Image Multi-Inpainting via Progressive Generative Adversarial Networks

Jiayin Cai Changlin Li Xin Tao Yu-Wing Tai
Kuaishou Technology
{caijiayin, lichanglin, taoxin, daiyurong}@kuaishou.com

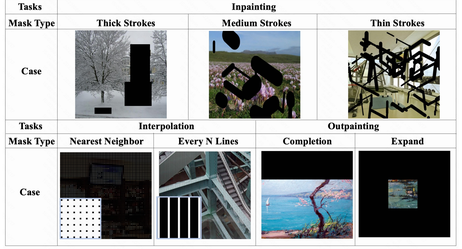
1. Overview

Background: Existing inpainting works:
- Hard to generalize well to multiple inpainting scenarios simultaneously.
- Require specific design for different types of masks.

Motivation:
- We address this problem by proposing a progressive learning scheme to an Semantic Aware Generative Adversarial Network (SA-Patch GAN).

Contributions:
- Progressive learning are applied to the network to make the overall training procedure more stable.
- We use semantic information from a pretrained deep network to enhanced semantic awareness of the discriminator in a Patch-GAN, which is a stabilization our training and improve our performance.
- Our method has achieved the 3rd place on the NTIRE 2022 Inpainting public leaderboard (the 3rd on both PSNR and SSIM) and significantly out- performs existing methods on benchmark datasets.

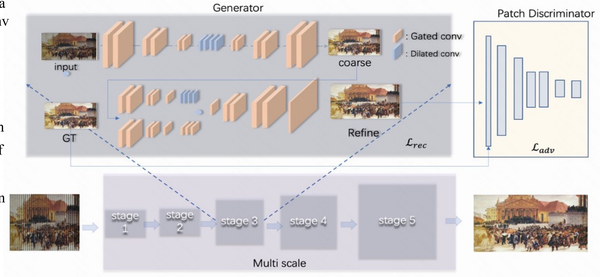
We list some examples of mask in 7 types and split these masks into three kinds of inpainting tasks: Inpainting, Interpolation, Out painting.



2. Methods

Total network.
We progressively apply our backbone in different scales at different image size in a coarse-to-fine manner. We use Coarse-to-fine network architecture with gated conv as backbone. The network architecture of our improved model is shown in Figure.

The network architecture of our improved model



Progressive learning Method
We use multi scales strategy in our method. Firstly, we set the number of stages as three in the $\frac{1}{4}, \frac{1}{2}, \frac{1}{1}$ scales inpainting task. That is, in each stage, the model performs $\frac{1}{4}x \rightarrow \frac{1}{4}x$, $\frac{1}{2}x \rightarrow \frac{1}{2}x$, $1x \rightarrow 1x$ inpainting tasks sequentially. The training starts from stage one, which produces the $\frac{1}{4}x$ scale image from the first stage. After the end of first stage, we unsampled it to $\frac{1}{2}x$ scale and combined it with background pixels from $\frac{1}{2}x$ scale input as the input of stage 2. We froze stage 1 parameters when train stage 2. When we train stage3, the procedure is the same with stage 2. We also found that adding two extra scale $\frac{\sqrt{2}}{4}$x and $\frac{\sqrt{2}}{2}$x after stage 1 and stage 2 could improve model performance.

Semantic aware patch GAN (SA-Patch GAN)
In the original cGAN paper, a one-hot class label y is passed into the discriminator in addition to the image x to be classified as real or fake. The discriminator output is:

$$D(x^*, y) = f_\phi(\phi(x^*)) + \langle \phi(x^*), f_y(y) \rangle \quad (1)$$

ϕ is a learned function mapping an image to a vector. f_ϕ is a learned fully-connected layer that maps that vector to a scalar, f_y is a learned fully-connected layer mapping y to a vector of the same size as the output of ϕ.
We change Eq.1 to add semantic condition:

$$D(x^*, M, x) = f_\phi(\phi(x^*, M)) + \langle \phi(x^*, M), f_C(C(x)) \rangle \quad (2)$$

M is the input mask. The architecture of ϕ is consists of six stride convolutional layers, followed by a fully connected layer. The output dimensions of ϕ and f_C are both 256.

