भारतीय प्रौद्योगिकी संस्थान दिल्ली
Indian Institute of Technology Delhi

Encoding in Style: A StyleGAN Encoder for Image-to-Image Translations

AMBRISH KASHYAP (2019CH70157)

April 2022

Keywords

- *pixel2style2pixel(pSp)*
- Generative Adversarial Networks (GAN)
- StyleGAN
- Latent Space $W+$
- Generator

Abstract

A new framework is proposed in this paper for image to image translations which is named as *pixel2style2pixel(pSp)*. This pSp framework differs from the conventional method of inverting first and then editing later. It is based on an encoder network. The encoder network outputs a series of Style Vectors, which is then directly used as an input for a pre-trained StyleGAN-generator. This paper further shows that using pSp framework also simplifies the process of image-to-image translations by reducing the computational time. Since this framework is not based upon pixel-to-pixel model so this framework is applicable globally to a lot of image translation problems. Some applications of pSp model have been shown in this paper and the results are compared with the state-of-art models. These results clearly show that pSp is a superior method than all other present methods for many specific tasks.

Introduction

GANs are a type of deep learning neural network which is used widely these days for a number of purposes. The most common use of GANs are in the field of image translations - which can include a number of tasks like generating new human images, generating images from sketches, aging of face, super resolution, face frontalization etc. A GAN basically consists of 2 neural networks namely Generator and Discriminator. The core idea of GAN is on the base that the generator will generate random things and it will be passed on to discriminator. The discriminator would be a trained neural network which could classify the things into real or fake. After training GAN for some time, the generator starts generating images which are actually fake but are classified as real by the discriminator. The Generator would be trained until it learns to fool the discriminator.

StyleGANs are a further extension to the GAN model architecture. It has following three added features:

1. StyleGANs are *progressive growing networks* - StyleGANs increase the size of generated image gradually. They start with generating a very low resoultion image and then increase it to high resolution image. StyleGANs use bi-linear sampling for up-scaling the size of images.

2. StyleGANs use *noise mapping* network - Generator start with a random noise. As we go deeper into the network, its role decreases in the output which reduces the quality of result. In StyleGANs we use noise mapping network to pass the noise to each convolutional layer inside the Generator architecture.

3. StyleGANs use *Adaptive Instance Normalization* - Adaptive Instance Normalization is done for each input in a convolutional layer. This increases the uniformity in input and increases the result quality.

Problem-Formulation

StyleGANs today have achieved very high quality on high resolution images. It also offers a latent space $W+$ which is disentangled. This feature gives additional editing controls over images. Recently, there has been a lot of development in trying to control the manipulations on the latent space to produce desired results. Majority of these methods use the technique of inverting the image first and then editing it. But there still lies a challenge while using this method. When we are inverting the real image to form a 512-dimensional vector $w \in W$, then this does not give us the exact reconstruction of the image.

To solve this challenge, some methods have started to encode the real image to a vector w which has a dimension of 18x512, one vector row for each input layer of StyleGAN. This improves the quality of reconstructed image but is computationally very expensive. Some optimization methods have been used to accelerate this process but they are not very effective yet.

This paper introduces an encoder architecture which encodes any random real image to the extended vector space *W+*. Our encoder is based upon a Feature Pyramid Network(FPN). FPN works as a feature extractor which takes a random real image as an input and outputs feature maps in a convolutional fashion. These extracted values are directly given as input to the generator of our model. This enables our encoder to directly create a reconstruction of image without much calculation or using any optimization.

This encoder gives us a lot of space to do manipulations on the latent space. But the prerequisite of this encoder is that the images which we are giving as input must be invertible which means that a code must be present to reconstruct the image back. To solve this problem we use a pretrained StyleGAN generator along with our encoder. This provides a complete framework for the image-to-image translations as here the input images are encoded from our encoder and then given as input to StyleGAN.

