# Comparison of Pix2Pix and CycleGAN's performances for illustration colorization

Maria Alba, Berta Benet, Maria Elena Budan

# 1 Introduction

# 1 Introduction

## Image to Image translation

Task of automatically translating images from one domain to another.

*Examples:*

- *Paintings → Images*
- *Edges → Real objects*
- *B&W → Color*

## Comparison

We aim to compare the Pix2Pix and CycleGAN architectures, both provided by Jun-Yan Zhu and Teasung Park.

## Our project

How does each architecture fit our dataset?

We provide a set of experiments and their respective results.

# 1 Colorization

" Technique of adding color to black and white pictures "

2

# Dataset

2

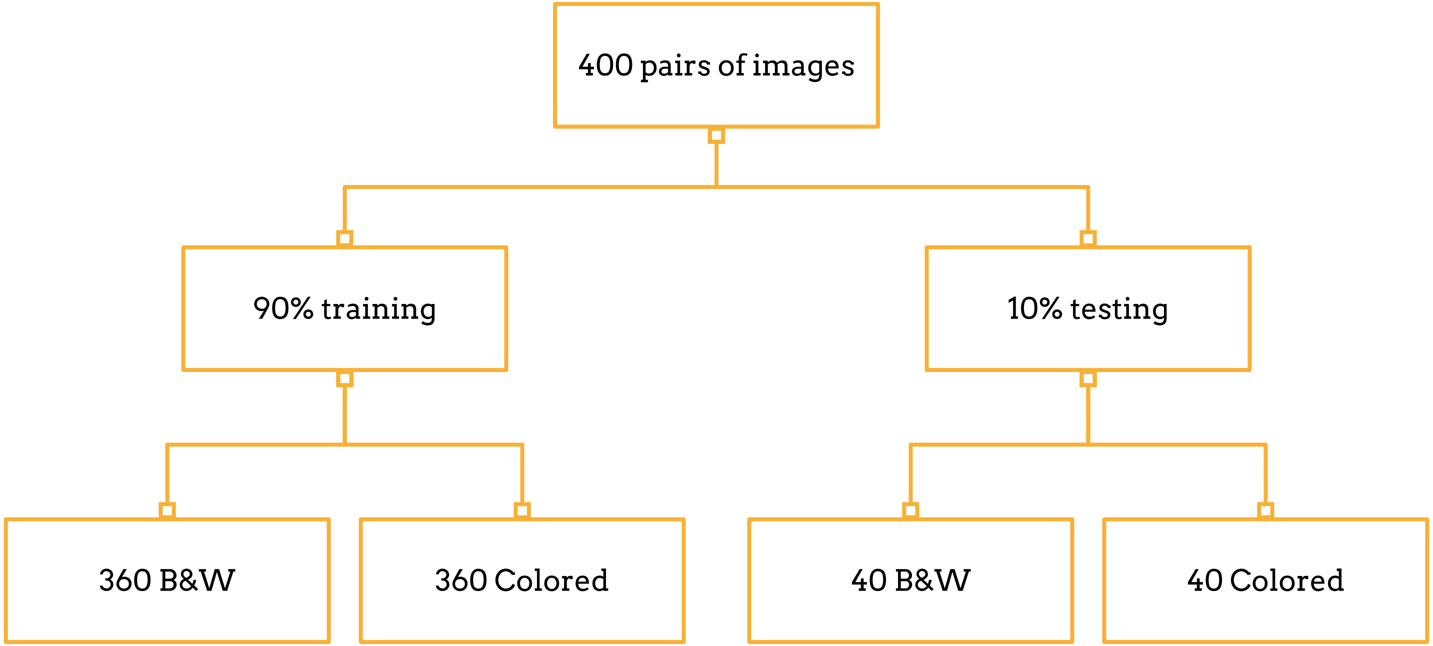
# Dataset

**400** pairs of illustrations from the Victorian era:

1. B&W + Colored
2. Resized versions (256x256)
3. High resolution

4

# Experiments - Dataset

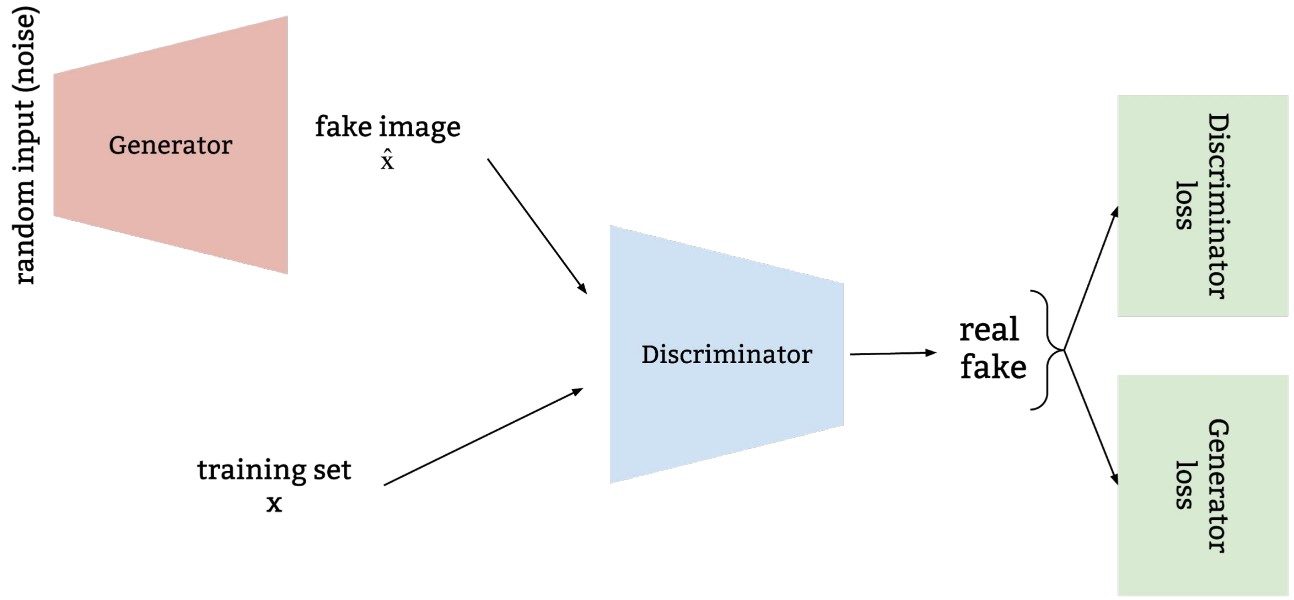


3

# Generative Adversarial Networks (GANs)

3

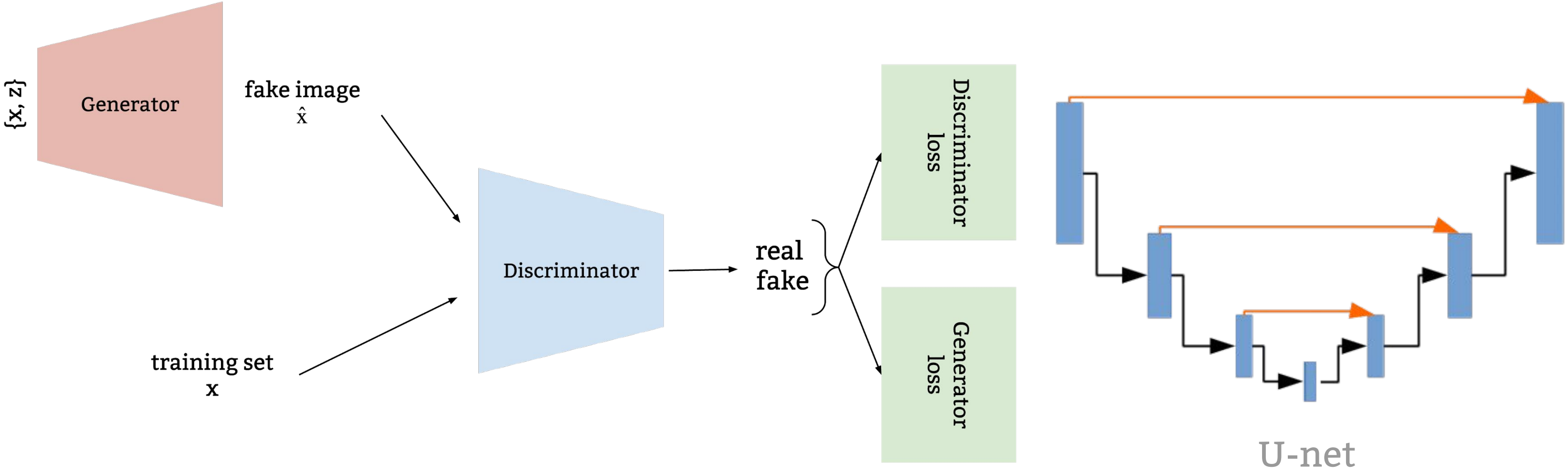
# GAN structure



3

