JoJoGAN: One Shot Face Stylization

Mattia Pannone

SAPIENZA
UNIVERSITÀ DI ROMA

Project for Neural Networks Exam

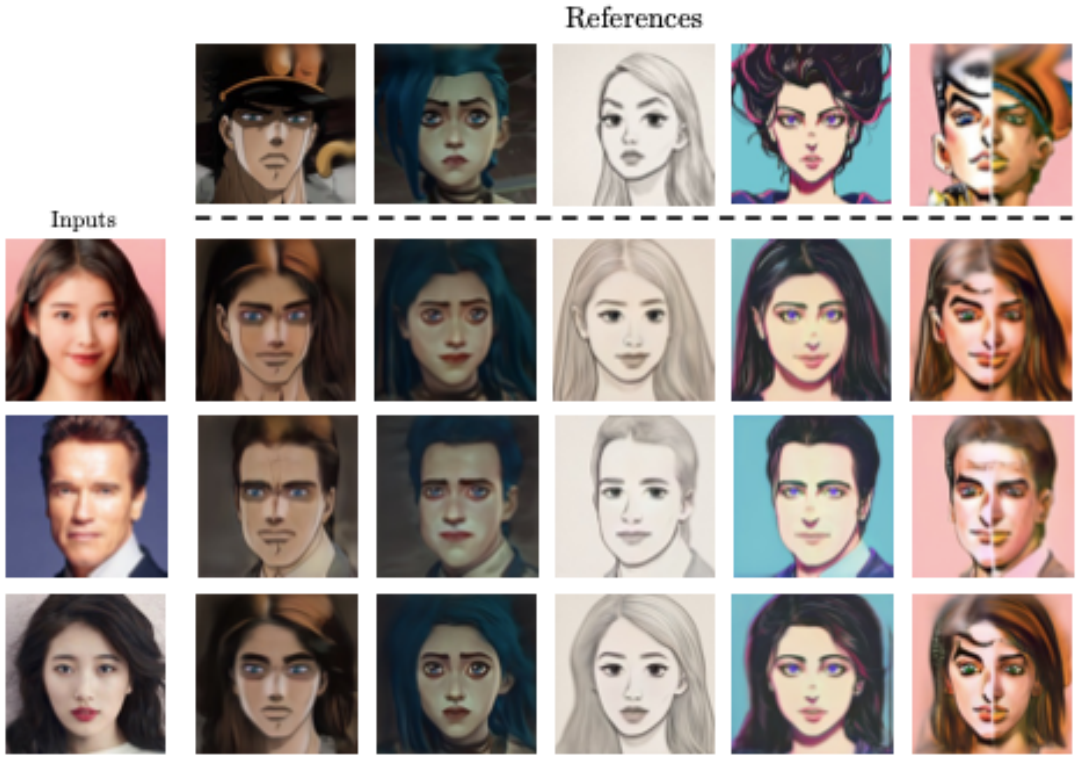
SAPIENZA
UNIVERSITÀ DI ROMA

Introduction and purposes:

- Transfer the style from a reference style image to another input image
- Learn a style mapper
- Use of GANs, (StyleGAN and Encoding for Editing)
- Use 4 steps to produce good results

