TransGAN For Image Inpainting

Group: Haoran Lin, Zihao Lin, Hongbo Peng, Qishen Mai

Abstract

In recent years, image inpainting methods based on deep learning have demonstrated significant advantages compared to traditional approaches. Image inpainting technique is an important research direction in the field of computer vision, and its goal is to generate realistic-looking and semantically sound images by filling in the missing regions in the images. This paper focuses on image inpainting techniques based on Generative Adversarial Network (GAN) algorithms and analyses their significant advantages over traditional methods in recent years through in-depth analysis. Generative Adversarial Networks can learn more complex data distributions by introducing the game mechanism of generators and discriminators, which provides a powerful modelling capability for image inpainting tasks. Compared with traditional methods, GAN-based image inpainting methods show more excellent performance in maintaining the overall consistency of the image and improving the fidelity of the inpainting effect. In this paper, considering the realism and semantic reasonableness, we propose an image inpainting method using GAN algorithm comprehensively, aiming to promote the development of image inpainting technology. We try to implement image inpainting with Place365-Standard and CelebA-HQ-256 using TransGAN algorithm and AOTGAN. The experiment shows that our model has certain feasibility and validity, but it needs more computing resource and training time to get better results. Our codes are available in this URL.

1. Introduction

With the continuous development of the field of computer vision, image inpainting techniques, as one of its important branches, show a wide range of application prospects in the fields of image editing and image rendering. One of the core challenges of image inpainting lies in how to effectively fill in the missing pixel regions in an image to generate a visual effect that is both close to the real and semantically reasonable. In recent years, the introduction of generative adversarial network (GAN) algorithms has brought new opportunities and challenges to the field of image inpainting.

Traditional image inpainting methods are mainly based on the local information of the image, such as pixel value, gradient, texture, etc., to estimate the content of the missing regions. Although these methods can deal with some simple image damage situations, they often produce unnatural or unrealistic inpainting results when facing complex missing images or large areas of image damage.

Generative Adversarial Networks, as a powerful deep learning tool, can learn complex data distributions by modelling the game between generators of real data distributions and adversarial discriminators to generate realistic images. In the image inpainting task, the generator network using GAN can learn and generate missing regions that match the style of the original image, while the discriminator network can evaluate the authenticity and semantic reasonableness of the generated results, thus improving the accuracy and realism of the inpainting results. Image inpainting techniques based on GAN algorithms have attracted much attention because of their potential to achieve high-quality inpainting while maintaining the overall consistency of the image.

In this work, we implement the image inpainting task with TransGAN and AOTGAN, where TransGAN is a model that generates images based on pure Transformer, and AOTGAN is a model that uses a generator based on the composition of AOT blocks. However, based on the open-source code, we find that TransGAN uses Gaussian noise as the generator's input and is used only for overall image generation but not image inpainting. So we use gaussian blur to soften the edges of the mask images and add mask to the gaussian noise as model input. In terms of data sets, we use CelebA-HQ-256 and Place365-Standard with their masked images as the model training data set, and obtain the experimental results.

2. Related work

To realize the image inpainting technique, our team used GAN as the basis model and found a relatively large image dataset.

2.1. Generative Adversarial Network

Generative Adversarial Network (GAN)[1] is a neural network model consisting of a generator and a discriminator, which is capable of learn the high-level semantic information of an image by means of adversarial learning to generate realistic images.

Generator (G) is a network capable of generating images from random noise, and its goal is to make the generated image as close as possible to the real image distribution.

Discriminator (D) is a network that can determine whether an image is from a real dataset or not, and its goal is to distinguish the generated image from the real image as much as possible.

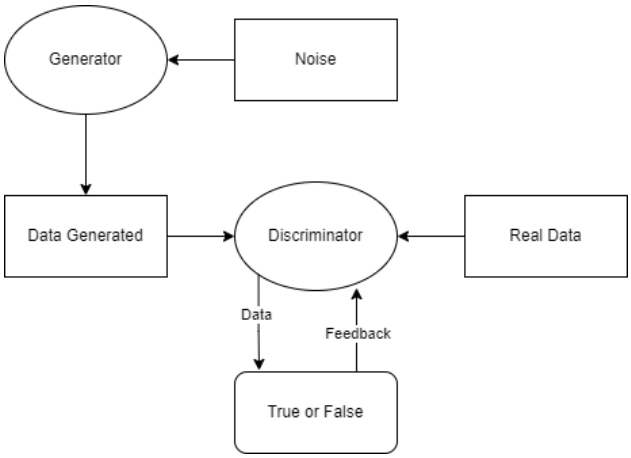

Figure 1: An overview of GAN

There is an adversarial relationship between the generator and the discriminator, where the generator tries to trick the discriminator into not being able to correctly determine whether an image is real or fake, while the discriminator tries to recognize the deception of the generator and improve the accuracy of its judgement.

The generator and the discriminator are trained alternately until a Nash equilibrium is reached, i.e., the image generated by the generator and the real image cannot be distinguished by the discriminator.

