Image inpainting using the latent space of StyleGAN

Tutor: Alasdair Newson (alasdair.newson@telecom-paris.fr)

1 Introduction

Inpainting is the process of automatically filling in a region in an image with content that seems visually plausible. Some of the main challenges are consistency with the border content, and correct structure and texture reconstruction. An example of such an inpainting can be seen in Figure 1:

Figure 1: **Example of image inpainting**.

In the lesson, we have studied Generative Adversarial Networks (GANs), and in particular we mentioned the "StyleGAN" [1], which is a recent network which produces good results on high-resolution images. The latent space of this GAN is very powerful, since it can represent images in a much more compact manner than does the image space. Therefore, in this project, we propose to use StyleGAN2's [2] latent space for inpainting. Careful, there are several StyleGANs, we propose to use StyleGAN2 [2] or StyleGAN3 [3]. A further remark: StyleGAN uses an intermediate latent space, denoted $\mathcal{W}$, which is put into the generator in a parallel fashion to produce the output image (see [1]). Therefore, we refer to a latent code here as w, instead of z.

The general idea is to solve the following inverse problem:

$$\arg\min_{w} \|(x - G(w)) \odot (1 - M)\|_2^2, \tag{1}$$

where G is the StyleGAN generator, x is the image to inpaint, and M is the inpainting mask (or occlusion), which is an image equal to 1 inside the hole to inpaint, and 0 outside. The $\odot$ operation represents element-wise multiplication. Thus, we try and find a w such that, when we generate from w using G, we have an image which looks like the input image (in the ℓ^2 sense), in the unoccluded region.

This problem can be tackled by gradient descent, using autodifferentiation with Pytorch tools (or Tensorflow if you wish). However, there is a first question of initialisation, ie how to find an initial point w. For this, you have two main options:

- Random initialisation;
- Initialise the point using an *encoder* to the latent space of StyleGAN2 or StyleGAN3. These encoders exist already trained, **do not** try and train one yourself.

You can try both appproaches and compare the results.

Once you have done these steps, you should carry out experiments to determine the advantages and in particular the disadvantages of this approach. In particular, we are interested in the following aspects:

- Robustness of the algorithm: what happens if we shift the image a bit, or change the hole slightly?
- Quality of the inpainting inside: regular, noisy, blurry?

1.1 Regularisation and/or semantic constraints

To go further in the project, we can modify the loss function to add various regularisations. At this point, you can choose one of two options (or both if you have time) to study:

- Regularisation to improve quality and robustness of inpainting. For example, can you propose a regularisation which reduces blurring ?
- Semantic constraints. Often, although we do not know the exact content of a region to inpaint, we have some idea of the semantic content; for example, I would like to inpaint this face, under the constraint that it is smiling. To do this, you can add a regularisation term. For this option, you will have to identify a pre-trained classifier for the images which you are considering (eg FFHQ or Celeb-A).

2 Tasks of the project

- Study and understand StyleGAN2 or StyleGAN3, and download the code and get it to work. There is at least one StyleGAN2 version which works easily: StyleGAN2-Ada, which has a good Pytorch implementation `https://github.com/NVlabs/stylegan2-ada-pytorch`;
- Potentially find an *encoder* for StyleGAN2/3. We can help you with this.
- Carry out optimisation-based inpainting in $\mathcal{W}$;
- Carry out and analyse experiments to identify disadvantages of this approach, potentially propose regularisation of loss to address these disadvantages;
- Using a pre-trained classifier, modify the loss function to add a semantic constraint to the inpainting problem;

References

[1] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020.

[3] Yuval Alaluf, Or Patashnik, Zongze Wu, Asif Zamir, Eli Shechtman, Dani Lischinski, and Daniel Cohen-Or. Third time's the charm? image and video editing with stylegan3. In *European Conference on Computer Vision*, pages 204–220. Springer, 2022.