# Pix2Pix architecture

# 3 cGAN structure




Objective function:

Loss of conditional GAN + L1 distance

3

# cGAN objectives

## Objective function

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

### Objective of a conditional GAN

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

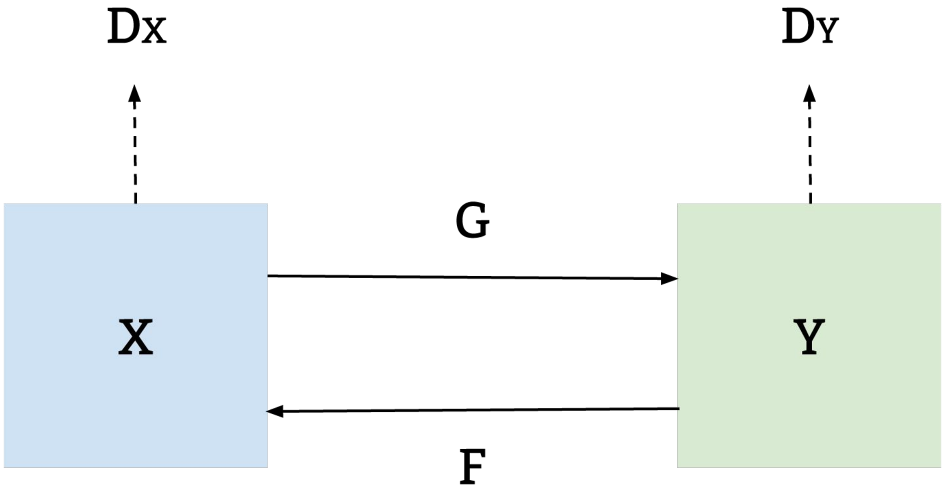
### L1 distance

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

3

# Cycle-GAN

# 3 Cycle-GAN structure




$$F(G(x)) \approx x$$

$$G(F(y)) \approx y$$

Objective function:

