CGI2018 manuscript No.
(will be inserted by the editor)

Image-to-Image Translation: From Line to Sketch

Anonymous Author(s)

Abstract We investigate a specific task of image-to-image translation: line generation sketch task. We delve into two methodologies: pix2pix and pixel2style2pixel. The pix2pix framework employs a conditional adversarial network, wherein a generator built upon ”U-Net” architecture collaborates with a convolutional ”PatchGAN” classifier serving as the discriminator. On the other hand, the pixel2style2pixel is based on a novel encoder network, that directly generate series of style vectors which are fed into a pretrained StyleGAN generator, forming the extended $W+$ latent space. We demonstrate that these two approaches are both effective at synthesizing portrait sketches from lines. Code is available at `https://anonymous.4open.science/r/line-generate-sketch-BB47`.

Keywords Sketch · GAN · Pix2pix · Pixel2style2pixel

1 Introduction

Portrait sketching stands as a profound artistic expression, employing shades of black, white, and gray to sculpt captivating visages through the delicate interplay of lines. Diverse artistic styles and techniques imbue these sketches with a vast range of emotions and atmospheres. Nonetheless, the traditional realm of portrait sketching places great reliance on the singular drawing aptitude of the artist, necessitating substantial time and dedication for each creation. Enter the realm of AI-assisted portrait sketching, fueled by the advent of image style transfer, promising a significant reduction in the laborious nature of portrait production. Nevertheless, the prevalent approach to generating portrait sketches revolves around input photos, which may not always be readily available in all circumstances [1]. To tackle this predicament and empower individuals to swiftly acquire exquisite, top-tier portrait sketches tailored to their preferences, we embark upon the mission of generating such sketches sans photographs, utilizing rudimentary facial line images as the foundation for crafting their corresponding portraits. Our journey unfolds through the utilization of two versatile image-to-image translation frameworks: the esteemed pix2pix (p2p) [13] and the revolutionary pixel2style2pixel (pSp) [36].

Address(es) of author(s) should be given

In recent times, Generative Adversarial Networks (GANs) have made remarkable strides in the realm of image synthesis, particularly when it comes to facial images. Cutting-edge methods in image generation have achieved astounding visual quality and realism, surpassing previous benchmarks. Notably, StyleGAN [17,18] introduces a groundbreaking generator architecture based on style, delivering state-of-the-art visual fidelity even for high-resolution images. What sets StyleGAN apart is its disentangled latent space, denoted as W [4,38,41] , which empowers precise control and editing capabilities.

Pixel2style2pixel introduces an innovative encoder architecture that takes on the crucial task of encoding any given image directly into the extended latent space $W+$. The architecture of the encoder draws inspiration from the Feature Pyramid Network [24] , enabling the extraction of style vectors from various scales of a feature pyramid. These style vectors are then inserted directly into a fixed, pretrained StyleGAN generator, aligning them with their respective spatial scales. The encoder's remarkable capability lies in its ability to faithfully reconstruct real input images, facilitating latent space manipulations without the need for time-consuming optimization procedures. While these manipulations open up vast possibilities for editing real images, they do possess inherent limitations. Specifically, the requirement for the input image to be invertible poses a significant constraint. In other words, there must exist a latent code that can accurately reconstruct the image. This prerequisite proves to be a considerable limitation, particularly in tasks such as conditional image generation, where the input image does not belong to the same StyleGAN domain. To overcome this restriction, an effective approach is to employ the encoder in conjunction with the pretrained StyleGAN generator as a comprehensive image-to-image translation framework. Within this framework, input images are directly encoded into the desired output latents, which are subsequently fed into StyleGAN to generate the target output images. This empowers the utilization of StyleGAN for image-to-image translation, even in scenarios where the input and output images do not originate from the same domain.

In the case of pix2pix, the focus shifts towards exploring GANs within the conditional framework. While GANs learn a generative model of data, conditional GANs (cGANs) excel in acquiring a conditional generative model [8]. This unique characteristic renders cGANs particularly suitable for image-to-image translation endeavors, where the generation process is conditioned upon an input image, giving rise to a corresponding output image.