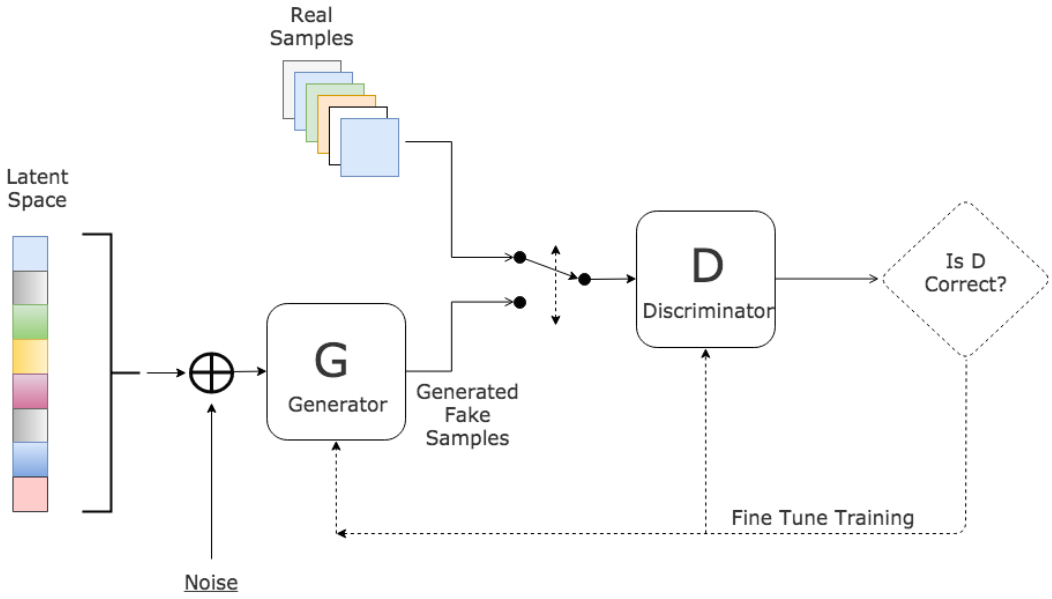
SAPIENZA
UNIVERSITÀ DI ROMA

Introduction and purposes:

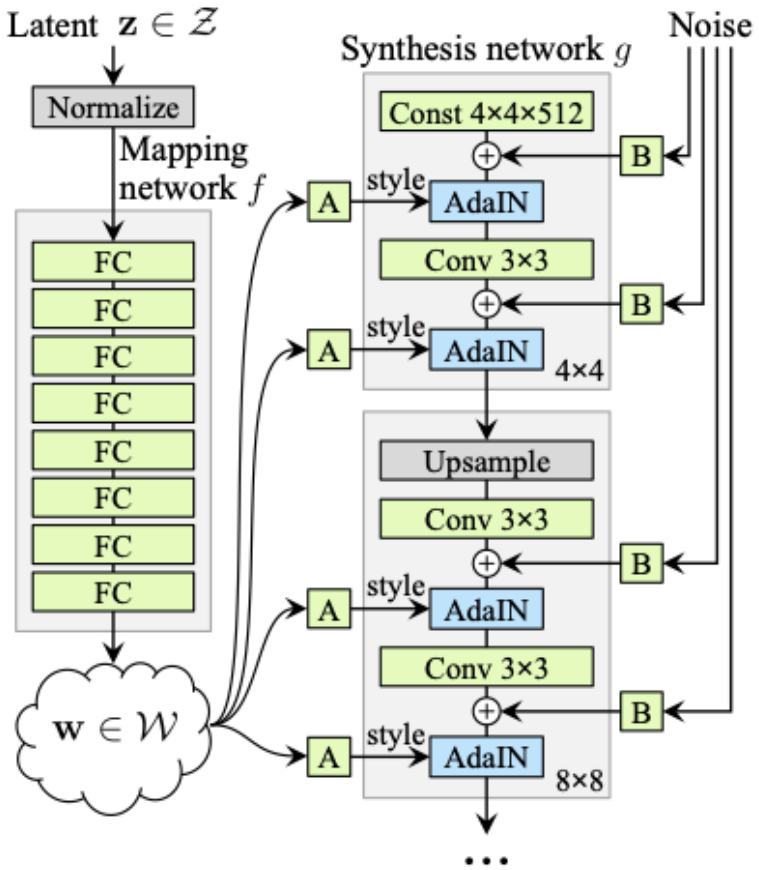


What is a GAN?