Coarse-to-Fine deep inpainting network
The network architecture of our improved model is shown in Figure. We use progressive learning based on Deepfillv2. We select Deepfillv2 because it achieves a good balance between efficiency and performance.
The model is based on gated convolutions which is used to learn a dynamic feature selection mechanism for each channel at each spatial location across all layers, significantly improve the color consistency and inpainting quality of free-form masks and inputs.

Quantitative results of our proposed model on Partial of Places datasets with different mask types. Our Partial test set contains 1, 000 × 7 × 4 images for seven type of mask on four dataset.

		Strokes			Interpolation		Completion	
	Mean	Thick	Medium	Thin	Every N Lines	Nearest Neighbor	Completion	Expand
PSNR ↑	22.89	23.330	23.992	27.284	31.772	24.873	16.130	12.877
SSIM ↑	0.785	0.866	0.879	0.910	0.940	0.757	0.688	0.454
LPIPS ↓	0.248	0.158	0.134	0.112	0.147	0.347	0.522	0.313
FID ↓	20.314	15.213	12.341	10.214	14.906	21.924	37.484	30.098

Comparing our methods with SOTA out-painting tasks on Places dataset. We provide FID scores since FID correlates with perceptual quality best.

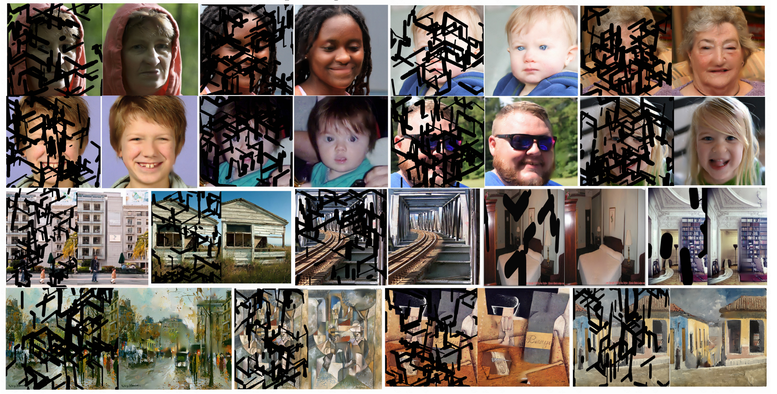
Method	FID ↓
Boundless [22]	35.02
NS-outpaint [29]	50.68
DeepFillv2 [30, 31]	56.14
Image2StyleGAN [1]	25.36
In&Out [3]	23.57
Very_Long [29]	**13.71**
Ours	18.33

Quantitative comparison of our model with SOAT conventional inpainting methods on Places2 validation images (1,000) with irregular masks. † denotes the results are copy from [32]

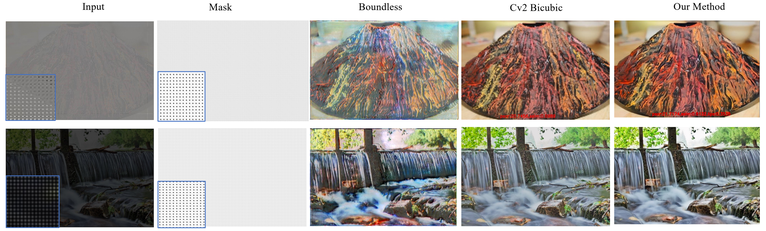
	PSNR ↑		SSIM ↑		FID ↓
Method	Thin	Thick	Thin	Thick	
EC† [17]	26.52	22.23	0.880	0.731	30.13
GC† [31]	26.53	21.19	0.881	0.729	30.13
MEDFE† [15]	26.47	22.27	0.877	0.717	31.40
PIC† [36]	26.10	21.50	0.865	0.680	33.47
ICT† [23]	26.6	23.32	0.880	0.724	25.42
AOT-GAN [33]	26.03	22.62	0.890	0.804	**5.47**
BAT-Fill† [32]	26.47	21.74	0.879	0.704	22.16
pluralistic [36]	26.47	21.74	0.879	0.704	25.42
Ours	**27.28**	**23.33**	**0.910**	**0.866**	18.33

3. Experimental results

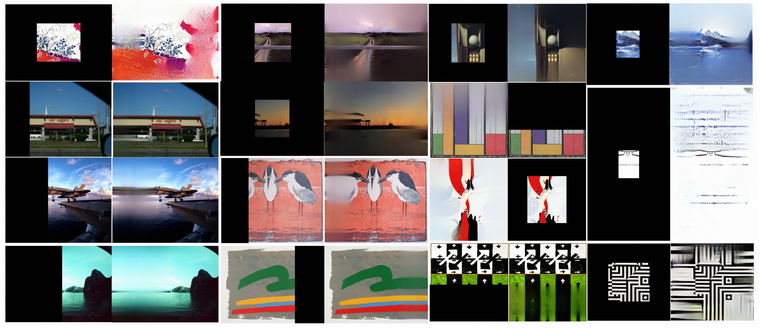
Conventional inpainting Task for stroke masks on four datasets.