2 Related Work

Conditional GANs. Prior and concurrent works have conditioned GANs on discrete labels [6,7,29] , text [34] , and, indeed, images. The image-conditional models have tackled image prediction from a normal map [40] , future frame prediction [28] , product photo generation [42] , and image generation from sparse annotations [16,33] (c.f. [35] for an autoregressive approach to the same problem). Several other papers have also used GANs for image-to-image mappings, but only applied the GAN unconditionally, relying on other terms (such as L2 regression) to force the output to be conditioned on the input. These papers have achieved impressive results on inpainting [31] , future state prediction [44] , image manipulation guided by user constraints [45] , style transfer [22] , and superresolution [21] . Each of the methods was tailored for a specific application. Pix2pix framework differs in that nothing is applicationspecific. This makes its setup considerably simpler than most others.

Unlike past work, for the generator Pix2pix use a “U-Net”-based architecture [37] , and for the discriminator it use a convolutional “PatchGAN” classifier, which only penalizes structure at the scale of image patches. A similar PatchGAN architecture was previously proposed in [22] to capture local style statistics.

Image-to-Image. Image-to-Image translation techniques aim at learning a conditional image generation function that maps an input image of a source domain to a corresponding image of a target domain. Isola et al. [14] first introduced the use of conditional GANs to solve various image-to-image translation tasks. Since then, their work has been extended for many scenarios: high-resolution synthesis [39] , unsupervised learning [19,25,26,46] , multi-modal image synthesis [3,11,47] , and conditional image synthesis [2, 23, 27, 30, 48] . The aforementioned works have constructed dedicated architectures, which require training the generator network and generally do not generalize to other translation tasks. This is in contrast to pixel2style2pixel that uses the same architecture for solving a variety of tasks.

3 Methods

3.1 The Pix2Pix Framework

3.1.1 Network architectures

The architectures of the generator and discriminator in our approach are derived from the work in [32] . Both the generator and discriminator employ modules consisting of convolution, BatchNorm, and ReLU activations [12] .

Generator: U-Net. The task of translating line figures into portrait sketches involves mapping a high-resolution input grid to a high-resolution output grid, where the structural characteristics of the input are roughly aligned with those of the output. Previous approaches [15, 31, 40, 42, 44] to similar problems have employed encoder-decoder networks [10] . In these networks, the input undergoes a series of layers that progressively downsample the data until reaching a bottleneck layer, after which the process is reversed. Such a network architecture necessitates that all information flows through all the layers, including the bottleneck layer.

To provide the generator with a mechanism to bypass this bottleneck and facilitate the flow of relevant information, our approach incorporates skip connections, following the general structure of a ”U-Net” [37] . Specifically, we introduce skip connections between each layer i and layer $n - i$, where n represents the total number of layers. Each skip connection simply concatenates all the channels at layer i with those at layer $n - i$. Figure 1 illustrates the distinction between the encoder-decoder network and the U-Net architecture.

Discriminator: PatchGAN It is widely recognized that using L1 and L2 loss functions in image generation tasks can result in blurry outputs [20] . These loss functions are effective at capturing low-frequency information. However, to model high-frequency details, it is more appropriate to focus on the local structure within image patches. In light of this, our approach employs a discriminator architecture called PatchGAN, which specifically penalizes the structure at the patch scale. The PatchGAN discriminator aims to classify whether each $N \times N$ patch in an image

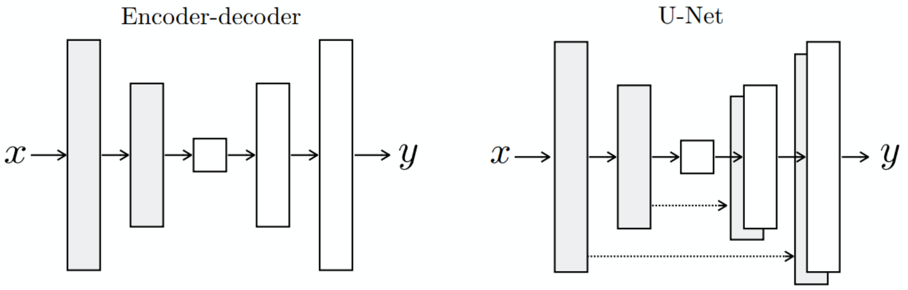

Fig. 1 Difference between encoder-decoder network and U-net

is real or fake. By convolving this discriminator across the image and averaging the responses, we obtain the final output D.

3.1.2 Objective Functions

The objective of a conditional GAN can be formulated as follows:

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

where G aims to minimize this objective against an adversarial D that seeks to maximize it, i.e., $G = \arg\min G \max D\mathcal{L}cGAN(G, D)$.

