AI-Signaturer: AI Automatic Signature Generation Based on Pix2pix

Busheng Zhang*

Department of Beijing University of Posts and Telecommunications, Beijing, China

*Corresponding author: 2023213779@bupt.cn

Abstract. This paper explores AI automatic signature generation technology based on the Pix2pix conditional generative adversarial network (cGAN), aiming to solve the issues of low efficiency and poor consistency associated with traditional manual signing. Leveraging the learning capability of deep learning models, this study is dedicated to achieving automatic, efficient, and high-quality signature generation, providing technical support for areas such as electronic documents. For this purpose, the paper has designed and trained a Pix2pix model consisting of a generator and a discriminator that can convert input images into signature images with specific font styles. In experiments, 900 signature samples collected manually by students were used for training. To optimize model performance, strategies such as early stopping and model checkpoints were introduced, and hyperparameters like learning rate and iteration counts were adjusted during training. Ultimately, the model achieved an average accuracy of 76% across three different types of input fonts—Kaiti, YaHei, and FangSong—and clearly displayed Chinese character features. However, for complex and dense font styles, there is room for improvement in the model's performance. Future research will focus on more refined data preprocessing and exploring more complex model structures like CycleGAN to enhance performance in multi-font recognition tasks, improving the model's generalization and robustness.

Keywords: Generative Adversarial Networks; Pix2pix; AI Automatic Signature Generation; Conditional Generative Adversarial Network; Deep Learning.

1. Introduction

With the continuous development of artificial intelligence technology, significant progress has been made in the field of image generation. Among these, image generation techniques based on generative adversarial networks (GANs) have shown great potential in tasks such as image generation and translation due to their unique adversarial training mechanism [1]. Pix2Pix, a variant of conditional GAN (cGAN), since its proposal in 2017, has stood out in image translation tasks for its ability to perform image-to-image mapping, such as synthesizing photos from label maps, reconstructing objects from edge maps, and coloring images [2]. Signatures, as an important means of identity verification and information recording, are particularly crucial for authenticity and aesthetics in practical applications. However, traditional manual signing methods suffer from inefficiency and inconsistency, making it difficult to meet the demands of modern efficiency and automation. Therefore, using Artificial Intelligence (AI) technology to automatically generate signatures becomes a promising research direction. Based on the Pix2Pix algorithm, this study aims to develop an AI signature generation model that learns to transform input images (such as handwritten signature samples) through specific fonts into output images (generated signatures) by training the model on the mapping relationship between image transformations. This model not only increases the efficiency of signature generation but also ensures consistency and authenticity, providing strong technical support for fields such as electronic documents and contract signings. This study will detail the basic principles of the Pix2Pix algorithm and its application in signature generation tasks, collect and prepare corresponding training datasets, design and train an AI signature generation model based on Pix2Pix, and evaluate and verify the model's performance. Additionally, it will discuss the potential and challenges of applying this model in real-world scenarios, offering guidance for future research directions.

2. Organization of the Text

2.1. Research Methods

2.1.1 Data Preparation

In this study, during the data preparation phase, a total of 900 data samples were adopted, collected manually by students. These numerous samples come from personalized signatures generated by various websites on internet platforms, as well as unique signatures written by some students, ensuring the diversity and richness of data sources. Notably, the collected signature images exhibit a high degree of regularity and angularity in the running-Kaiti font style, significantly enhancing the overall consistency and recognizability of the dataset.

To further improve the efficiency and accuracy of data processing, all signature images underwent a strict formatting process. Specifically, each image file was renamed according to the "XXX.png" naming rule, where "XXX" directly corresponds to the signature text displayed in the image. This naming strategy greatly facilitates subsequent data retrieval and classification. Moreover, all carefully formatted signature images were systematically integrated into a dedicated folder, providing significant convenience for upcoming preprocessing procedures and laying a solid material foundation for the smooth progress of the entire research process.

2.1.2 Data Preprocessing

After the data collection stage, the obtained data must undergo a series of preprocessing steps to be converted into a suitable format for model training. First, the data will be systematically stored in a dedicated folder named “targeted_image” for easy access and reading by subsequent programs. Then, through preset program logic, in another folder marked “generated_image”, standard FangSong, Microsoft YaHei, and Kaiti font images corresponding one-to-one with each data item name are generated. These generated images serve as the input images for model training, aimed at enhancing the model’s generalization and recognition accuracy through diversified font forms.

Subsequently, to further optimize the organization structure of the dataset, a specific “rename” program was used to digitally rename the existing dataset. For instance, files originally named “jiangwen.png” were renamed to “1.png”. This process ensured that files in the target folder and the input folder had consistent naming rules and numbers, facilitating subsequent data matching and processing work.