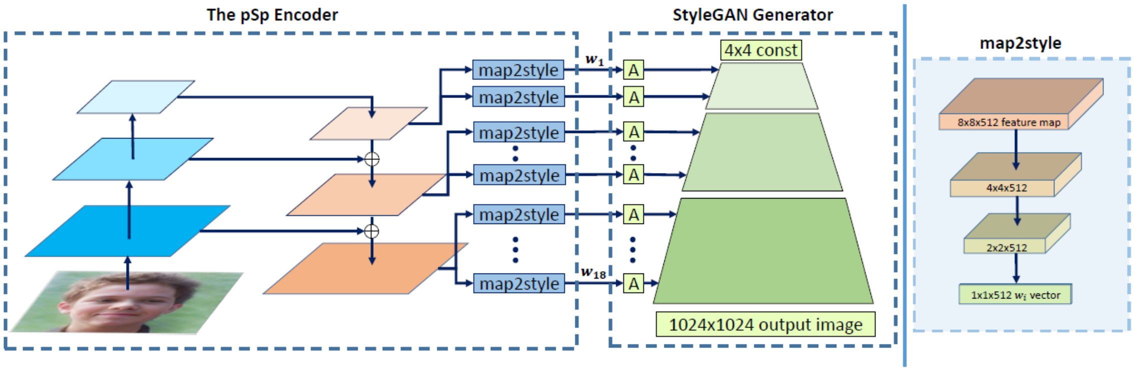

Figure 1: The pSp framework architecture. Feature Maps are extracted from the real image using standard feature pyramid. For each of the 18 target style, a small mapping network is trained to extract learned styles from feature map. This map2style is a fully connected convolutional network which uses Leaky ReLU as the activation function. Each of the 18 generated vector having a dimension of 1x512 is then given as input in the StyleGAN.

Most of the present day models have a specific architecture for solving a specific image to image translation problem. In contrast, our model offers a generic framework which is applicable for a lot of image-to-image translation problems. Our framework reduces the computational cost of training and using a pre-trained StyleGAN generator also eliminates the need of an adversary discriminator network. For instance, there are many models which use residual feature maps from the encoder in the input given to generator. This actually creates a bias which makes the model suitable for only a single type of image translation problem. The proposed model in this paper also supports the multi-modal synthesis for the image translation problems which have ambiguous outputs. These problems may include image generation from sketches, converting a low resolution image to a high resolution image etc. This framework has one additional feature which is even though the input and output images are from different domain this framework gives the desired output. Our method firsts encodes the images to style vectors and then back to images, which is the reason why we call this *framework pixel2style2pixel* (pSp).

The pSp Framework

The pSp framework uses both the SyleGAN generator and the extended Latent Space $W+$ for image translation tasks. The encoder required for this task must input each image to its accurate encoding in the latent domain. An easy way to achieve this would be to directly encode a given arbitrary image into $W+$ with the help of single 512-dimensional vector which is obtained from last layer of encoding network. The problem with this technique is that it learns all the 18 vectors together which again reduce the output quality because the model then fails to recognise the minute details of image.

Experimental results from the StyleGAN paper show that the we can divide different style input into roughly 3 classes - coarse, medium and finer details. In pSp, this observation is used with the help of a feature pyramid. This is the reason why feature pyramid generates 3 levels of feature maps. We use map2style network for extracting the features which is a fully connected convolutional network. After passing through map2style, the generated styles are then passed through a matching affine transformation A and then given as input to the generator.