Previous approaches have shown the benefits of combining the GAN objective with a more traditional loss, such as the L2 distance [31] . However, our approach utilizes the L1 distance instead of L2, as L1 encourages less blurring. The L1 loss is defined as follows:

$$\mathcal{L}L1(G) = \mathbb{E}x, y, z[|y - G(x, z)|1]. \tag{2}$$

Therefore, the final objective is given by:

$$G^* = \arg\min_G \max_D \mathcal{L}cGAN(G, D) + \lambda\mathcal{L}_{L1}(G), \tag{3}$$

where λ represents constants that determine the weights of the losses.

3.2 The Pixel2Style2Pixel Framework

3.2.1 Network architectures

The Pixel2style2pixel (pSp) framework leverages the power of a pretrained StyleGAN generator and the $W+$ latent space. To effectively utilize this representation, a strong encoder is needed to accurately encode each input image into the latent domain. A simple technique to embed into this domain is directly encoding a given input image into W+ using a single 512-dimensional vector obtained from the last layer of the encoder network, thereby learning all 18 style vectors together. However, directly encoding an input image into a single 512-dimensional vector for the entire $W+$

latent space results in a bottleneck and limits the ability to fully represent the finer details of the original image, leading to lower reconstruction quality.

To address this, pSp extends the encoder backbone with a feature pyramid, inspired by the observation in StyleGAN that different style inputs correspond to different levels of detail, divided into coarse, medium, and fine. The feature pyramid generates three levels of feature maps, and a simple intermediate network called ”map2style” extracts styles from these feature maps, as shown in Fig:2.

The extracted styles, aligned with the hierarchical representation, are then fed into the generator according to their respective scales. This allows the generator to generate the output image by translating the input pixels to output pixels through the intermediate style representation. The complete architecture of the pSp framework is illustrated in Fig:2.

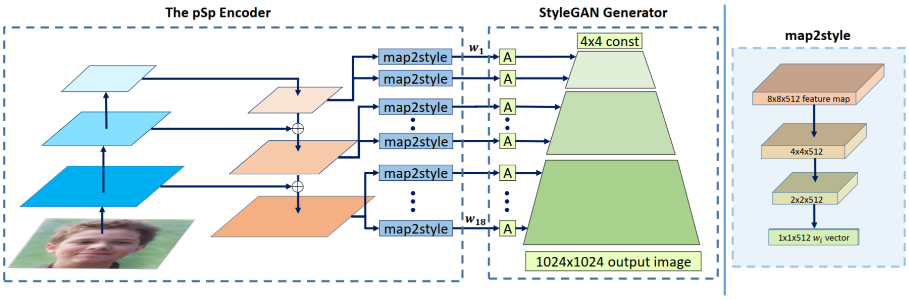

Fig. 2 pSp architecture.

In the pSp framework, the average style vector of the pretrained generator is denoted as $\overline{\mathbf{w}}$. Given an input image $\mathbf{x}$, the output of the model is defined as:

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

Here, $E(\cdot)$ represents the encoder and $G(\cdot)$ represents the StyleGAN generator. The encoder learns to encode the input image into a latent code with respect to the average style vector. By incorporating the average style vector during encoding, this formulation provides a better initialization for the model, leading to improved performance.

3.2.2 Objective Functions

While the style-based translation is the core part of the pSp framework, the choice of losses is also crucial. The encoder is trained using a weighted combination of several objectives. First, pSp utilize the pixel-wise L2 loss,

$$\mathcal{L}_2\left(\mathbf{x}\right) = ||\mathbf{x} - pSp(\mathbf{x})||_2. \tag{4}$$

In addition, to learn perceptual similarities, pSp utilize the LPIPS [43] loss, which has been shown to better preserve image quality [9] compared to the more standard perceptual loss [15] :

$$\mathcal{L}_{\text{LPHPS}}\left(\mathbf{x}\right) = ||F(\mathbf{x}) - F(pSp(\mathbf{x}))||_2, \tag{5}$$

where $F(\cdot)$ denotes the perceptual feature extractor.

To encourage the encoder to output latent style vectors closer to the average latent vector, pSp additionally define the following regularization loss:

$$\mathcal{L}_{\text{reg}}\left(\mathbf{x}\right) = ||E(\mathbf{x}) - \overline{\mathbf{w}}||_2. \tag{6}$$

Similar to the truncation trick introduced in StyleGAN, adding this regularization in the training of the pSp encoder improves image quality without harming the fidelity of our outputs.