Adversarial losses + cycle consistency loss

3

## Cycle-GAN objectives

### Objective function

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

### Loss function

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

3

## Cycle-GAN objectives

### Adversarial losses

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

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

### Cycle consistency loss

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

3

## How they differ?

3

# How they differ?

| Pix2Pix | CycleGAN |
|---|---|

**Pix2Pix**

- **Pairwise domain**

*(Needs paired images from the two corresponding domains.)*

- **Only forward mapping**

**CycleGAN**

- **Non-pairwise domain**

*(Works with unordered image collections.)*

- **Forward & backward mapping**

# 4 Experiments

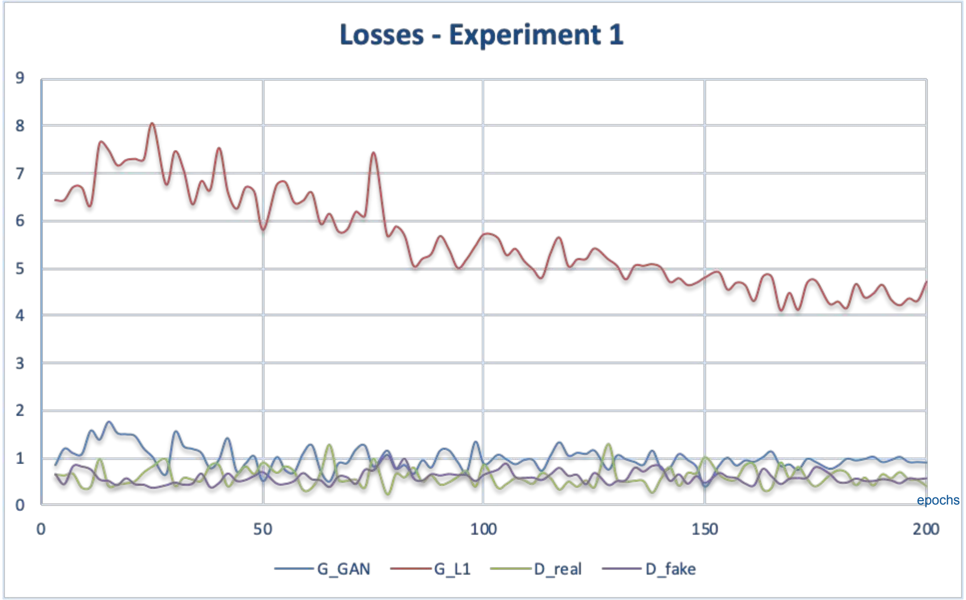
# 4 Experiments

| EXPERIMENT | # 1 | # 2 | # 3 | # 4 |
|---|---|---|---|---|
| Model | Pix2Pix | | CycleGAN | |
| Batch size | 32 | 1 | 1 | 1 |
| Training Dataset | 90%<br>720 paired images | 90%<br>720 paired images | 45%<br>360 unpaired img. | 90%<br>720 images |
| | - 360 colored<br>- 360 B&W model generated | - 360 colored<br>- 360 B&W model generated | - 180 colored<br>- 180 B&W | - 360 colored<br>- 360 B&W |