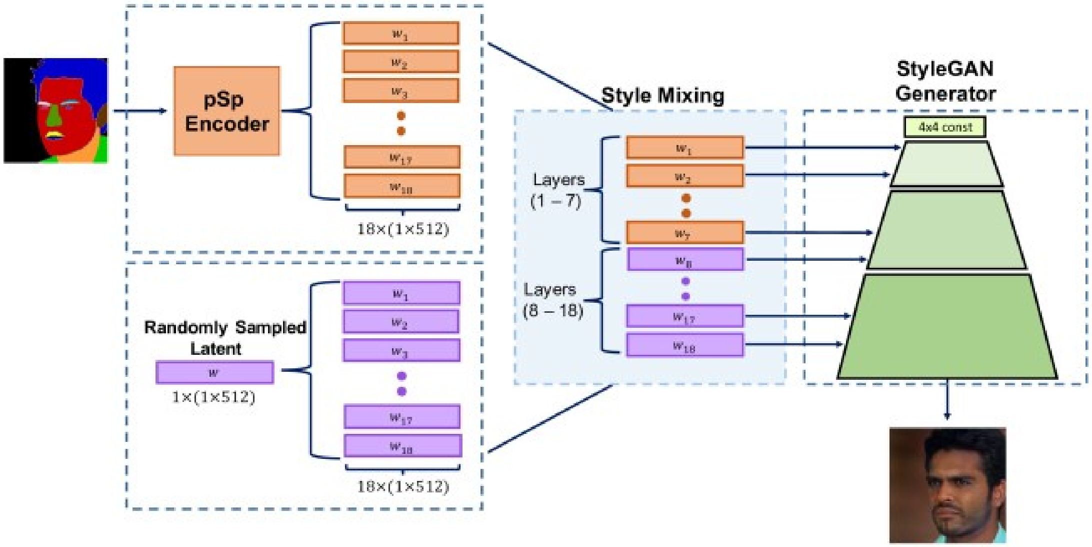

Figure 2: Style Mixing for Multi-Modal Generation

We define $\overline{w}$ as our average style vector of pretrained generator. If we take our input image as **x**, then the output of our model would be:

$$pSp(x) := G(E(x) + \overline{w})$$

Here, E() denotes the Encoder and G() denotes the generator. Through this formula, our encoder learns the latent code with respect to the average style vector $\overline{w}$.

Loss Functions Used in pSp

Since our framework consists of several layers of neural networks, therefore we chose a loss function which is a linear combination of several loss functions. The choice of loss function is very important as it is a principal component of training our neural networks. The losses which we use in our framework are:

- We first take a pixel-wise loss to measure the loss based on every pixel of an image.

$$\mathcal{L}_2(x) = ||x - pSp(x)||_2$$

- We use another loss known as LPIPS (Learned Perceptual Image Patch Similarity) Loss. This loss is used to learn the perceptual similarities in images as this loss preserves the image quality in a much better way as compared to its predecessors.In the below equation, F(.) represents the perceptual feature extractor from an image.

$$\mathcal{L}_{LPIPS}(x) = ||F(x) - F(pSp(x))||_2$$

- We also use a regularization loss to improve the quality of results of our encoder. Using this loss ensures better quality of images and it has no adverse effects on our output.

$$\mathcal{L}_{reg}(x) = ||E(x) - \overline{w}||_2$$

- When we encode the facial images, then a major problem is to maintain the individual identity of each face. For this purpose, we use a loss function which mainly aims to measure the cosine similarity of the output image with the source image. In the below equation, R is pretrained ArcFace network.