Finally, a common challenge when handling the specific task of encoding facial images is the preservation of the input identity. To tackle this, pSp incorporate a dedicated recognition loss measuring the cosine similarity between the output image and its source,

$$\mathcal{L}_{\text{ID}}\left(\mathbf{x}\right) = 1 - \langle R(\mathbf{x}), R(pSp(\mathbf{x}))\rangle, \tag{7}$$

where R is the pretrained ArcFace [5] network.

In summary, the total loss function is defined as

$$\mathcal{L}(\mathbf{x}) = \lambda_1\mathcal{L}_2(\mathbf{x}) + \lambda_2\mathcal{L}_{\text{LPPS}}(\mathbf{x}) + \lambda_3\mathcal{L}_{\text{ID}}(\mathbf{x}) + \lambda_4\mathcal{L}_{\text{reg}}(\mathbf{x}), \tag{8}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are constants defining the loss weights. This curated set of loss functions allows for more accurate encoding into StyleGAN compared to previous works and can be easily tuned for different encoding tasks according to their nature.

4 Experiments

4.1 Pix2Pix

To optimize pix2pix network, we follow the standard approach: we alternate between one gradient descent step on D, then one step on G. In addition, we divide the objective by 2 while optimizing D, which slows down the rate at which D learns relative to G. We use minibatch SGD and apply the Adam solver, with a learning rate of 0.0002.

Fig:3 shows the training process. In the beginning, p2p learns a blurred outline. Then lines appear, with many repeated geometrical elements however. After training for 200 epochs, the sketch is relatively good. But the overall picture is rough. Finilly p2p fits sketches that is quiet sililar to the real output.

4.2 Pixel2Style2Pixel

For Pixel2Style2Pixel, We first finetune a StyleGanv2 using the model pretrained on FFHQ and the sketch images in the training set. The samples from the finetune StyleGanv2 model is shown in Fig:4.

We can conlcude from the samples that we only need to finetune the StyleGan for 3k-4k iterations in order to capture the sketch style.

After finetuning StyleGan, we start to trian the Pixel2Style2Pixel model.