- Generative Adversarial Network
- Produce new data as much as possible to real data learning their distribution
- Made by 2 nets: Generator and Discriminator

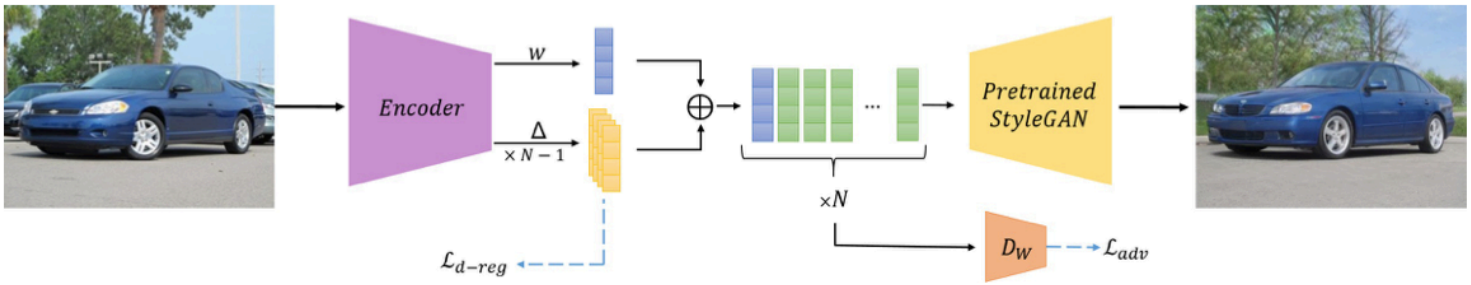


StyleGAN



- Styled-based GAN architecture
- Has been used as a prior for numerous tasks such as super-resolution and face restoration
- Produce styles that control the layers of the synthesis network via adaptive instance normalization (AdaIN)
- JoJoGAN creates a paired dataset from a single style reference by manipulating a pre-trained StyleGAN2

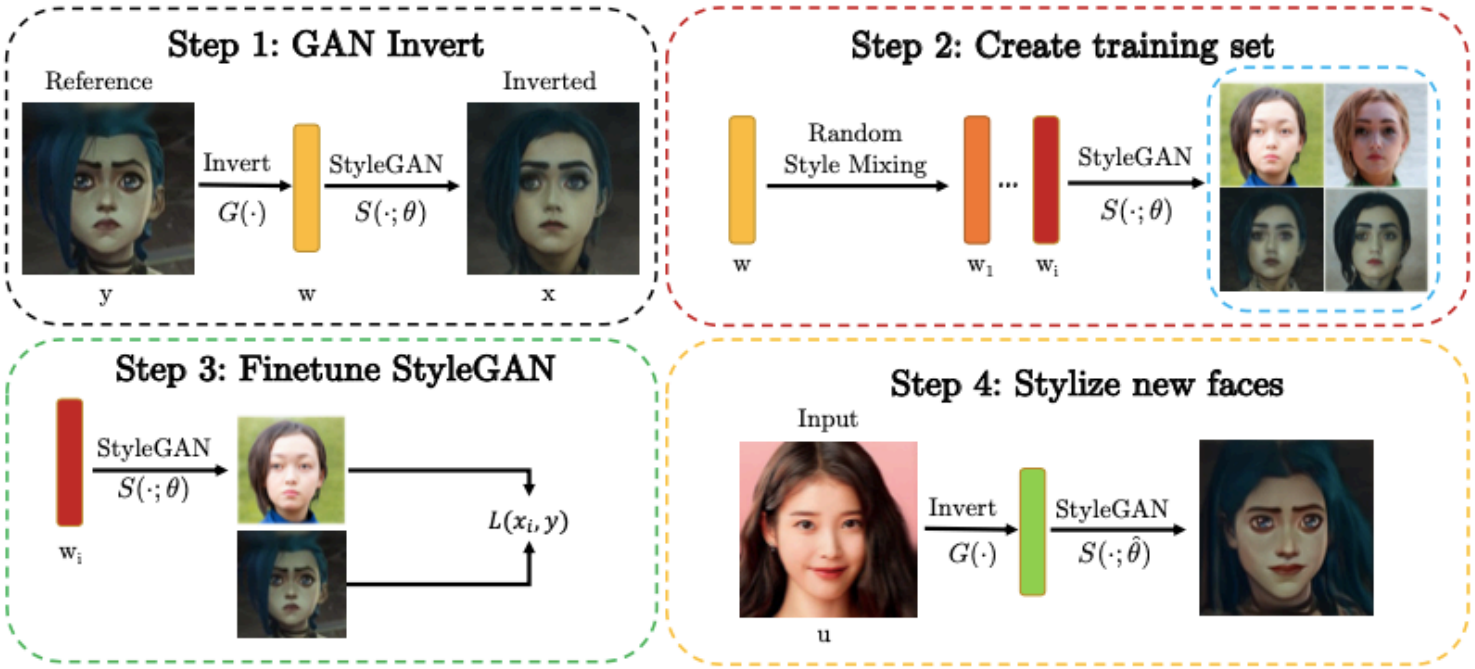
Encoder for Editing (e4e)