2.2. Aggregated Contextual Transformations for High-Resolution Image Inpainting

AOT-GAN is made for high resolution image inpainting. To achieve this goal, Yanhong Zheng's team constructs the generator by stacking multiple layers of a proposed AOT block. An AOT block consists of three parts: split, transformation, and merge. This special design make generator possible to forecast each output pixels. To make generated images more clearly, they propagate Soft Mask-Guided PatchGAN(SM-PatchGAN), enforcing the ability of discriminator to distinguish generated images from real ones.[2] Compared to other GANs, AOT-GAN can better achieve promising completions in practical use.

2.3. Generation Adversarial Network with Transformer

TransGAN uses transformer to replace the original convolution of GAN. It consists of a memory-friendly transformer-based generator that progressively increases feature resolution, and correspondingly a multi-scale discriminator to capture semantic contexts and low-level textures simultaneously. To scale up TransGAN to high-resolution generation, Yifan Jiang's team introduce a new module of grid self-attention for alleviating the memory bottleneck further. They also develop a unique training recipe including a series of techniques that can mitigate the training instability issues of TransGAN, such as data augmentation, modified normalization, and relative position encoding.[3] Compared to other GANs, TransGAN has better performance in its best architecture.

3. Dataset Description

3.1. CelebA

CelebA is the abbreviation of CelebFaces Attribute, which means Celebrity Face Attribute dataset, which contains 202,599 face images of 10,177 celebrity identities[4], and each image is well labelled with feature markers, including face bounding box and annotation box, 5 facial feature point coordinates, and 40 attribute markers. CelebA is openly provided by the Chinese University of Hong Kong, and it is widely used in face-related computer vision training tasks, and it can be used for face attribute marking training, face detection training, and landmark labelling.

3.2. Place365-Stardand

In total, the Places2 dataset contains over 10 million images with over 400 unique scene categories. The dataset has between 5,000 and 30,000 training images per category[5], which is consistent with the frequency of real-world scenes.

The Places2 dataset consists of 3 main datasets: Place365-Standard, Place365-Challenge 2016 and Place-Extra69.

In this study, only Place365-Standard will be included. Place365-Standard is a core subset of the Places2 Database, which has 1.8 million training images from 365 scene categories for training the Place365 CNN. There are 50 images per category in the validation set and 900 images per category in the test set.

The Place365-Stardand can be divided into 3 parts:

1. High-resolution images(512*512)
2. small images(256*256)
3. small images(256*256) with easy directory structure

Regardless of the original aspect ratio, the images in the above file have been resized to 128 * 128.

4. Approach

In this project, we use two GAN models, AOT-GAN and TransGAN. AOT-GAN is a model proposed by Yanhong Zeng et al. in 2021. It is used to solve the following two problems: image content reasoning from distant contexts, and fine-grained texture synthesis for a large missing region. On the Place365-Standard dataset, AOT-GAN achieved a relative improvement of 38.60% in FID.[2] As for TransGAN, it is a completely convolution-independent GAN that uses a pure Transformer architecture, created by University of Texas at Austin student Yifan Jiang et al. TransGAN uses data augmentation, generator's multi-tasking of the auxiliary loss function and self-attention local initialization to improve model performance. Specifically, TransGAN has a FID of 18.28 on the STL-10 dataset. It also reaches the inception score of 9.02 and FID of 9.26 on CIFAR-10 dataset, and 5.28 FID on CelebA 128×128 dataset.[3]

4.1. Aggregated Contextual-Transformation GAN (AOT-GAN)

4.1.1 Generator and Discriminator

The discriminator consists of a stack of 5 2-Dimension convolutional layers, progressively downsampled by convolutional stride steps and padding, using spectral normalization to stabilize parameter updates. Activation uses LeakyRELU to avoid neurons stop updating. The input is a dataset image of shape 3 × 512 × 512 (C, H, W) (When discriminating the true image) or a generated composite image (When discriminating the A-image, which consists of the missing parts generated by the generator and the rest of the dataset image stitched together). The output shape is 1×30×30.

The first and last ends of the generator are three-layer encoder and decoder structures, respectively, and the middle layer of the feature extraction module (the blue part in the generator structure in the figure below) consists of an 8-layer AOT Block stack. The input consists of the training set image with the part to be complemented removed (3×512×512) stacked with a mask layer (1×512×512). 3-channel RGB images are added with a 1-channel mask layer so that the input is shaped as 4×512×512 (C, H, W). The output is a complemented image of shape 3 × 512 × 512 (C, H, W).