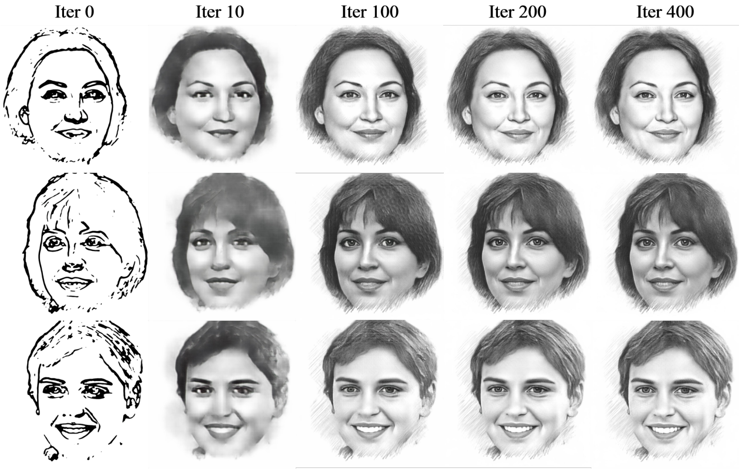

Fig. 3 Training process of p2p

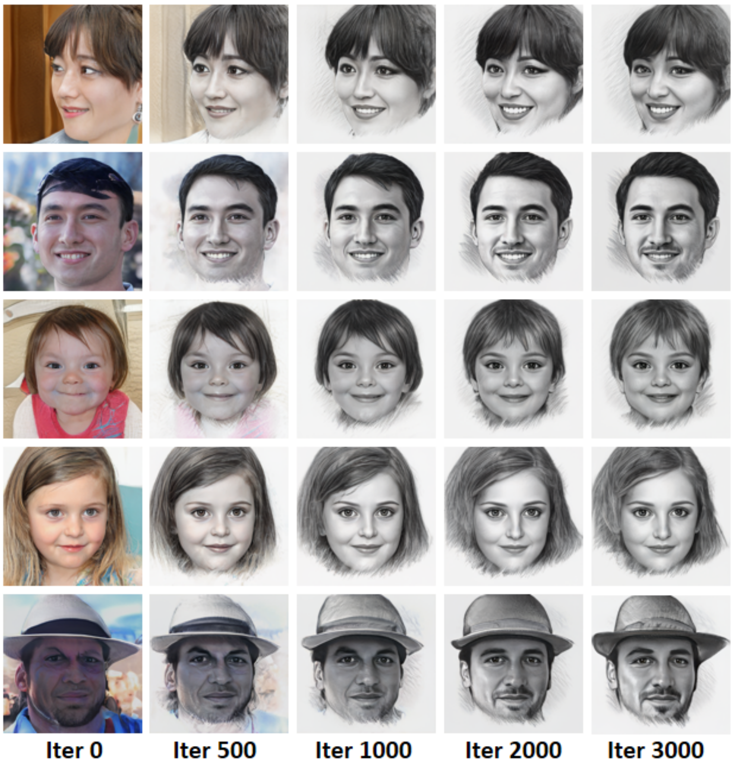

Fig. 4 Samples from the StyleGan

4.3 Performance

With a dataset consisting of 420 training images, we aimed to assess the model's performance in an intuitive manner. To achieve this, we partitioned the dataset into a training set (images 1-400) and a test set (images 401-420). Subsequently, we evaluated the performance of both the p2p and pSp models on the task of converting lines to sketches. To quantify their performance, we employed two metrics for evaluation.

FID The smaller the value of the FID, the closer the two Gaussian distributions are, and the better the performance of the GAN. In practice, it is found that FID has relatively good robustness to noise, can have a good evaluation of the quality of the generated image, and the score given is more consistent with human visual judgment, and the computational complexity of FID is not high.

SSIM SSIM stands for Structural Similarity Index Measure. It is a widely used method for assessing the similarity between two images. SSIM is designed to mimic the human perception of image quality by considering both the structural information and the statistical properties of the images.

4.3.1 Performance on Training Set

Firstly, we assess the performance of p2p and pSp models on the training set. The results are depicted in Figure 5. It is evident that p2p excels in preserving the distinctive features of the portrait sketch, such as the shape of the mouth and overall facial expression. This outcome can be expected as pSp employs StyleGAN, which introduces more variability in the generated images. Nevertheless, it should be noted that both p2p and pSp produce high-quality outputs on the training set, as the overall expression, color, and shape of the generated figures closely resemble the ground truth.

4.3.2 Performance on Test Set

To assess the models' generalization ability, we applied them to our separate test set, as illustrated in Figure 6. Similarly to the training set, p2p outperforms pSp in retaining the characteristics of the portrait sketch. However, on the test set, p2p's generated sketches are not flawless. Compared to the ground truth, the overall image appears rough, with slightly messy lines and areas of blurring. In these aspects, pSp exhibits better performance.

To enhance the quality of figures generated by p2p, we explored several potential optimizations. Notably, we discovered that removing the dropout layer from the generator U-net significantly improves the quality of the generated figures, as depicted in Figure 6. In the pix2pix paper [13], Isola et al. mentioned that dropout was introduced to enhance stochasticity by adding noise. However, in our task of generating accurate portrait sketches, the addition of noise is unnecessary. Therefore, removing dropout proves to be beneficial.

Table 1 presents the quantitative metrics results on the real test set, where DP denotes the dropout rate. It is observed that while the sketches generated by pSp exhibit some bias, their quality is relatively high compared to p2p. After removing the dropout in p2p, there is a significant improvement in the SSIM metric. This improvement aligns with the higher consistency of SSIM with human visual perception, as observed in Figure 6. However, without dropout, the FID metric deteriorates.

