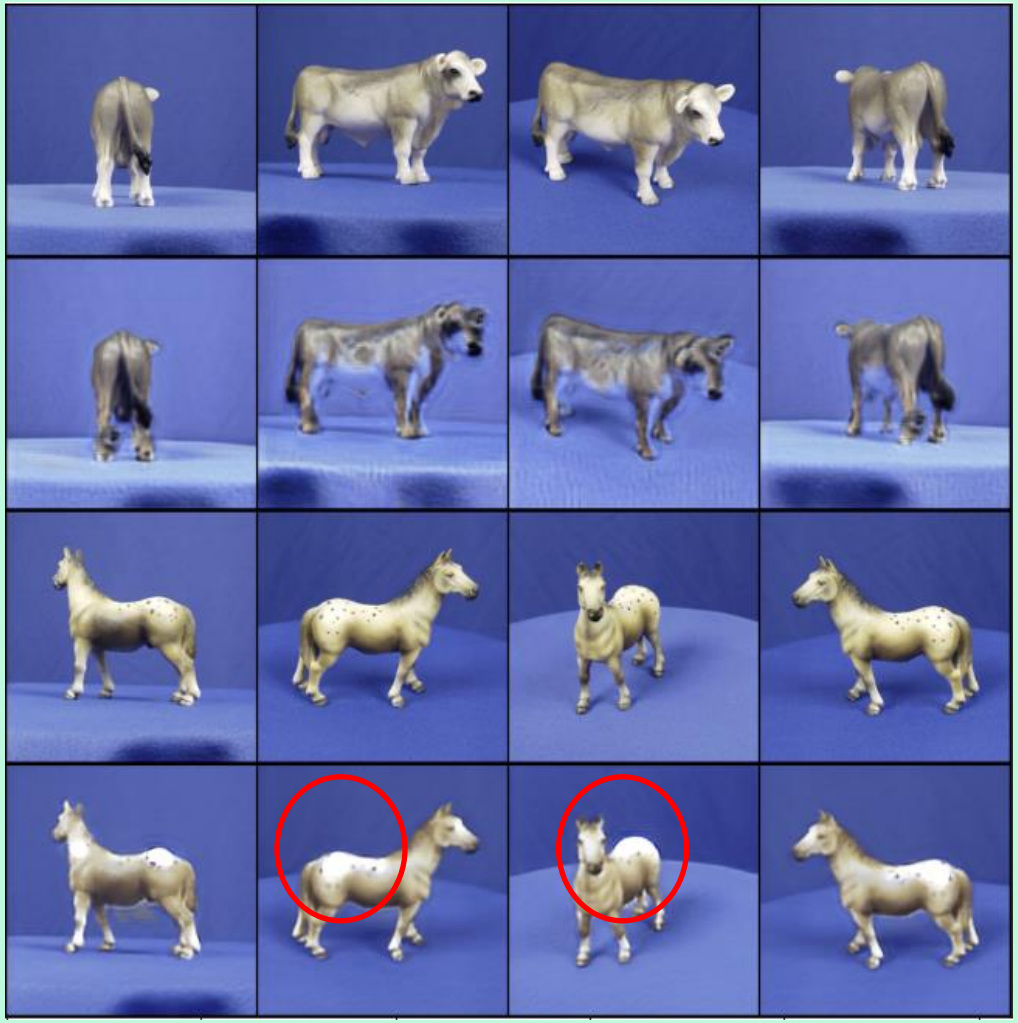
PART3 : UNPAIRED NEURAL STYLE TRANSFER

GOAL : Generate COW with HORSE's feature and inverse

Training details :

- 200 epochs
- Learning rate : 0.0002 and decrease learning rate every 100 epoch
- Loss function = GAN loss + cycle loss + Identity loss
- Optimizer : Adam optimizer
- Replay buffer : We show generated images to discriminator periodically to stabilize training
- Weight initialization with normal distribution : mean 0, std 0.02

RESULTS :