Quantitative results of our model on Interpolation task compared to Boundless[22] and CV2 Bicubic interpolation.



Out-painting Task for Completion and Expend masks on four datasets.



Interpolation-inpainting Task for Every N Line and Nearest Neighbor on four datasets.

Ablation study on Partial of Places test set. The test set contains 1,000 images for each type of mask. B means our backbone. B_{3s} demonstrate 3 stage multi-scale progressive learning. and B_{5s} demonstrate 5 stage backbone (adding two extra scale $\frac{\sqrt{2}}{4}$x and $\frac{\sqrt{2}}{2}$x after stage 1 and stage 2). SA means add semantic aware Patch GAN discriminator in the model.

			Strokes			Interpolation		Completion	
		Mean	Thick	Medium	Thin	N Lines	Neighbor	Comp	Expand
whole model	PSNR↑	**22.89**	23.330	23.992	27.284	31.772	24.873	16.130	12.877
	SSIM↑	**0.785**	0.866	0.879	0.910	0.940	0.757	0.688	0.454
	LPIPS↓	**0.248**	0.158	0.134	0.112	0.147	0.347	0.522	0.313
	FID↓	**20.314**	15.213	12.341	10.214	14.906	21.924	37.484	30.098
		Mean	Thick	Medium	Thin	N Lines	Neighbor	Comp	Expand
B_{5s}+SA	PSNR↑	22.01	22.506	23.505	27.127	30.739	22.709	16.280	13.192
	SSIM↑	0.776	0.862	0.877	0.907	0.922	0.675	0.662	0.471
	LPIPS↓	0.260	0.180	0.142	0.131	0.166	0.359	0.540	0.301
	FID↓	21.613	17.201	13.251	12.005	16.211	23.014	38.129	31.482
		Mean	Thick	Medium	Thin	N Lines	Neighbor	Comp	Expand
B_{3s}+SA	PSNR↑	21.89	22.368	23.383	27.022	30.422	22.704	16.295	12.944
	SSIM↑	0.774	0.862	0.876	0.906	0.917	0.670	0.676	0.397
	LPIPS↓	0.263	0.184	0.145	0.133	0.171	0.361	0.520	0.330
	FID↓	21.939	17.512	13.492	12.395	16.592	23.288	38.529	31.771
		Mean	Thick	Medium	Thin	N Lines	Neighbor	Comp	Expand
B+SA	PSNR↑	21.66	23.508	24.066	26.527	28.241	19.778	15.363	12.126
	SSIM↑	0.774	0.856	0.866	0.891	0.876	0.498	0.702	0.488
	LPIPS↓	0.289	0.199	0.166	0.147	0.179	0.371	0.577	0.383
	FID↓	22.530	17.892	13.625	12.766	16.983	23.504	39.733	33.207
		Mean	Thick	Medium	Thin	N Lines	Neighbor	Comp	Expand
B	PSNR↑	20.9	23.196	23.623	25.489	27.227	17.627	15.327	12.056
	SSIM↑	0.670	0.855	0.858	0.875	0.841	0.395	0.703	0.487
	LPIPS↓	0.302	0.204	0.173	0.161	0.199	0.394	0.588	0.401
	FID↓	22.912	18.533	13.935	12.805	17.029	23.881	40.428	33.771

Quantitative results of our proposed model on all of the datasets for four datasets.

	PSNR ↑		SSIM ↑		LPIPS ↓		FID ↓
Datasets	mean	std	mean	std	mean	std	mean
FFHQ	25.06	8.669	0.838	0.147	0.239	0.173	21.345
Places	23.41	7.892	0.787	0.195	0.255	0.193	18.334
ImageNet	23.804	8.781	0.776	0.221	0.249	0.213	18.854
WikiArt	23.142	7.305	0.759	0.204	0.276	0.185	26.395