During the data loading phase, PyTorch framework’s Dataset and DataLoader classes, combined with the glob module, were used to efficiently load annotated images and target images from specified paths and arrange them orderly. To ensure the consistency and effectiveness of data input, detailed data preprocessing procedures were defined, including converting image data into tensor format, uniformly adjusting image sizes to 256x256 pixels, and performing normalization. Furthermore, a custom CMP_dataset dataset class inheriting from torch.utils.data.Dataset was created, implementing two core methods, __getitem__ and __len__, to obtain samples and the total number of samples in the dataset as needed.

Finally, by configuring DataLoader instances, batch loading of the dataset was realized, with the shuffle parameter set to True to ensure random shuffling of data during training, effectively avoiding biases caused by data order and further improving the training effect and generalization ability of the model.

2.1.3 Model Architecture

The model architecture adopts the classic structure of Pix2pix. In this study, the generator architecture consists of six downsampling modules, five upsampling modules, and one output layer. Each downsampling module includes a convolutional layer (Conv2d), LeakyReLU activation function, and batch normalization layer (BatchNorm2d). The upsampling process is implemented via transposed convolutional layers (ConvTranspose2d), also using LeakyReLU activation functions and batch normalization; notably, Dropout layers are introduced in some upsampling stages to enhance the model's generalization ability. The last layer of the generator uses transposed convolution operations to convert feature maps into RGB-formatted image outputs. On the other hand, the discriminator design includes two downsampling modules plus an additional convolutional layer. Its basic components are similar to those of the generator, i.e., convolutional layers, LeakyReLU activation functions, and batch normalization layers. Particularly, this discriminator receives concatenated results of annotation images and images to be detected as input and ultimately produces a single-channel output indicating the probability of the given image pair belonging to real data.

2.1.4 Model Hyperparameter Settings and Training

Regarding hyperparameter settings, both the generator and discriminator learning rates were set to 1e-3, and Adam optimizer was used for parameter updates, with betas values configured as (0.5, 0.999). The batch size (BATCHSIZE) was determined to be 32. For loss functions, binary cross-entropy loss (BCELoss) was chosen for the discriminator, while the generator's loss comprised adversarial loss (also based on BCELoss) and L1 loss, with the latter's weight LAMBDA set to 7. The entire model training process was expected to last 30 epochs.

The model training phase first involved initializing the generator and discriminator and transferring them to GPU devices (if available). Then, corresponding optimizers and loss functions were defined. In each epoch, the following procedure was executed by iterating over the data loader: First, the discriminator was trained, calculating the loss of real image pairs (d_real_loss) and the loss after generating fake images (d_fake_loss), then backpropagating and updating the discriminator parameters based on the sum of the two losses; subsequently, the generator training part included generating fake images, obtaining feedback from the discriminator to calculate adversarial loss (gen_loss_crossentropyloss) and L1 loss (gen_l1_loss), and finally completing backpropagation and parameter updates based on the weighted total loss. Additionally, after each epoch, average loss situations were recorded and printed, and generated images were drawn to visually demonstrate the training effects.

For result visualization, after completing each epoch, corresponding images were generated using data from the test set (such as extended/.jpg and extended/.png files), and differences between inputs, actual outputs, and generated outputs were displayed comparatively. At the same time, changes in the loss functions of the discriminator and generator during training were plotted to analyze the model's learning dynamics.

3. Experimental Process and Results

In this study, detailed observations and analyses were conducted on the effects of three different types of input fonts—namely, Kaiti, YaHei, and FangSong—and the loss functions were visualized as shown in fig. 1, fig. 2 and fig. 3. By plotting the loss change curves of these three fonts, researchers could intuitively compare and evaluate the impact of different fonts on the model training process. The results showed that the loss values corresponding to Kaiti, YaHei, and FangSong fonts eventually stabilized between approximately 0.8 and 0.9, indicating that although the model has certain learning capabilities, there is still room for further exploration and development in terms of optimization space. Moreover, based on the analysis of the test set data, the model's average accuracy was about 76%, which further confirmed the view that the input font type (whether standard Songti, Kaiti, or YaHei) has little influence on model performance.