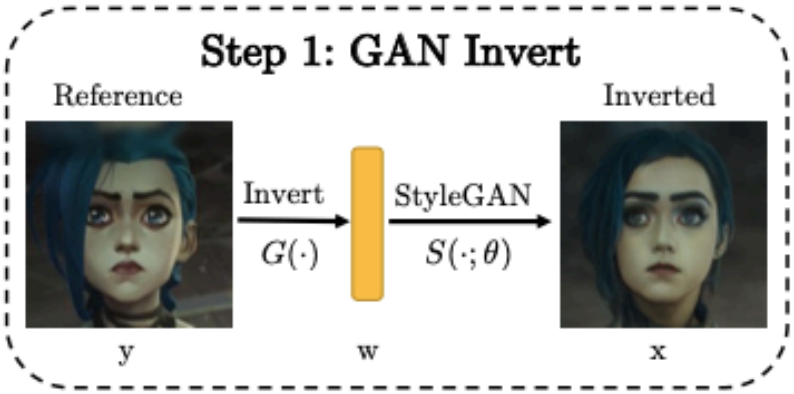
- Used for GAN inversion
- Pre-trained encoder that is specifically designed to allow for the subsequent editing of inverted real images

1. The encoder receives an input image and outputs a single style code w together with a set of offsets
2. Applying regularization terms, the encoder's final learned representation lies close to W

Steps of JoJoGAN

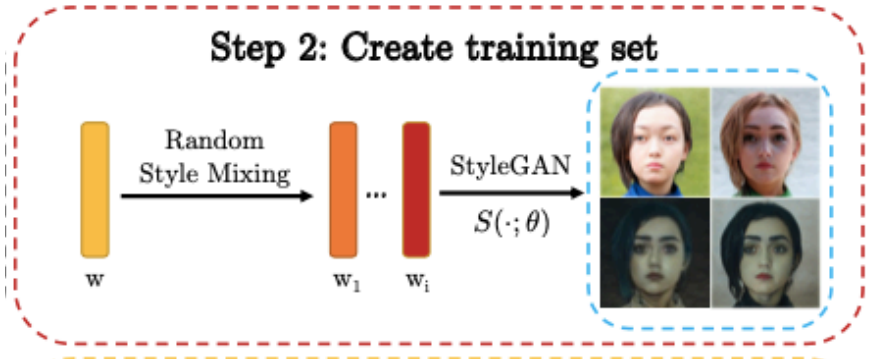


1. GAN Inversion



- Produce style code “w” of a reference image
- GAN inverter is trained to produce codes that result in realistic faces
- Given w, StyleGAN will generate a plausible real face image
  $x = S(w; \theta)$

2. Training set

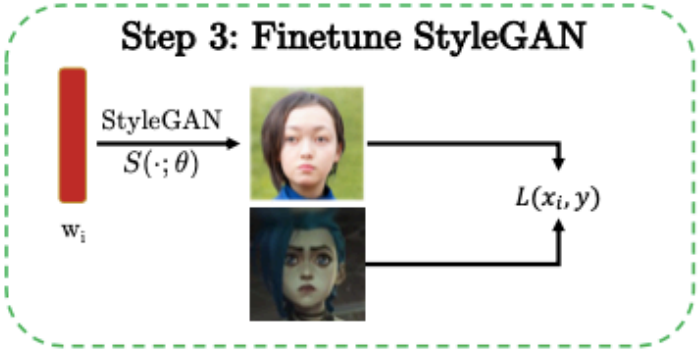


- Use w to find a set of style codes W that are "close" to w
- New style codes by:

$$w_i = M \cdot w + (1 - M) \cdot FC(z_i)$$

- Pairs (w_i, y) for $w_i \in$ W will be our paired training set

3. Fine-tune StyleGAN



- Finetune StyleGAN to obtain obtain $\hat{\theta}$ such that $S(w_i; \hat{\theta}) \approx y$

$$\hat{\theta} = \underset{\theta}{\text{argmin}}\, \text{loss}(\theta) = \underset{\theta}{\text{argmin}}\, \frac{1}{N} \sum_{i}^{N} \mathcal{L}(S(w_i;\theta), y)$$

- Perceptual loss: LPIPS, built on a VGG backbone

$$\mathcal{L}(S(w_i;\theta), y) = \| D(S(w_i;\theta)) - D(y) \|_1$$

- Choose use the difference in discriminator activations at particular layers

4. Inference

- For input u, our stylized face is S(G(u); θˆ)
- Can generate random stylized samples by sampling random noise

SAPIENZA
UNIVERSITÀ DI ROMA

My personal work

- Original Material available on git hub
- I proceed to re-organize functions
- Reproduce results with pre-trained models
- Trained new models

SAPIENZA
UNIVERSITÀ DI ROMA

My personal work

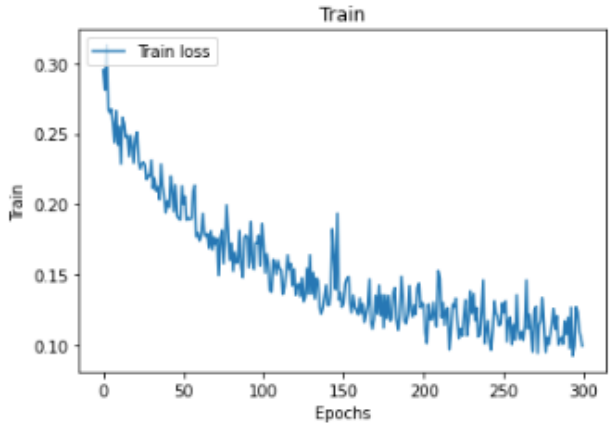
- load_generator() and load_discriminator()
- elaborate_image()
- compute_output_on_given_image() and compute_output_on_random_images()
- load_new_reference ()
- train_on_new_reference()
- plot_train_trend()