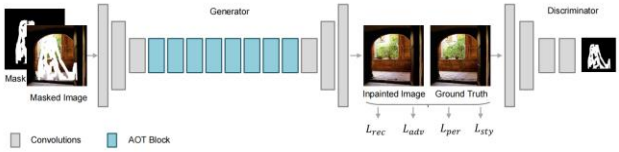

Figure 3: An overview of AOT-GAN

The aggregation of long range contextual information is handled by the 8-layer AOT Block in the generator. Compared with the Residual Block commonly used in ordinary CNN networks, the AOT Block makes the following improvements:

A set of 3×3 convolutions is used instead of the Identity cross-layer connection in the normal residual block. Moreover, AOT Block adds a Gate threshold between the aggregation module and the two sets of channels formed by the added set of 3×3 convolutions, allowing the model to automatically decide whether or not to use the aggregation channels in the spatial dimension. This is mainly done to reduce the color deviation between the filled area of the image and the original preserved area.

Instead of the first 3×3 convolutional layer in the normal residual block, four sets of null convolutions with expansion rates of 1,2,3, and 4 are spliced together in the channel dimension. Each set of inflated convolutions has one-fourth of the original number of channels, and the number of channels is kept constant when spliced together. In this way, the AOT Block would be able to aggregate contextual features with different spatial distances.

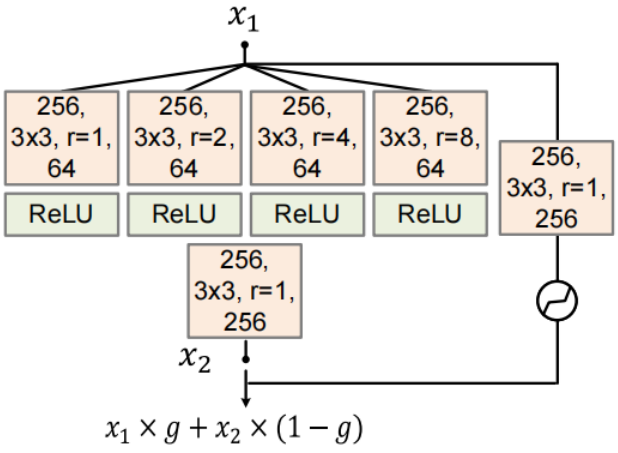

Figure 4: Structure of AOT Block

4.1.2 Loss Function

The loss function of AOT-GAN uses the loss of mean squared error, which is the loss function of "least squares GAN".

$$L_{adv}^{D} = E_{z\sim p_z}\left[\left(D(z) - \sigma(1-m)\right)^2\right] + E_{x\sim p_{data}}[(D(x)-1)^2] \quad (1)$$

The generator loss of AOT-GAN consists of four parts: Reconstruct Loss, Perceptual Loss, Style Loss, and Adversarial Loss.

$$L_{adv}^{G} = E_{z\sim p_z}[(D(z)-1)^2 \odot m] \quad (2)$$

We represent a real image as x and its corresponding binary inpainting mask as m (with a value of 0 for known pixels and 1 for missing regions). The generator is denoted as G. The loss is computed through a direct calculation of pixel-wise errors between the generated image and the original image, commonly referred to as L1 Loss. This involves pixel-wise multiplication with the inpainting mask m。

$$\mathbf{L_{rec}} = \|x - G(x \odot (1-m), m)\|_1 \quad (3)$$

Perceptual Loss[6] calculates the L1 Loss between the generated image and the feature maps of each layer output by the VGG19 model pre-trained on ImageNet.

$$\mathbf{L_{per}} = \sum_i \frac{\|\phi_i(x) - \phi_i(z)\|_1}{N_i} \quad (4)$$

Style Loss[7] calculates the SSIM (Structure Similarity Index Measure) between the generated image and the original image. takes the brightness, contrast, and structure of the image into account, rather than just pixel-level comparisons. The SSIM value ranges from -1 to 1, where 1 means that the two images are identical. A higher SSIM value indicates that the two images are more similar in structure.

$$L_{sty} = E_i[\|\phi_i(x)^T\phi_i(x) - \phi_i(z)^T\phi_i(z)\|_1] \quad (5)$$

The total generator loss is the weighted sum of these four Loss components, with the following weighting ratios: $\lambda_{adv} = 0.01$, $\lambda_{rec} = 1$, $\lambda_{per} = 0.1$, and $\lambda_{sty} = 250$.

$$\mathbf{L} = \lambda_{adv}L_{adv}^{G} + \lambda_{rec}L_{rec} + \lambda_{per}L_{per} + \lambda_{sty}L_{sty} \quad (6)$$

4.2. TransGAN

TransGAN consists of a memory-friendly generator and a Patch-Level discriminator that progressively increases the feature resolution while decreasing the embedding size.

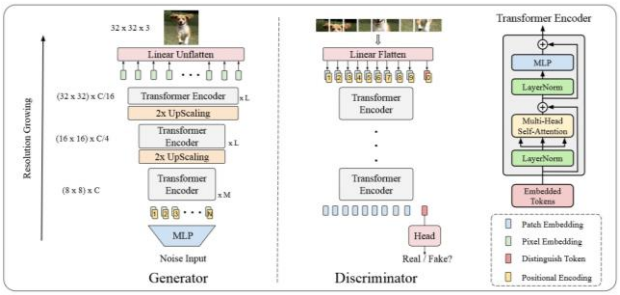

Figure 5: Structure of TransGAN

On one hand, traditional GANs suffer from mode collapse and exhibit training instability. On the other hand, conventional GAN architectures rely on convolutional operations, which have the drawback of limited receptive fields, leading to the loss of fine details in deeper layers. To address these issues, the authors propose TransGAN.