# 5 Results

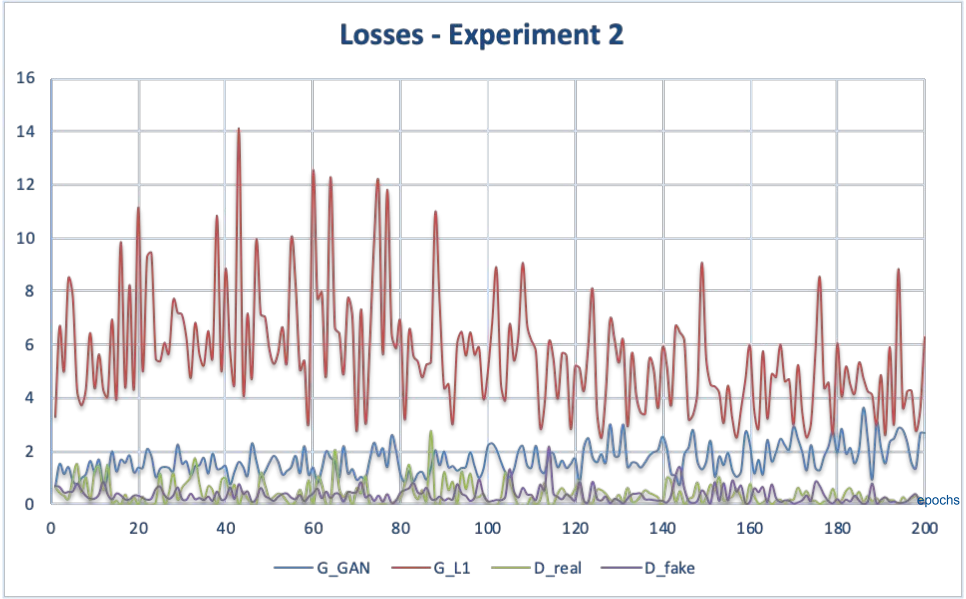
5

# Training - Results



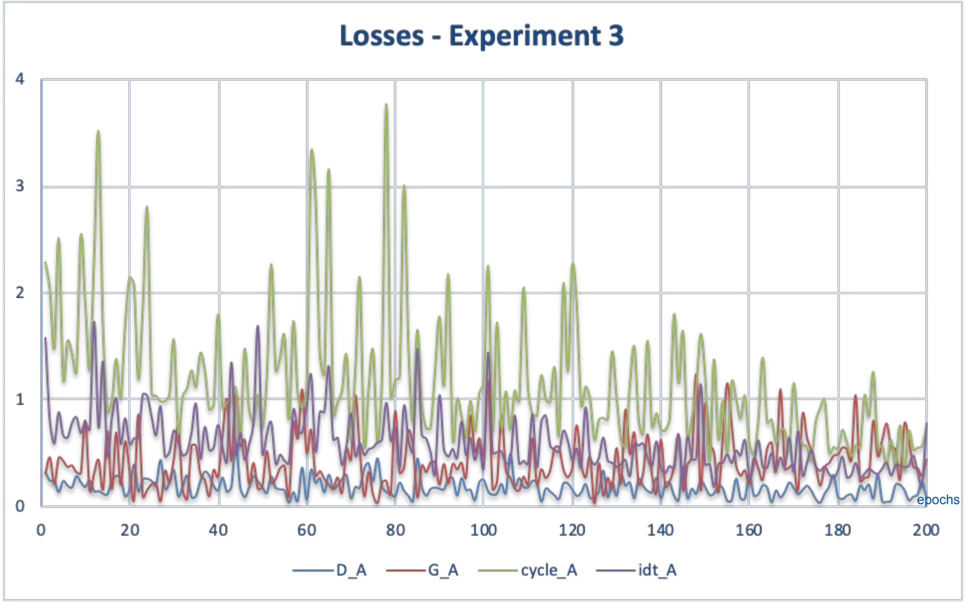

| | | |
|---|---|---|
| Training time | ≃ 1.5h | |
| # Param. Network G | 54.410 M | |
| Losses last iter | G_GAN | 0.89 |
| | G_L1 | 4.721 |
| | D_real | 0.411 |
| | D_fake | 0.565 |

# 5 Training - Results




| | | |
|---|---|---|
| Training time | ≃ 3.5h | |
| # Param. Network G | 54.410 M | |
| Losses last iter | G_GAN | 2.710 |
| | G_L1 | 6.290 |
| | D_real | 0.180 |
| | D_fake | 0.100 |