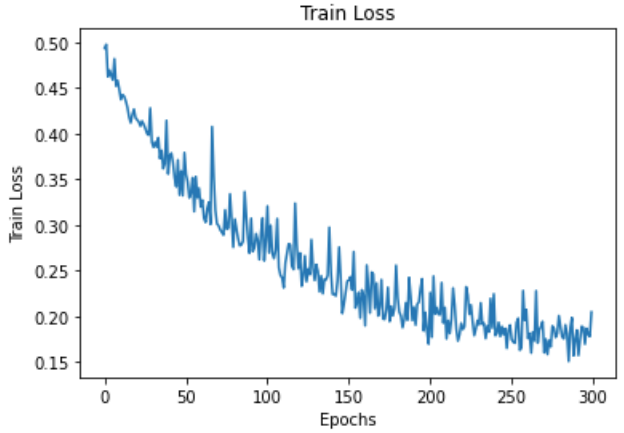
SAPIENZA
UNIVERSITÀ DI ROMA

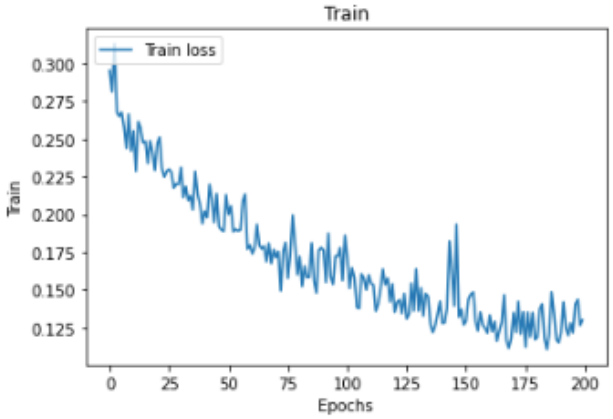
My personal work

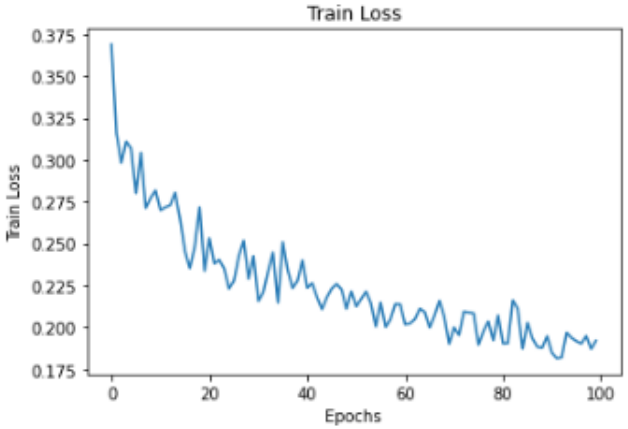
- New (one-shot) trainings with two new reference style images
- 300 epochs for more than 3 hours
- One training using two images of two similar styles at the same time
- Varying the alpha parameter (interpolation factor) used in the formula $f = (1 - \alpha)f_i^A + \alpha f_i^B$

Results

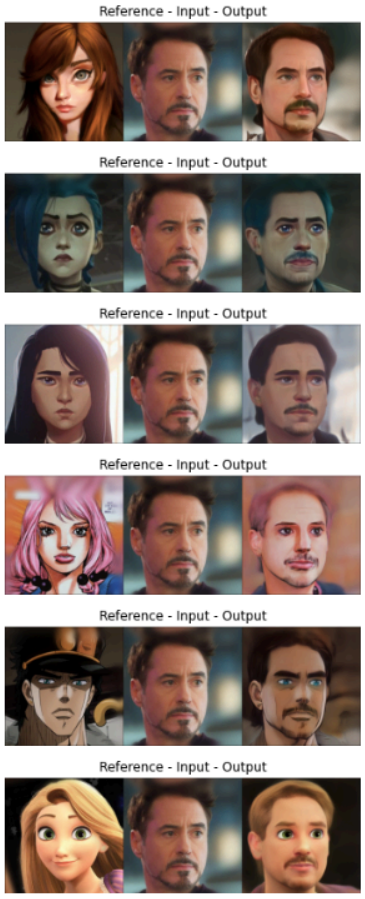

Train
Train loss
Train
Epochs
0.30
0.25
0.20
0.15
0.10
0
50
100
150
200
250
300

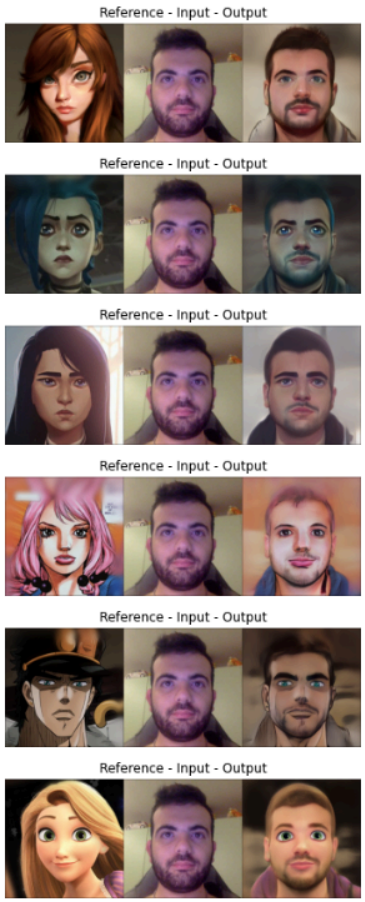

Train Loss
Train Loss
Epochs
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0
50
100
150
200
250
300