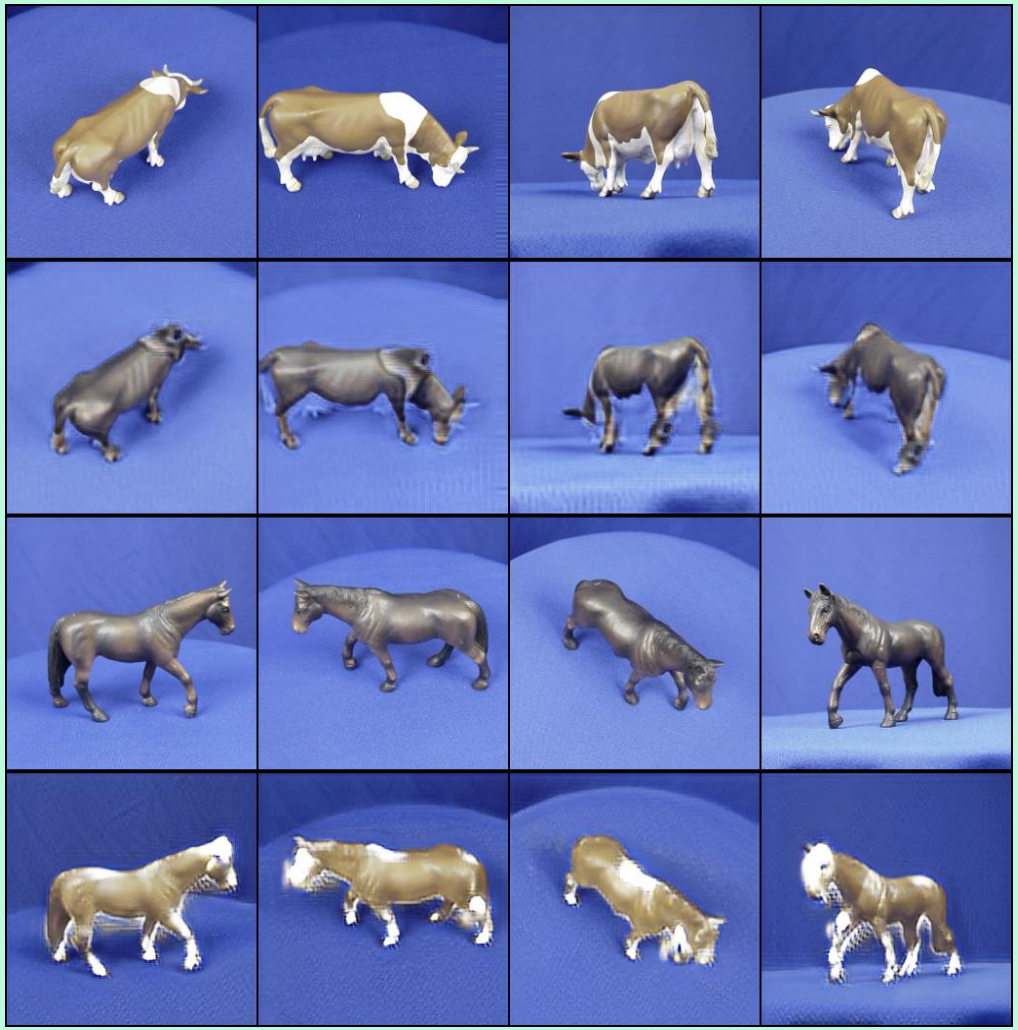
PART3 : UNPAIRED NEURAL STYLE TRANSFER

GOAL : Generate COW with HORSE's feature and inverse

Training details :

- 200 epochs
- Learning rate : 0.0002 and decrease learning rate every 100 epoch
- Loss function = GAN loss + cycle loss + Identity loss
- Optimizer : Adam optimizer
- Replay buffer : We show generated images to discriminator periodically to stabilize training
- Weight initialization with normal distribution : mean 0, std 0.02

RESULTS :



Conclusion

First experience of Python programming

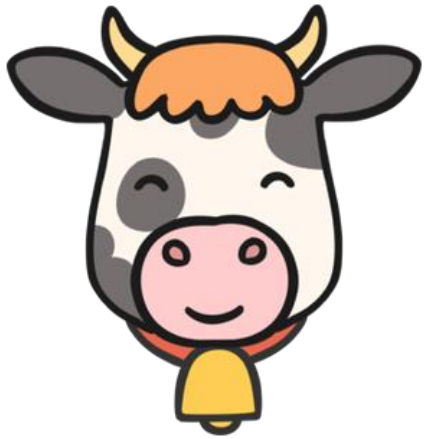
Learned to search Pytorch Library

Introduction into reading scientific paper

Importance of data

Concepts of Neural Network

Importance of hyper parameters



Thank you