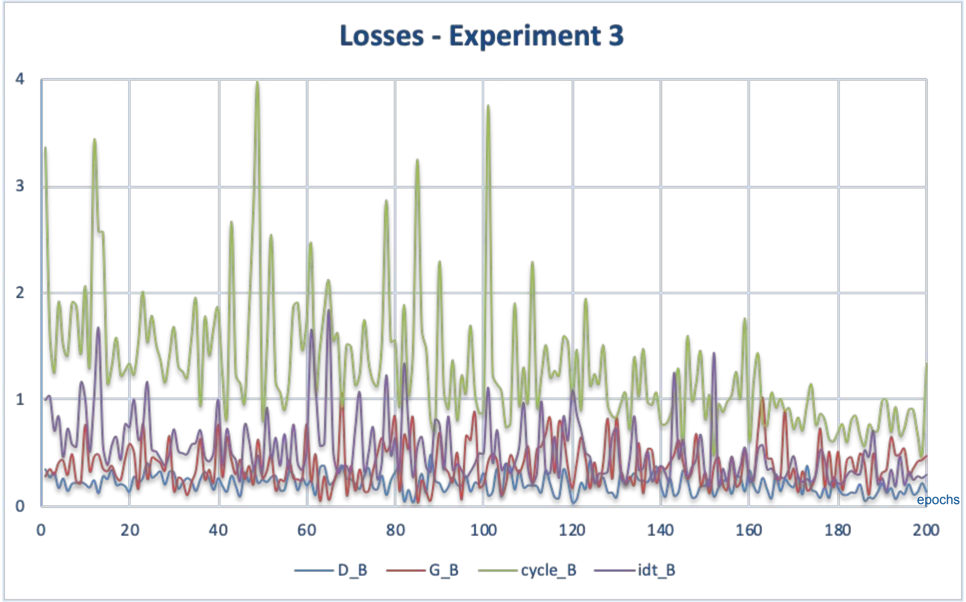
# 5 Training - Results




| | | |
|---|---|---|
| Training time | ≃ 5.7h (for both transf) | |
| # Param. Network G_A | 11.378 M | |
| Losses last iter | D_A | 0.085 |
| | G_A | 0.438 |
| | cycle_A | 0.714 |
| | idt_A | 0.780 |