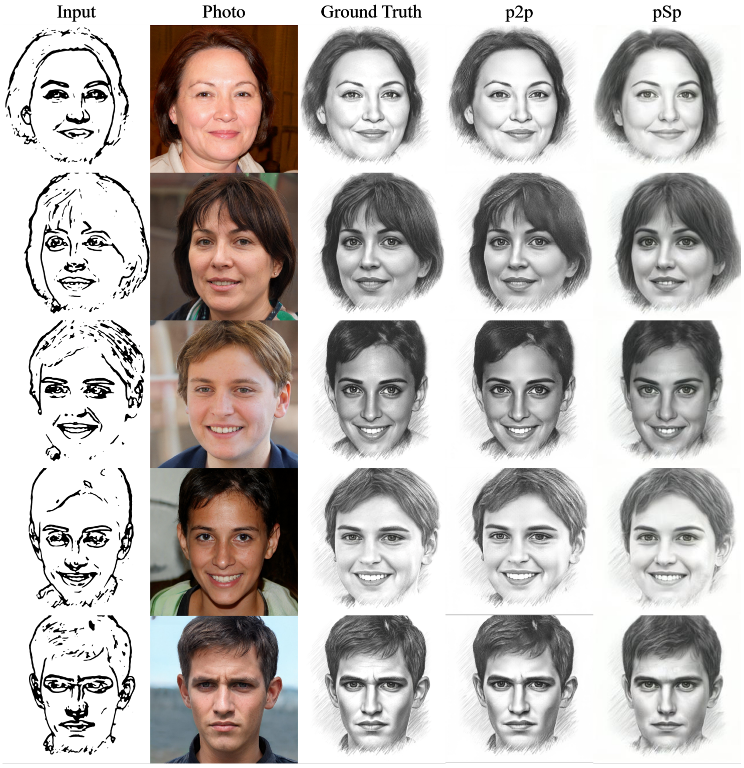

Fig. 5 Result on the training set.

This may be attributed to the increased risk of overfitting in p2p without the noise introduced by dropout. As FID is relatively robust to noise, a certain level of noise does not heavily affect its performance. By striking a balance between noise and quality, we adjust the dropout rate to achieve the best results. Ultimately, the best FID obtained is **0.2136**, while the best SSIM achieved is **0.7830**.

Table 1 Quantitative metrics result of different model.

Model	pSp	p2p(DR=0.5, default)	p2p(no dropout)	p2p(DR=0.2)
FID	0.2777	0.2848	0.3292	**0.2136**
SSIM	0.7301	0.7288	**0.7908**	0.7830

5 Discussion

After the experiments, we conclude that Pix2Pix and pixel2style2pixel have different approaches and characteristics. Here are the advantages and disadvantages of each:

- **Pix2Pix:**

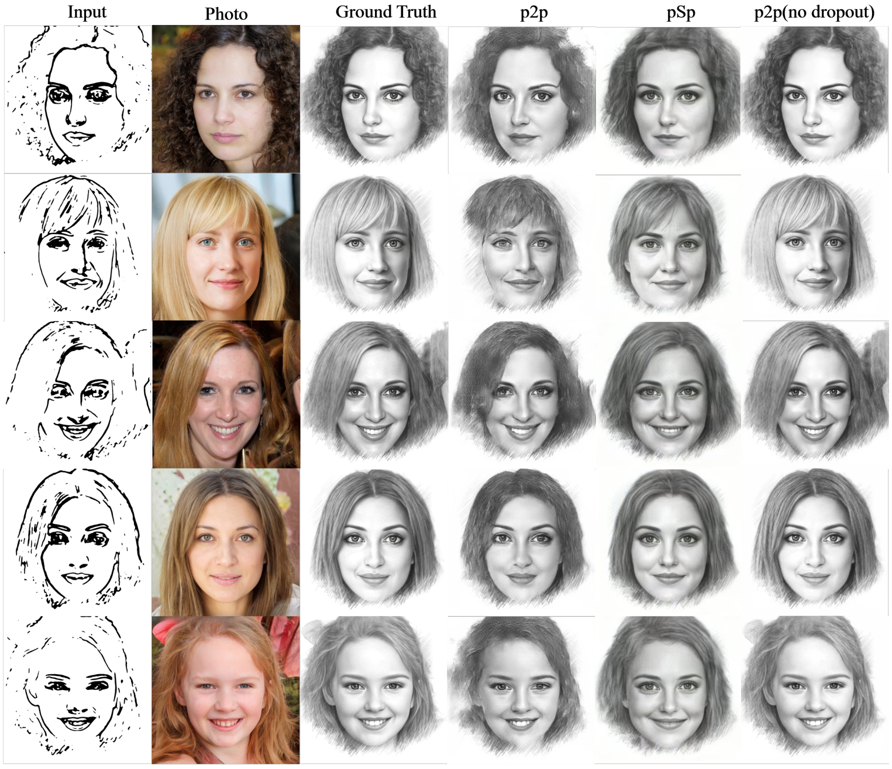

Fig. 6 Result on the test set.

- **Advantages:**
  1. Effective for paired image translation: Pix2Pix is designed for paired image translation tasks, where a model is trained to convert images from one domain to another based on a training dataset containing corresponding input-output pairs. It has been successfully applied to tasks such as image colorization, semantic segmentation, and edge-to-photo translation.
  2. Conditional adversarial training: Pix2Pix utilizes a conditional generative adversarial network (GAN) framework, which enables the generation of visually appealing and realistic output images. The adversarial training process helps in capturing the high-frequency details and textures of the translated images.
  3. Flexibility in input and output types: Pix2Pix can handle various types of image translation tasks, including grayscale to color, segmentation maps to images, and edges to photos. It allows for flexible input-output mappings, making it versatile for different applications.
- **Disadvantages:**
  1. Dependency on paired training data: The main limitation of Pix2Pix is its reliance on paired training data. It requires a large dataset with corresponding input-output image pairs, which can be challenging and time-consuming to obtain in certain cases.