In the generator, the input is random noise, initially processed through a multi-layer perceptron (MLP) to form a long sequence. Subsequently, the sequence undergoes Transformer encoder blocks, producing an elongated sequence. The up sampling module comprises reshaping, up sampling, and reshaping stages. Initially, the long sequence is reshaped to 8×8×C (transforming the 1D sequence into a 2D image feature). Then, using bicubic interpolation, the sequence is up sampled, increasing the sampling resolution to 16×16×C image features without reducing the dimension. It is then reshaped back into a 1D sequence. This reshaped 1D sequence undergoes steps 2 and 3 again, generating 32×32×C image features, reshaped into a 1D sequence. Another iteration through steps 2 and 3 produces 64×64×C image features, reshaped into a 1D sequence. At this point, without immediately performing step 3, the up sampling module is improved. Unlike step 3, the bicubic interpolation is replaced with the pixel shuffle module. The 4th long sequence is reshaped into 64×64×C image features, up sampled using pixel shuffle, transforming it into features, and then reshaped back into a 1D sequence. a, after another round of Transformer encoder blocks, the generated long sequence is reshaped again, undergoes another round of pixel shuffle, transitioning from features to features. Finally, a linear weighting is applied, resulting in an image of size 256×256×3.

The following functions are the loss of generator of TransGAN. Depending on different parameters, it uses opposite of the discriminator score Loss, WGAN-GP Loss or Least Square GAN(LSGAN) Loss.

$$\mathbf{L_G}(\boldsymbol{x}) = -\frac{\mathbf{1}}{\boldsymbol{N} * \boldsymbol{M}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} \sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{M}} \boldsymbol{x_{ij}} \tag{7}$$

$$\boldsymbol{L_G^{WGAN-GP}} = \boldsymbol{L_G}(\boldsymbol{x}) + \frac{\mathbf{1}}{\boldsymbol{lz} + \mathbf{1} \times \mathbf{10^{-5}}} \tag{8}$$

$$\boldsymbol{lz} = \overline{|\boldsymbol{f_1} - \boldsymbol{f_2}|} + \overline{|\boldsymbol{z_1} - \boldsymbol{z_2}|} \tag{9}$$

$$\boldsymbol{L_G^{LSGAN}} = \left(\boldsymbol{D}\left(\boldsymbol{G}(\boldsymbol{z})\right) - \boldsymbol{c}\right)^2 \tag{10}$$

Here parameter x represents the discriminator's score matrix for generated images. $\boldsymbol{N}$ and $\boldsymbol{M}$ are the dimensions of the tensor which is about fake image. Parameter f_1 represents first half generated images and f_2 represents remaining half generated images. Parameter z_1 represents for the masks of first half images and z_2 represents for the masks of remaining half generated images. Parameter c represents the real image matrix. It will be set to 1 when the image is real, 0 when it is fake.

The discriminator's task is to distinguish between real and fake images, essentially a classification task. The authors have devised a multi-scale discriminator that takes inputs of varying sizes at different stages. The image is initially divided into P $\times P$, 2P $\times 2P$, 4P $\times 4P$ blocks, serving as different scales. The first scale has a size of $\frac{H}{P} \times \frac{W}{P} \times 3$, initially transformed through linear weighting into $\frac{H}{P} \times \frac{W}{P} \times \frac{C}{4}$. Similarly, the second scale $\frac{H}{2P} \times \frac{W}{2P} \times 3$ is transformed into $\frac{H}{2P} \times \frac{W}{2P} \times \frac{C}{4}$, and the third scale $\frac{H}{4P} \times \frac{W}{4P} \times 3$ is transformed into $\frac{H}{4P} \times \frac{W}{4P} \times \frac{C}{2}$. The tokens obtained from the first scale are used as input for the first transformer block, while the tokens from the second and third scales are connected to the tokens of the second and third stages, respectively, capturing more texture information. Similar to the generator but in reverse, the tokens are initially input into transformer blocks, and the resulting one-dimensional vectors are reshaped into two-dimensional feature maps. Average pooling layers are employed between stages to down sample the feature map resolution. At the end of these blocks, a [cls] token is appended at the beginning of the 1D sequence to output the real or fake prediction.