Train
Train loss
Train
Epochs
0.300
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0
25
50
75
100
125
150
175
200


Train Loss
Train Loss
Epochs
0.375
0.350
0.325
0.300
0.275
0.250
0.225
0.200
0.175
0
20
40
60
80
100

Results (from pre-trained models)

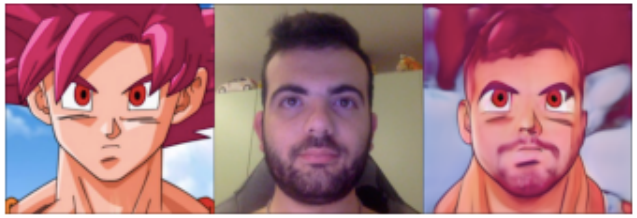




Results (new training)

Reference - Input - Output

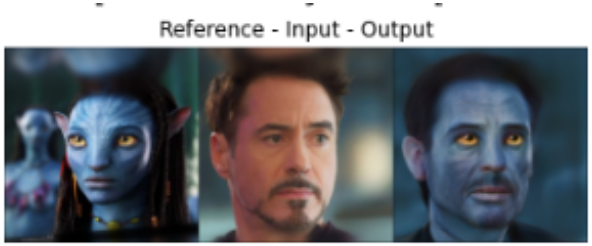
Reference - Input - Output

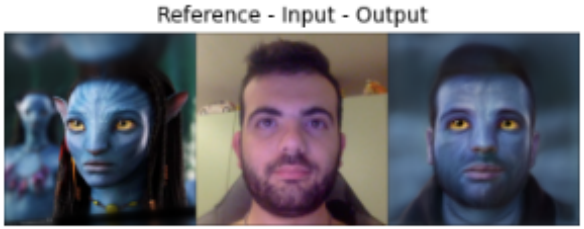


Input & Output

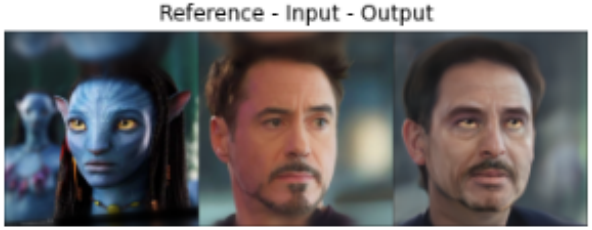
Results (new training)