    2. Limited generalization: Due to the dependency on paired data, Pix2Pix may not generalize well to unseen or diverse input examples that are significantly different from the training set. It struggles to generate coherent outputs for inputs that deviate substantially from the training data distribution.
    3. Difficulty in handling global structure changes: Pix2Pix is better suited for local image translation tasks rather than tasks involving global structural changes. It may face challenges when translating images with large-scale deformations or transformations that require significant changes in the global structure.
- **Pixel2Style2Pixel:**
  - **Advantages:**
    1. Unpaired image translation: Pixel2Style2Pixel is designed for unpaired image translation tasks, where the model learns to map images from a source domain to a target domain without requiring corresponding input-output pairs during training. It allows for more flexibility and scalability in data collection.
    2. Style transfer and image synthesis: Pixel2Style2Pixel not only performs direct image translation but also enables style transfer and image synthesis capabilities. It can generate diverse outputs by combining style and content from different images, providing more creative possibilities.
    3. Robustness to global structure changes: Pixel2Style2Pixel handles global structural changes more effectively compared to Pix2Pix. It can generate outputs with large-scale deformations and transformations, making it suitable for tasks involving significant changes in the global structure.
  - **Disadvantages:**
    1. Lack of direct supervision: Since Pixel2Style2Pixel operates in an unpaired setting, it lacks direct supervision during training. This can lead to challenges in maintaining the desired level of quality and coherence in the translated images, especially when dealing with complex and diverse datasets.
    2. Style inconsistency: While Pixel2Style2Pixel allows for style transfer, it may encounter issues with style inconsistency or artifacts in the generated outputs. Achieving consistent and accurate style transfer can be a challenging task.
    3. Longer training time: Pixel2Style2Pixel typically requires more training time and computational resources compared to Pix2Pix due to its unpaired nature and the need to learn style representations and mappings.

In summary, Pix2Pix is suitable for paired image translation tasks and offers good performance with paired training data. On the other hand, Pixel2Style2Pixel excels in unpaired image translation, style transfer, and image synthesis, providing more flexibility in data collection and handling global structural changes. However, Pixel2Style2Pixel may face challenges in maintaining consistency and requires more computational resources. The choice between the two frameworks depends on the specific requirements and characteristics of the image translation task at hand.

Regrettably, due to time constraints, we were unable to make detailed adjustments to the pSp model. However, we would like to discuss some potential improvements.

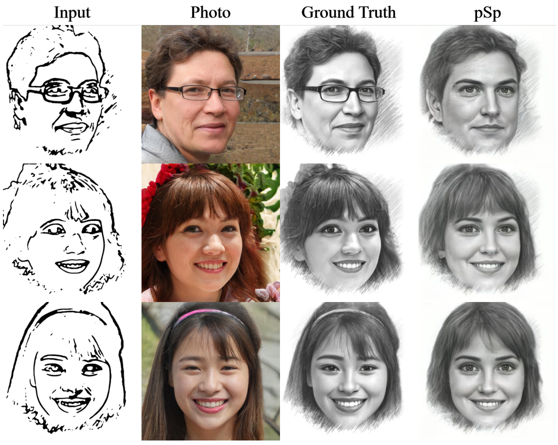

Fig. 7 Errors on pSp model.

Initially, we trained a pSp model that generates 256 × 256 images and then obtained 1024 × 1024 images through interpolation. Training the pSp model directly on 1024 × 1024 images might yield better results.

The training set consists not only of line figures and portrait sketches but also RGB pictures of real faces. This additional information can be particularly valuable for the pSp model. Specifically, during pSp training, we can fine-tune StyleGAN to generate sketch-style figures. Then, by using line figures as input and sketch figures as output, we can train the pSp encoder. By incorporating real face pictures, we can supervise the encoder to derive better style vectors.