The following functions are the loss of discriminator of TransGAN. Depending on different parameters, it is divided into LSGAN Loss and WGAN-GP Loss. Here, $\boldsymbol{RV}$ represents the discriminator's score for real images, and $\boldsymbol{FV}$ represents the discriminator's score for generated images. The function $\boldsymbol{ReLU}$ denotes the rectified linear unit. $\boldsymbol{GP}$ corresponds to the gradient penalty term in WGAN-GP, and $\boldsymbol{phi}$ is the weight of the gradient penalty.

$$\boldsymbol{L_D^{LSGAN}} = (\boldsymbol{D}(\boldsymbol{x}) - \boldsymbol{b})^2 + \left(\boldsymbol{D}\left(\boldsymbol{G}(\boldsymbol{Z})\right) - \boldsymbol{a}\right)^2 \tag{11}$$

Here parameter b indicates for real images, a indicates for fake images.

$$\begin{aligned} \boldsymbol{L_D^{WGAN-GP}} = &-\frac{\mathbf{1}}{\boldsymbol{N} * \boldsymbol{M}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} \sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{M}} \left(\boldsymbol{RV_{ij}}\right) \\ &+ \frac{\mathbf{1}}{\boldsymbol{N} * \boldsymbol{M}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{N}} \sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{M}} \left(\boldsymbol{FV_{ij}}\right) \\ &+ \frac{\boldsymbol{GP} \times \mathbf{10}}{\boldsymbol{phi^2}} \end{aligned} \tag{12}$$

We also use other loss functions to compare with. WGAN

To enable TransGAN for image restoration, we replaced the original Gaussian noise in the input with Gaussian noise processed through a mask. We augmented the Generator by incorporating information from both the mask and the real image. In the final output, the information generated by the Generator replaces the portions of the real image covered by the mask. This completes the construction of a Transformer Generator capable of image restoration. The core of the entire image restoration process lies in how to integrate the mask with the Gaussian noise. Our proposed approach involves randomly selecting a mask from the mask dataset for use in Gaussian noise during the current iteration. The mask is first dimensionally aligned with the hidden layer through an Average pooling Layer(Avgpool Layer) to match the dimensions of the Gaussian noise. Subsequently, we apply a certain degree of scaling to the mask, emphasizing higher values in the masked region and reducing values in the non-masked region. This aims to focus the Gaussian noise on the masked area. The third step involves applying Gaussian blur to the mask to achieve a smoother transition between masked and non-masked regions, reducing abrupt changes in numerical values. Finally, this processed mask is applied to the randomly generated Gaussian noise during model training. Figure 6 is the structure of Generator for inpainting.

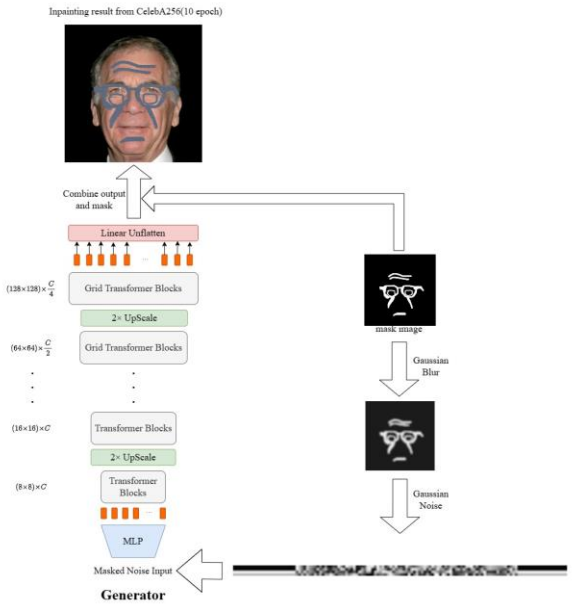

Figure 6: TransGAN Generator's architecture for inpainting

5. Experiment Result

In this experiment, Place365-Standard and CelebA-HQ-256 datasets are used to test the performance of AOT-GAN and TransGAN. Place365-Standard was resized to 128× 128, due to the hardware condition and time constraints. And CelebA-HQ-256 remains 256×256 resolution.

Regarding the masks, for datasets like Place365-Standard with diverse image content, this project employs a self-generated mask dataset. The process involves randomly selecting a point on the image and then randomly determining the length and width of the mask, thereby creating mask images.

Performance evaluation is conducted using four metrics: MAE (Mean Absolute Error), PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), FID (Fréchet Inception Distance) and Inception Score.

1. L1 error is commonly employed in prior studies, calculating the mean absolute error between inpainted and original images to assess per-pixel reconstruction accuracy.
2. Peak Signal-to-Noise Ratio (PSNR) stands out as a classical image quality assessment metric widely utilized in various inpainting approaches.
3. Structural Similarity Index (SSIM)[8] provides a comprehensive comparison between inpainted results and original images, considering aspects such as luminance, contrast, and structure.
4. Fréchet Inception Distance (FID)[9] serves as a popular deep metric for perceptual rationality. FID quantifies the distance between the distributions of real and fake image features. It's noteworthy that deep metrics have been demonstrated to align more closely with human perception.
5. The Inception Score is a metric used to evaluate the quality and diversity of generated images by a generative model, especially in the context of generative adversarial networks (GANs). Introduced by Salimans et al. in 2016, it leverages the Inception v3 model[10], originally designed for image classification, as a pretrained feature extractor.