$$\mathcal{L}_{ID}(x) = 1 - \langle R(x), R(pSp(x)) \rangle$$

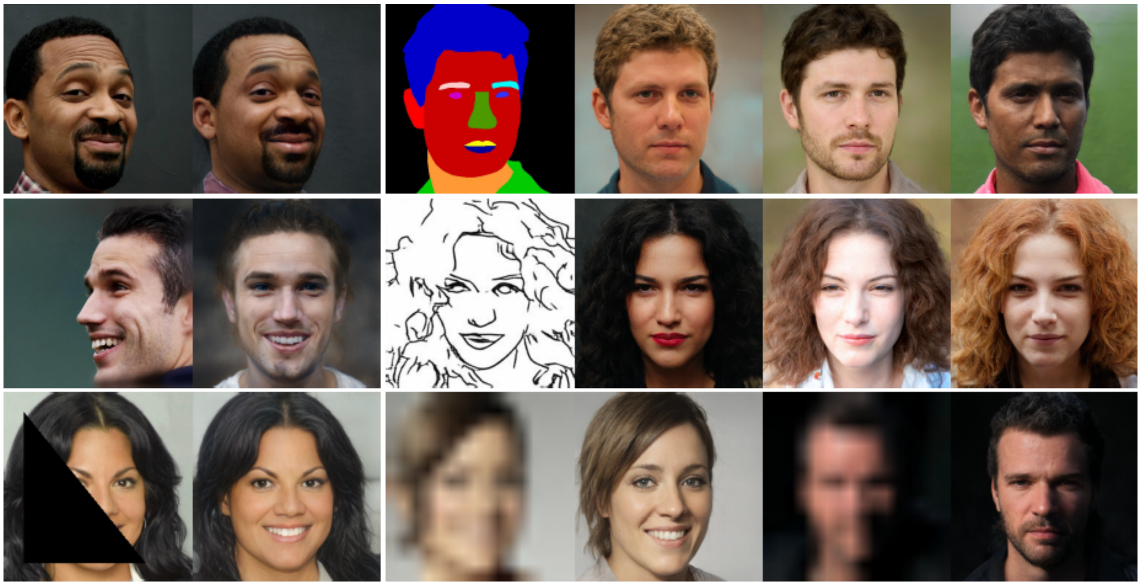

Figure 3: Applications of the proposed pixel2style2pixel framework. This framework can be used for various image-to-image translation tasks including StyleGAN inversion, facial frontalization, super-resolution, artificial image synthesis and multi-modal conditional image synthesis.

Hence, the overall loss function is defined as:

$$\mathcal{L}(x) = \lambda_1 \mathcal{L}_2(x) + \lambda_2 \mathcal{L}_{LPIPS}(x)$$

$$+\lambda_3 \mathcal{L}_{reg}(x) + \lambda_4 \mathcal{L}_{ID}(x)$$

In this function $\lambda_1, \lambda_2, \lambda_3 and \lambda_4$ are constants which denote the weight of each individual loss function. This loss function ensures a much more accurate encoding and can be easily tuned. Our framework works globally rather than locally due to the usage of style domains. Also, this framework has a layer wise representation which leads to support the multi-modal synthesis. The style mixing performed proves to be beneficial as it gives the framework an edge over several other models.

Applications

For judging the effectiveness of the framework, we evaluate it on several applications, some of which include:

Face from Sketch

Generation of faces from the sketch provided is a useful task which can be performed using many models. Most of these models do a pixel by pixel analysis. This process results in poor quality of results when the input is a sparse or partially incomplete sketch. DeepFaceDrawing solves this problem by using a collection of dedicated mapping networks. But this results in a lot of calculations which causes these models to be slow. pSp framework provides a simple method for doing this work. For this task we make a data set which represents sketches drawn by hand using **CelebA-HQ dataset**.

We then compare our model with pix2pixHD and DeepFaceDrawing. Same data set was used for testing purpose in all the methods. Since DeepFaceDrawing lacks the open source availability of code, we here directly use the results published in their research paper. We can see this comparison in Figure 4.

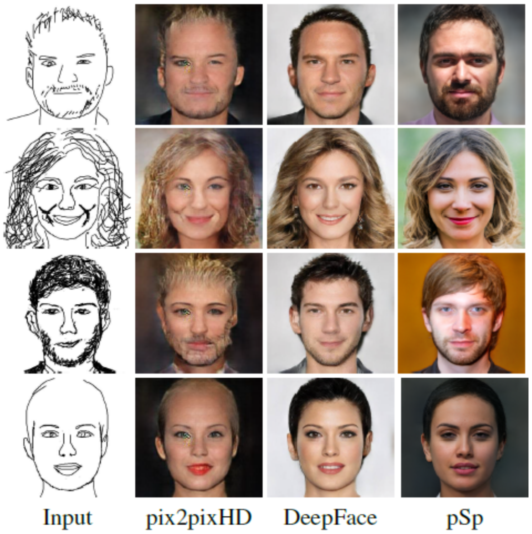

Figure 4: Comparison of images drawn from sketches using pix2pixHD, DeepFaceDrawing and pSp framework.

In Figure 4, we can clearly see that the results generated using DeepFaceDrawing model is better than that of pix2pixHD model. Our pSp model, although trained on a different dataset generates results that are slightly better than results of DeepFaceDrawing. The power to retain the finer details of face like facial hair is present in only our model, as can be clearly seen from the figure. This shows that our framework is globally applicable and the quality of results generated is also comparable to the state of the art models for particular tasks. Note that here we display the results of other models which are directly taken from their results. Our code contains only pSp model and its results.