During pSp model training, we observed some prominent errors. As shown in Figure 7, pSp struggles to generate decorations such as glasses or hair clasps accurately. This issue may arise from two main factors. Firstly, there might be an inadequate number of samples with decorations in the training set, which limits pSp's ability to learn about decorations. Secondly, the pretrained StyleGAN might not have learned enough about faces with decorations. A bias exists between the StyleGAN pretrained dataset and the line-sketch dataset. To address this problem, obtaining more examples of faces with decorations could prove to be a simple yet effective solution.

6 Conclusion

In this work, we use two frameworks, pix2pix and pixel2style2pixel, to solve an image-to-image translation task: line generation sketch task. Both methods achieve good performance in this task, especially p2p framework. Notably, we adjust the dropout of p2p's generator to balance the noise and quality. Our work demonstrate that general-purpose image-to-image translation frameworks can outstandingly solve the specific line to sketch task.

References

1. Cgi-portrait sketch generation (cgi-psg2023) challenge. URL `http://www.cgs-network.org/cgi23/cgi-psg2023/`
2. Chen, S.Y., Su, W., Gao, L., Xia, S., Fu, H.: Deep generation of face images from sketches (2020)
3. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains (2020)
4. Collins, E., Bala, R., Price, B., Süsstrunk, S.: Editing in style: Uncovering the local semantics of gans (2020)
5. Deng, J., Guo, J., Yang, J., Xue, N., Kotsia, I., Zafeiriou, S.: ArcFace: Additive angular margin loss for deep face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 5962–5979 (2022). DOI 10.1109/tpami.2021.3087709. URL `https://doi.org/10.1109%2Ftpami.2021.3087709`
6. Denton, E., Chintala, S., Szlam, A., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks (2015)
7. Gauthier, J.: Conditional generative adversarial nets for convolutional face generation (2015)
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (eds.) Advances in Neural Information Processing Systems, vol. 27. Curran Associates, Inc. (2014). URL `https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`
9. Guan, S., Tai, Y., Ni, B., Zhu, F., Huang, F., Yang, X.: Collaborative learning for faster stylegan embedding (2020)
10. Hinton, G.E., Salakhutdinov, R.: Reducing the dimensionality of data with neural networks. Science **313**, 504 – 507 (2006)
11. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation (2018)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015)
13. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1125–1134 (2017)
14. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks (2018)
15. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution (2016)
16. Karacan, L., Akata, Z., Erdem, A., Erdem, E.: Learning to generate images of outdoor scenes from attributes and semantic layouts (2016)
17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks (2019)
18. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan (2020)
19. Katzir, O., Lischinski, D., Cohen-Or, D.: Cross-domain cascaded deep feature translation (2019)
20. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric (2016)
21. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network (2017)
22. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks (2016)
23. Li, Y., Chen, X., Wu, F., Zha, Z.J.: Linestofacephoto: Face photo generation from lines with conditional self-attention generative adversarial network (2019)
24. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection (2017)
25. Lira, W., Merz, J., Ritchie, D., Cohen-Or, D., Zhang, H.: Ganhopper: Multi-hop gan for unsupervised image-to-image translation (2020)
26. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks (2018)

27. Liu, X., Yin, G., Shao, J., Wang, X., Li, H.: Learning to predict layout-to-image conditional convolutions for semantic image synthesis (2020)
28. Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error (2016)
29. Mirza, M., Osindero, S.: Conditional generative adversarial nets (2014)
30. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization (2019)
31. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting (2016)
32. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks (2016)
33. Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw (2016)
34. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text to image synthesis (2016)
35. Reed, S.E., van den Oord, A., Kalchbrenner, N., Bapst, V., Botvinick, M.M., de Freitas, N.: Generating interpretable images with controllable structure (2017)
36. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2287–2296 (2021)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015)
38. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing (2020)
39. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans (2018)
40. Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks (2016)
41. Yang, C., Shen, Y., Zhou, B.: Semantic hierarchy emerges in deep generative representations for scene synthesis (2020)
42. Yoo, D., Kim, N., Park, S., Paek, A.S., Kweon, I.S.: Pixel-level domain transfer (2016)
43. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric (2018)
44. Zhou, Y., Berg, T.L.: Learning temporal transformations from time-lapse videos (2016)
45. Zhu, J.Y., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold (2018)
46. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks (2020)
47. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation (2018)
48. Zhu, P., Abdal, R., Qin, Y., Wonka, P.: SEAN: Image synthesis with semantic region-adaptive normalization. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2020). DOI 10.1109/cvpr42600.2020.00515. URL `https://doi.org/10.1109%2Fcvpr42600.2020.00515`