5.1. AOT-GAN

The operating environment for AOT-GAN is Ubuntu 22.04, and it utilizes a 3050Ti 4GB graphics card. The training dataset comprises the Place365-Standard dataset, which is split into training, testing, and validation sets. The model is trained for a total of 180,000 iterations. Subsequently, the generated model is employed for image restoration on images from two datasets with applied masks.

In this project, several images from the test set were chosen to showcase the model's performance and were compared in a table with the performance of the pre-trained model published by the authors of AOT-GAN. ↓ means lower is better and ↑ means higher is better. Best results are **highlighted** and underlined.

Figure 7: A part of test picture in Place365-Standard with mask

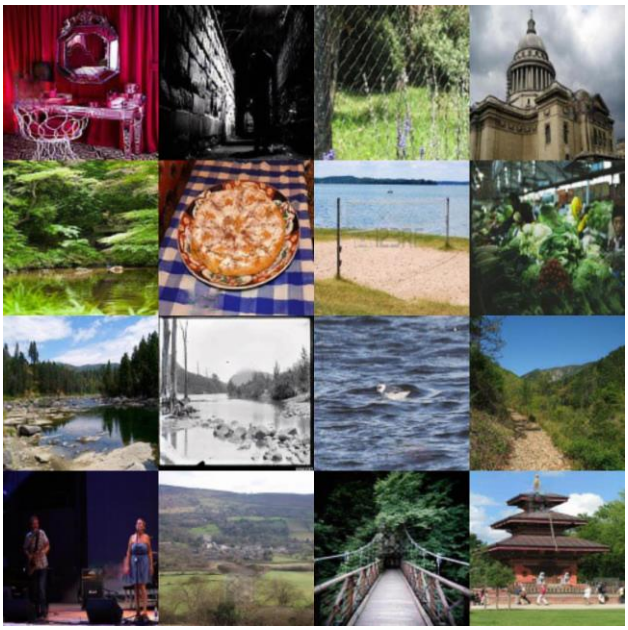

Figure 8: A part of generated pictures from Place365-Standard

Model	MAE↓	PSNR↑	SSIM↑	FID↓
Pre-trained Model	0.10260	20.19112	0.65648	327.19713
Ours	**0.036360**	**28.22898**	**0.91983**	**321.68214**

Table 1: Quantitative comparisons on Place365-Standard of our AOT-GAN model with pre-trained model. Calculation results here is based on 16 images in Figure 7.

Figure 9: A part of test picture in CelebA-HQ-256 with mask

Figure 10: A part of generated pictures from CelebA-HQ-256

Model	MAE↑	PSNR↓	SSIM↓	FID↓
Pre-trained Model	0.32461	9.28235	**0.18351**	117.11430
Ours	**0.33194**	**9.26776**	0.19245	**91.25946**

Table 2: Quantitative comparisons on CelebA-HQ-256 of our AOT-GAN model with pre-trained model. Calculation results here is based on 3000 images in Figure 9.

For Place365-Standard, the pre-trained model shows relatively poor performance with a higher MAE, lower PSNR, and SSIM compared to the model trained for 180,000 iterations.

After 180,000 generations, there is an improvement in image quality, as indicated by lower MAE, higher PSNR, and SSIM. The FID also decreases, suggesting a better alignment between the distributions of real and generated image features.

For CelebA-HQ-256, the pre-trained model demonstrates good performance with low MAE, high PSNR, SSIM, and relatively low FID. After 180,000 generations, there is a slight increase in MAE, decrease in PSNR and SSIM, but the FID also decreases, indicating potential improvements in capturing more complex features of CelebA-HQ-256 images.

In summary, the models trained for 180,000 iterations generally exhibit improved image quality compared to the pre-trained model, as reflected in lower MAE, higher PSNR, and SSIM. FID scores suggest that the distributions of generated and real image features become more aligned with training. However, further analysis and potentially additional evaluation metrics are advisable for a comprehensive understanding of the models' performance. However, from the generated images, it can be observed that our model performs well in repairing regions where the covered image content is relatively simple. In cases where the masked areas are more complex, with a variety of colors and significant variations, the restoration results are not as satisfactory. Further training may be required for the model to handle such intricate regions effectively.

5.2. TransGAN

The operating environment for TransGAN is Ubuntu 16.04, with a RTX3090 24GB graphics card installed. The training dataset comprises the Place365-Standard dataset and CelebA-HQ-256 dataset, which are split into training, testing, and validation sets. The model is trained for 27 epochs in Place365-Standard and 10 epochs in CelebA-HQ-256. Subsequently, the generated model is employed for image restoration on images from two datasets with applied masks.

In TransGAN's experiments, we compared our model with JPGNet and TransInpaint on FID score in Table 3. In Table 4, we compared the result of inpainting on Place365-Standard dataset with different loss functions, including Wgangp-eps loss, Wgangp-mode and Lsgan

loss. In Table 5, we compared the result of inpainting on Place365-Standard dataset with different running epochs. Furthermore, we also compared the inpainting results of AOTGAN and TransGAN on CelebA-HQ-256 dataset. ↓ means lower is better and ↑ means higher is better. Best results are **highlighted** and underlined.