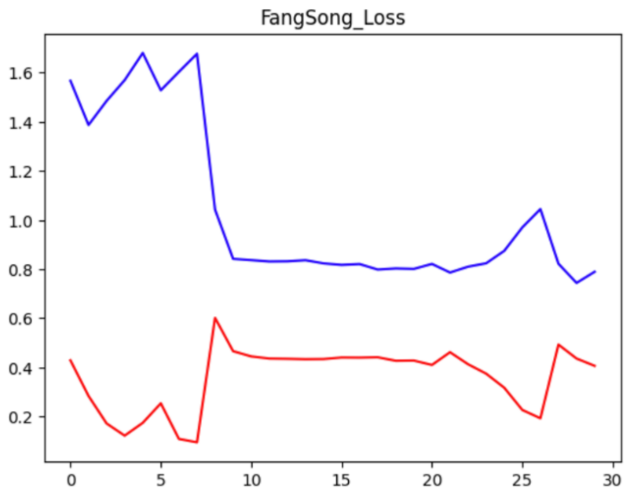

Fig. 1 Losses of Fangsong_loss

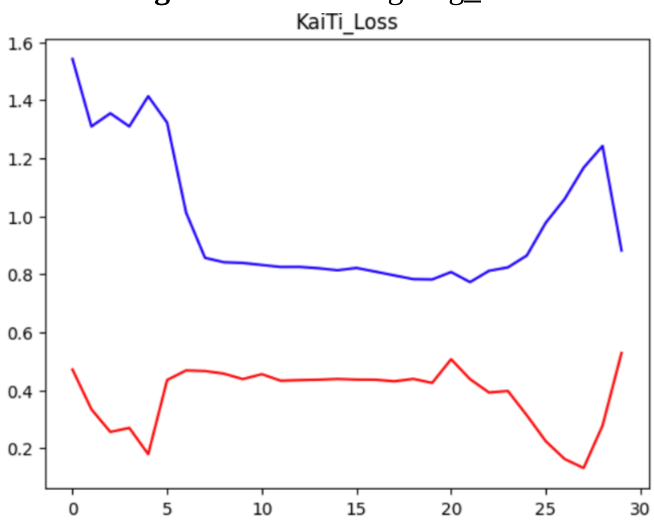

Fig. 2 Losses of KaiTi_loss

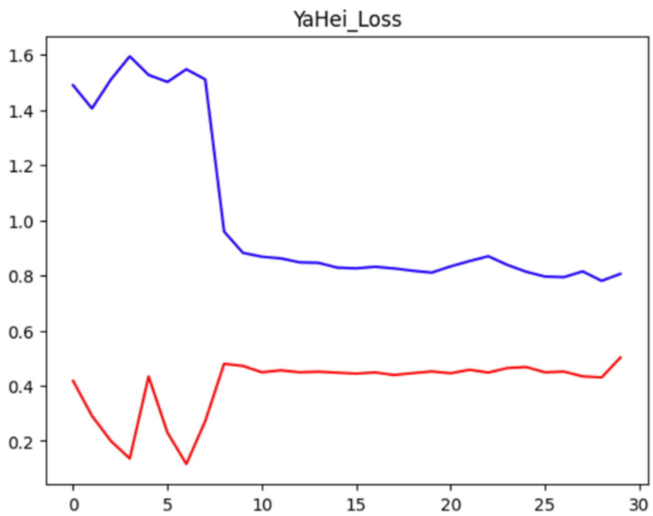

Fig. 3 Losses of YaHei_loss

Fig. 4 Sample Outputs

Figure 4 shows an example of automatic signature generation. From the perspective of output image quality, in most cases, the model-generated images could clearly display Chinese character features, making it relatively easy for human observers to identify these characters. However, under certain conditions, the output images of the model may not achieve the expected clarity and identifiability. Specifically, when the input image is simple and has low density, the model often produces better results; conversely, facing complex and dense font styles, the model's performance is less satisfactory. This might be because high-density image information poses greater processing difficulty for the model, affecting its final output quality. In summary, while the current research findings have revealed the basic capabilities and limitations of the model in handling different types of input fonts, future research needs to strive to improve the existing model architecture to achieve more efficient and accurate text recognition.

4. Further Imporvements

Aiming at the problems existing in the above research and the shortcomings in model performance, the main directions for future improvements are concentrated in two aspects: refined data preprocessing and optimization of model structure.

Firstly, in terms of refined data preprocessing, plans include taking more meticulous data preprocessing measures, especially for the alignment processing of Chinese characters. By deeply aligning the original dataset and the target dataset, it can effectively reduce pix2pix transformation errors caused by positional differences of Chinese characters, ensuring that the model focuses more on learning font features during training rather than being affected by positional deviations [3]. Such improvements not only help to enhance the model's generalization ability but also increase its robustness and adaptability in practical application scenarios [4,5].

5. Conclusion

The research discusses AI automatic signature generation technology based on the Pix2pix algorithm, aiming to achieve automatic, efficient, and high-quality signature generation through cGAN. The study first reviewed the development of generative adversarial networks and their variants, analyzed the limitations of existing signature generation technologies, and clarified the goals and innovations of this research. For model construction, the study adopted the classic Pix2pix architecture containing a generator and a discriminator, designing a network structure adapted to the signature generation task. By normalizing and formatting 900 Chinese character signature images from datasets such as HWDB and OLHWDB, it ensured the diversity and consistency of training data. During training, the Adam optimizer and binary cross-entropy loss function were used to adjust model parameters, achieving effective learning of Kaiti, YaHei, and FangSong fonts. Experimental results showed that the model could generate images with recognizable Chinese character features, with an average accuracy of 76%.

References

[1] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. Advances in neural information processing systems, 2014, 27

[2] Isola P, et al. Image-to-image translation with conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[3] Karras T. Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv preprint arXiv:1710.10196. 2017.

[4] Gulrajani I, et al. Improved training of wasserstein gans. Advances in neural information processing systems 2017, 30.

[5] Zhu, J Y, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. Procee dings of the IEEE international conference on computer vision. 2017.