Face from Segmentation Map

A segmentation map is a data object which stores the nearby geographical information which consists of using different colors to represent various facial features. Generation of facial images from segmentation maps is a popular application of deep learning models. We will use our pSp model to generate images from segmentation maps. Also, we will compare the results of our model to pix2pixHD approach and 2 state of the art models based on pix2pixHD method: SPADE and CC_FPSE. We will directly use the results of these models for comparison.
We visually compare the results of all the models considered on CelebAMask-HQ dataset which contains 19 semantic categories. All these methods are based on pix2pixHD, which causes the results of all these methods to suffer from similar kinds of defects. In comparison, our method generates much higher quality results on a wide range of inputs. Our model generates a number of outputs for same pose and attributes having different fine features for even a single semantic map. The figure below shows the results of our model as compared to other models.

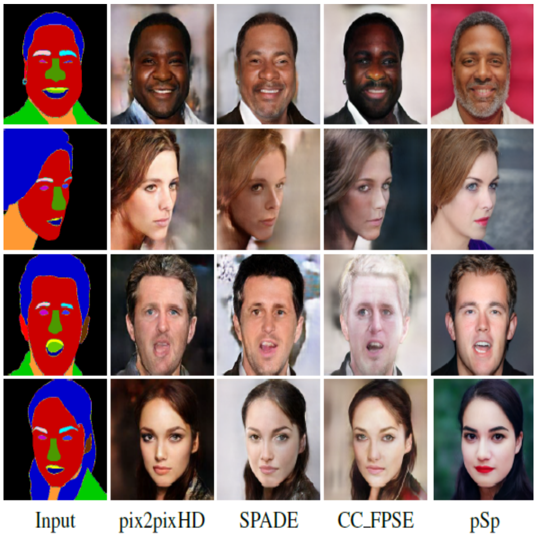

Figure 5: Comparison of images drawn from segmentation maps using pix2pixHD, SPADE, CC_FPSE and pSp framework.

We can see from the results that our model generates better quality results than the other models. Our pSp model captures the details of faces like colour consistency, shape of mouth etc. in a much better way than the other models. Still, for verification of the same a study was conducted where many persons studied almost 8500 pictures. Almost 94.72 % users prefer pSp over pix2pixHD, 95.25 % users prefer pSp over SPADE and 93.06 % users prefer pSp over CC_FPSE. These results clearly indicate that pSp is the best model among all models.Our code contains only our model results. rest of the results are taken from their respective papers.

Discussion

We have shown above some applications of our pSp framework. pSp can also be used for many other purposes which are not limited to facial domain. We can use it for a variety of task provided we have a pre-trained StyleGAN generator. Although our model generalizes well to a lot of tasks, there are certain assumptions which one should consider before using this model. First assumption is that all the images which are produced as an output are the images which can be generated using a StyleGAN generator. Thus, it may be hard to give a high quality output if similar examples were not used in the training process of StyleGAN generator. Also, when we use global approach of pSp then it might miss sone finer details like earrings.

Conclusion

In this paper, we present a novel architecture named pixel2style2pixel (pSp). This method may be used to directly map a real image to an extended latent space $W+$. This method also saves computational cost as no optimization is needed in this model. The styles are taken in an orderly manner from the images and are given as input in orderly manner to a StyleGAN generator. Now we use our encoder and StyleGAN decoder simultaneously to present a general framework for solving various image to image translation tasks. We show some applications of pSp and prove that its results are of much higher and better quality than the other present dedicated models. Our model provides a general framework which is responsible for solving a variety of problems by introducing only slight challenges. This might help other people who are currently doing research to use StyleGANs for actual image-to-image translation tasks.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In Proceedings of the IEEE international conference on computer vision, pages 4432–4441, 2019.

[2] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, and Hongbo Fu. DeepFaceDrawing: Deep generation of face images from sketches. ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2020), 39(4):72:1–72:16, 2020.

[3] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4401–4410, 2019.

[4] Tsung-Yi Lin, Piotr Doll´ar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[5] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1125–1134, 2017.

[6] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In Advances in Neural Information Processing Systems, pages 570–580, 2019.

[7] H.Kodanmana, CLL788-Process Data Analytics Class Notes