Method	FID↓
JPGNet ACM-MM'21[11]	14.62
TransInpaint ICCVW_2023[12]	**4.46**
TransGAN (Ours Run 27 epoch)	18.66

Table 3: Quantitative comparisons on CelebA-HQ-256 of our TransGAN model which has run 27 epochs with JPGNet and TransInpaint.

Compared with other models, our TransGAN model got higher FID score, meaning that the images generated by our model are faker than other models.

Loss function	Inception Score↑	FID↓
Wgangp-eps (epoch10)	**35.843**	**22.142**
Wgangp-mode (epoch10)	34.057	27.117
Lsgan (epoch10)	34.905	31.348

Table 4: Quantitative comparisons on Place365-Standard of our TransGAN model which has run 10 epochs with different loss functions including Wgangp-eps, Wgangp-mode and lsgan.

In Table 4, among these three loss functions, we find that wgangp-eps loss function got the highest IS and lowest FID in epoch 10. Thus, we would like to use wgangp-eps as the loss function to train our model.

Epoch	Inception Score↑	FID↓
6	33.948	21.418
10	35.843	22.142
27	**41.256**	**18.662**

Table 5: Quantitative comparisons on Place365-Standard of our TransGAN model with different running epochs.

In Table 5, we set wgangp-eps as loss function, founding that with the epoch number increased, the inception score increased, and FID score decreased, indicating that our method to inpaint image is viable. However, it needs more time to train.

We have visualized the loss of training process by Tensorboard. The loss figures are in appendix with Figure A.1 and Figure A.2.

6. Team Member Contribution

Team Member	Contribution	Proportion
Haoran Lin	Writing research report; Searching for datasets; Run AOT model	20%
Zihao Lin	Writing research report; Configurate environment for TransGAN; Train TransGAN model	30%
Hongbo Peng	Creating presentation PPT; Trying to migrate code from AOT-GAN to TransGAN	20%
Qishen Mai	Writing research report; Configurate environment for TransGAN; Run AOT model; Evaluate TransGAN model	30%

7. Acknowledgement

We would like to express our sincere gratitude to all the individuals who contributed to the successful completion of this project. The collaborative efforts and support from our fellow colleagues have been invaluable. First and foremost, we extend our heartfelt thanks to our project team members. Their dedication, hard work, and insightful discussions significantly enriched the project. Each team member played a crucial role in achieving our goals.

We are also thankful to our advisors and mentors for their guidance and continuous support throughout the project, especially Mr. Fan and Mingle Hong. Their expertise and constructive feedback were instrumental in shaping the direction of our work.

Furthermore, we appreciate the assistance received from our friends who provided valuable insights, feedback, and encouragement. Their contributions greatly enhanced the quality of our research.

Lastly, we want to acknowledge the collective effort of everyone who directly or indirectly contributed to the success of this project. Your collaboration and commitment have been instrumental, and we are truly grateful for the collaborative spirit that defined this project.

This work would not have been possible without the collective effort and commitment of all involved. Thank you for being part of this journey.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. WardeFarley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in NeurIPS, 2014, pp. 2672–2680.

[2] Zeng Y, Fu J, Chao H, et al. Aggregated contextual transform ations for high-resolution image inpainting[J]. IEEE Transa ctions on Visualization and Computer Graphics, 2022.

[3] Jiang Y, Chang S, Wang Z. Transgan: Two pure transformer s can make one strong gan, and that can scale up[J]. Advanc es in Neural Information Processing Systems, 2021, 34: 147 45-14758.

[4] Large-scale CelebFaces Attributes (CelebA) Dataset, http://m mlab.ie.cuhk.edu.hk/projects/CelebA.html

[5] Places, http://places2.csail.mit.edu/

[6] Johnson J, Alahi A, Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14. Springer International Publishing, 2016: 694-711.

[7] Gatys L A, Ecker A S, Bethge M. Image style transfer using convolutional neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2414-2423.

[8] Wang Z, Simoncelli E P, Bovik A C. Multiscale structural similarity for image quality assessment[C]//The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. Ieee, 2003, 2: 1398-1402.

[9] Heusel M, Ramsauer H, Unterthiner T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium[J]. Advances in neural information processing systems, 2017, 30.

[10] Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training gans[J]. Advances in neural information processing systems, 2016, 29.

[11] Guo Q, Li X, Juefei-Xu F, et al. Jpgnet: Joint predictive filtering and generative network for image inpainting[C]//Proceedings of the 29th ACM International Conference on Multimedia. 2021: 386-394.

[12] Shamsolmoali P, Zareapoor M, Granger E. TransInpaint: Transformer-based Image Inpainting with Context Adaptation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 849-858.