Alpha = 1



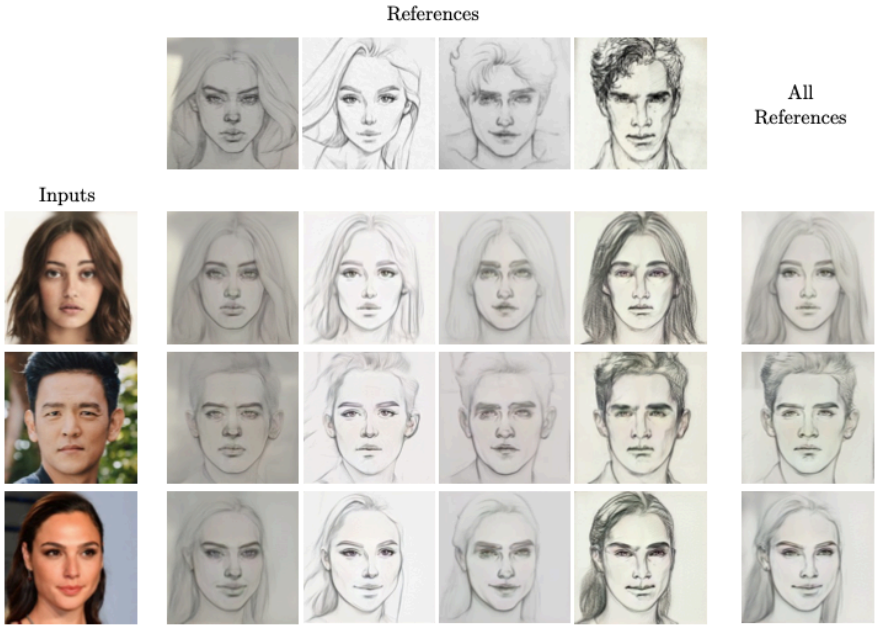


Alpha = 0

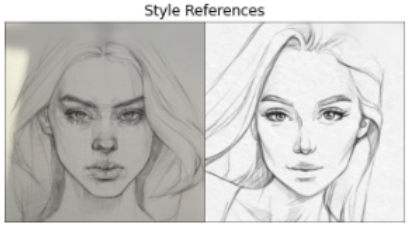


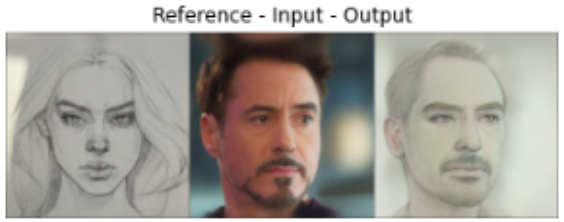
Results (new training)

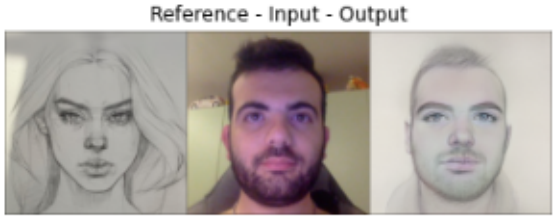
Original training

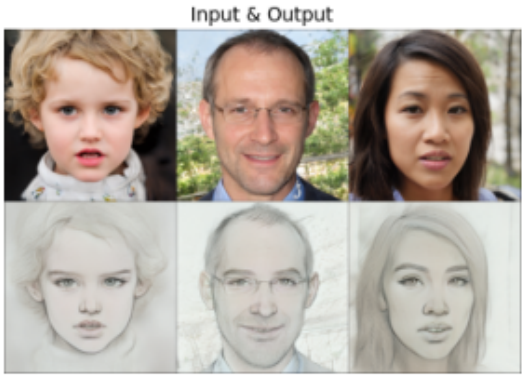


My training





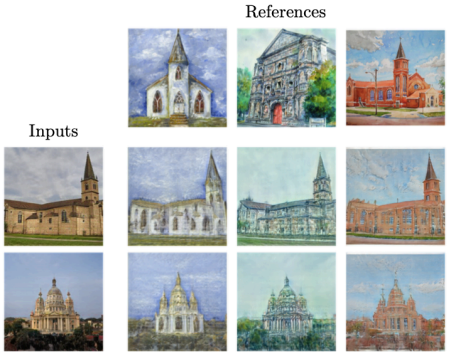




SAPIENZA
UNIVERSITÀ DI ROMA

Conclusions

- JoJoGAN is very competitive with respect to the state of the art in face stylization
- Produce extremely high quality images that get the style details right
- It is capable to produces smooth and consistent stylization as the face moves and changes expressions.
- It is possible to train a model on a different domain rather than faces and it works with same good results



SAPIENZA
UNIVERSITÀ DI ROMA

References

- Min Jin Chong and D.A. Forsyth: JoJoGAN: One Shot Face Stylization
- Ian J. Goodfellow, Jean Pouget-Abadie∗, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair†, Aaron Courville, Yoshua Bengio‡: Generative Adversarial Nets
- Tero Karras, Samuli Laine, Timo Aila: A Style-Based Generator Architecture for Generative Adversarial Networks
- Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, Daniel Cohen-Or: Designing an Encoder for StyleGAN Image Manipulation
- Tengfei Wang, Yong Zhang, Yanbo Fan, Jue Wang, Qifeng Chen: High-Fidelity GAN Inversion for Image Attribute Editing
- Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, Timo Aila: Analyzing and Improving the Image Quality of StyleGAN
- Antonio Gulli, Amita Kapoor, Sujit Pal : Deep Learning with Tensorflow 2 and keras

JoJoGAN: One Shot Face Stylization

Mattia Pannone

Thanks for your attention!!

SAPIENZA
UNIVERSITÀ DI ROMA

Project for Neural Networks Exam