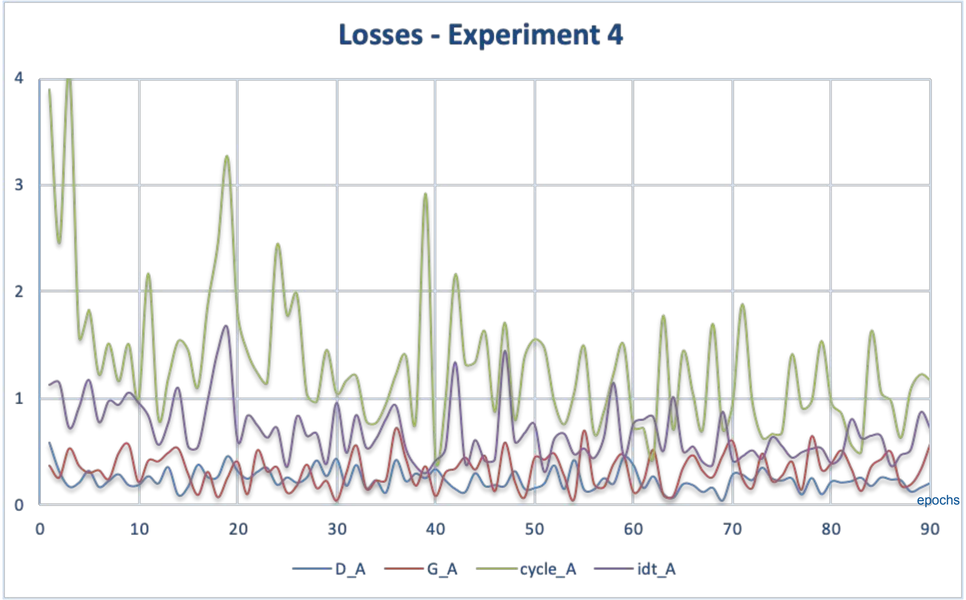
5

# Training - Results



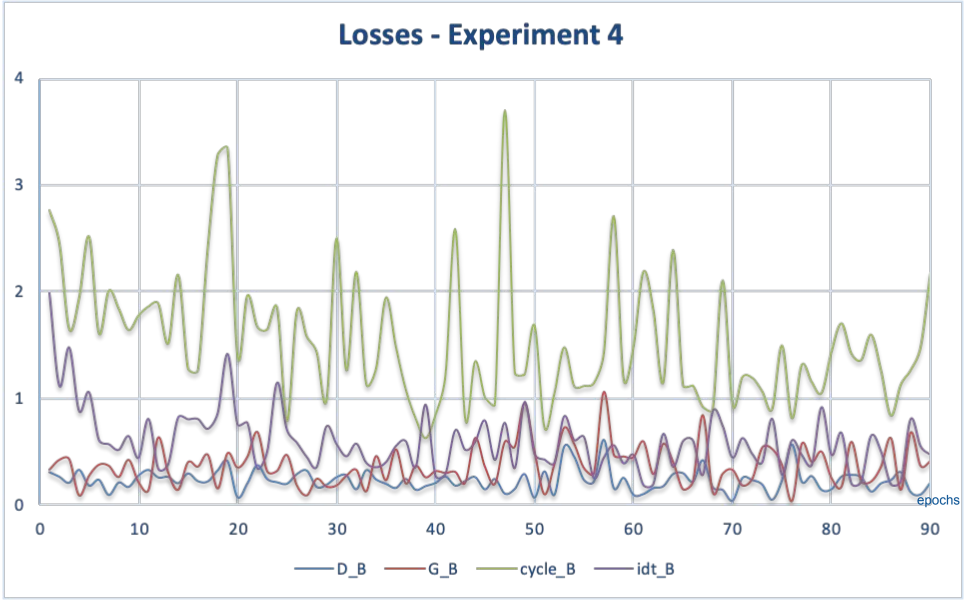

| | | |
|---|---|---|
| Training time | ≃ 5.7h (for both transf) | |
| # Param. Network G_B | 11.378 M | |
| Losses last iter | D_B | 0.134 |
| | G_B | 0.472 |
| | cycle_B | 1.333 |
| | idt_B | 0.290 |

# 5 Training - Results




| | | |
|---|---|---|
| Training time | ≃ 5.4h (for both transf) | |
| # Param. Network G_A | 11.378 M | |
| Losses last iter | D_A | 0.203 |
| | G_A | 0.585 |
| | cycle_A | 1.165 |
| | idt_A | 0.723 |

# 5 Training - Results




| | | |
|---|---|---|
| Training time | ≃ 5.4h (for both transf) | |
| # Param. Network G_B | 11.378 M | |
| Losses last iter | D_B | 0.218 |
| | G_B | 0.408 |
| | cycle_B | 2.180 |
| | idt_B | 0.479 |

5

# Results

## Training time

Same | Batch size
Amount of data
Num. of iterations

Pix2Pix converges faster but learns only one transformation

## Qualitative testing

- AMT

5

# Testing - Results

## Original pair

# 5 Testing - Results

## Pix2Pix

Original B&W

Experiment 1

Experiment 2

Original Color

Only difference between Experiment 1 and Experiment 2 is the batch size

# 5 Testing - Results

| CycleGAN: B&W → Color | Fake colored | Reconstructed |
| --- | --- | --- |

Experiment 3

Experiment 4

# 5 Testing - Results

| CycleGAN: Color → B&W | Fake B&W | Reconstructed |
| --- | --- | --- |
| Experiment 3 |  |  |
| Experiment 4 |  |  |

# 5 Testing - Results

Experiment 1

Experiment 2

Experiment 3

Experiment 4

# 6 Conclusions

# 6 Conclusions

Original | Experiment 1 | Experiment 2 | Experiment 3 | GT

# 7 References

# 7 References


[1] Hoyeol, K. (2019). Victorian400. Retrieved from https://www.kaggle.com/elibooklover/victorian400

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, & Alexei A. Efros. (2018). Image-to-Image Translation with Conditional Adversarial Networks. Retrieved from https://arxiv.org/pdf/1611.07004.pdf

[3] Jun-Yan Zhu, Taesung Park, Phillip Isola, & Alexei A. Efros. (2018). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. Retrieved from https://arxiv.org/pdf/1703.10593.pdf

[4] junyanz/pytorch-CycleGAN-and-pix2pix. (2018). Retrieved from https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix

**Our repository:**
Maria Alba, Berta Benet, Marilena Budan (2020). Comparison of Pix2Pix and CycleGAN's performances for illustration colorization. https://github.com/marilenabudan/Colorization_Pix2Pix_CycleGAN

# Thank you for listening

## Any questions?