A. Loss in Training

Our model has been trained at least 20 epochs. For Places365-standard, our batch size is 16 and the training set has 36500 * 0.7 photos, so each epoch's iteration is 1596. Figure A.1 and Figure A.2 are the images of the generator and discriminator loss value changes respectively, where the horizontal axis represents the total number of iterations, and the vertical axis represents the loss value. The smoothing of scalars is set to 0.6.

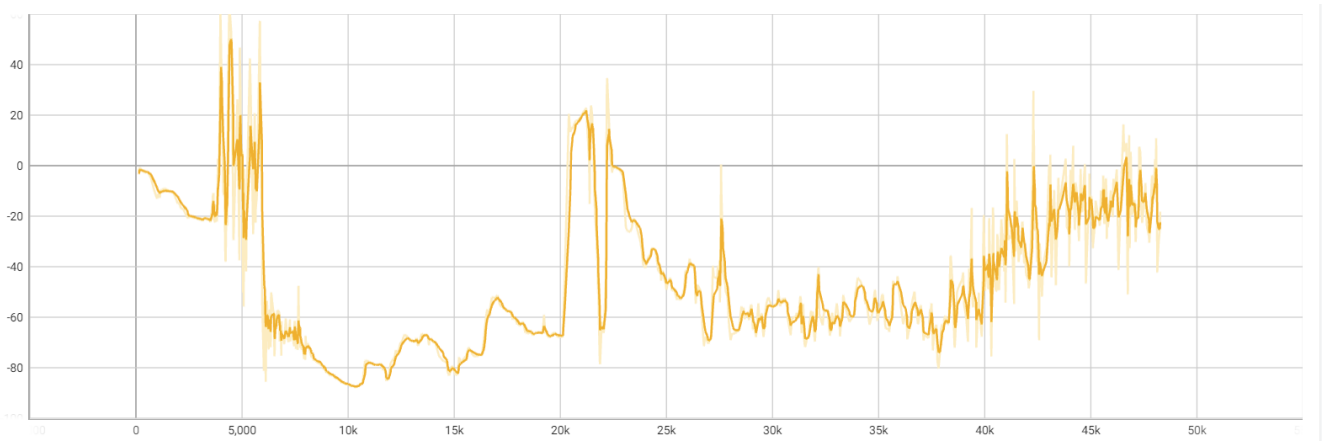

Figure A.1: The loss of generator, using wgangp-eps as loss function.

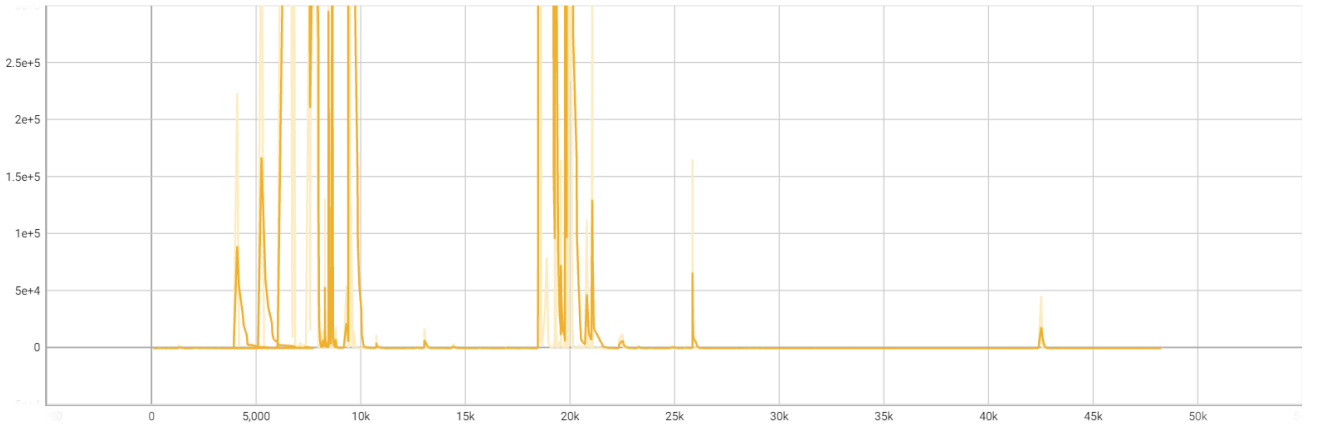

Figure A.2: The loss of discriminator, using wgangp-eps as loss function.

B. Inpainting Results of TransGAN

This part is our current training results of our model for inpainting images in CelebA-HQ-256 and Place365-Standard. Due to the limited time, lack of computing resources and the stochasticity of gaussian noise, our inpainting results are not good enough as expected. Here are some inpainting results with different masks of our models.

Figure B.1: Inpainting results of lsgan loss function, epoch 28, Place365-Standard

Figure B.2: Inpainting results of wgangp-eps loss function, epoch 27, Place365-Standard

Figure B.3: Inpainting results of wgangp-mode loss function, epoch 10, Place365-Standard

Figure B.4: Inpainting results of wgangp-eps loss function, epoch 10, CelebA-HQ-256