Intelligent Painter: New Masking Strategy and Self-Referencing with Resampling

Chun-Chuen Hui[1,2], Wan-Chi Siu[1,2] Life-FIEEE, Ngai-Fong Law[2] SrMIEEE, H. Anthony Chan[1]FIEEE
[1]School of computing and Information Sciences, Caritas Institute of Higher Education
[2]Department of Electronic and Information Engineering, The Hong Kong Polytechnic University
Hong Kong, China

***Abstract*—Painting with our own hands is not everyone's talent. Some of us may dream big to create our own artwork but do not have the ability to do so. With the help of deep learning techniques, we nowadays can generate text-based painting. However, just typing text to create our own artwork is still different from doing it yourself (DIY). We proposed an application called intelligent painter, which can let users decide the placement of the objects and use the diffusion models to fill all the gaps after the users finish their placement. In this paper, we propose two major contributions to make better generation of images by (i) a new masking strategy and (ii) speeding up the process by 50% compared with resampling Denoising Diffusion Probabilistic Models (DDPM), with a self-pre-processing input step.**

Keywords— Deep learning; Image processing; Diffusion model; intelligent painter; Image synthesis Introduction

I. INTRODUCTION

Image generation makes use of deep learning approaches which have performed decent results in the past decade, including **G**enerative **A**dversarial **N**etworks (GANs) [1-3], **V**ariational **A**uto**E**ncoders (VAEs) [4-6] and **D**iffusion **M**odels (DMs) [7, 8]. Without any conditioning, these generative models produce images from a latent space. The results depend highly on the training data and usually are limited to a specify type of scenes, such as human face or bedroom etc. Conditioning methods [8-10], no matter by texts or images, can guide the generation process to fit for the target result of the users. However, sometimes users may just want to choose specific objects and post them on preferred locations, just like drawing by their own hands.

We have proposed an Intelligent Painter, inputs and corresponding results are shown in Fig.1. Unlike other conditioning methods that do not allow users to insert the objects freely. Our Intelligent Painter is a novel idea that allows users to import their preferred objects and decide their preferred locations, sizes and tilt angles; and even the user can also rank the priority of each object, meaning that some of the objects can be put in front of another object. Just as shown in Fig.1, the bamboo tree can overlap and be in front of the house, and our painter can produce good result with it. Our early approach made [11] use of masking and resampling method [12]. We propose in this paper to add early stopping of resampling to obtain better image quality. Note that inpainting is targeted to reconstruct missing area using features from the known part of the input, e.g. a human face with only eyes covered. Inpainting models can use training results of human faces to recover the eyes. Meanwhile in our work, we allow users to insert a few individual objects into a plain paper, and use diffusion approach to fill up the gap in order to link the objects. Note that in this case objects selected may not be consistent to each other. Resampling [12] produces harmonized inpainting results by repeating back-and-forth the diffusion processes, which are noise adding and denoising processes. Using resampling method indeed can generate acceptable scene to fill up the unknown area. Our focus is: Does the result generated fit the scene? At the same time, resampling with diffusion model is very time consuming. For example, using one single RTX3090 graphic card with 2770 steps, it requires around 220 seconds to generate one image.

In this paper, we have two major contributions. 1) With a new masking strategy, we can produce results that are consistent between inserted objects and the gap (say with NIQE -8.51%). 2) We propose to use self-referencing to pre-process the input

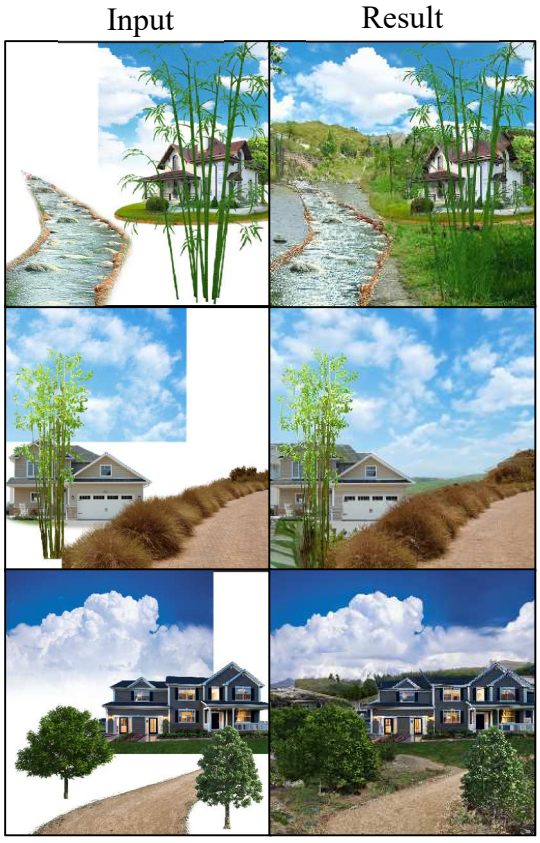

Fig. 1. Using our intelligent painter, users can determine any object to be inserted. Not only the positions of objects, but the size, tilt angle and object overlapping (above or below another object) can also be adjusted freely.

979-8-3503-3959-8/23/$31.00 ©2023 IEEE

and successfully lower the required resampling steps by 50% while maintaining consistence of the picture, hence the new processing time is just half of the former approach, with even better-quality pictures.

II. Related Work

A. Denoising Diffusion Probabilistic Model (DDPM)

To generate a high-quality image, DDPM can be used, which has been confirmed in many related works [13]. Thanks are given to its repetitive design base on Markov chain. In DDPM, there are two processes: they are the forward (add noise) and reverse (denoise) processes which are shown below,

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}N \quad (1)$$

or

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}N \quad (2)$$

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t N \quad (3)$$

Eqns. 1 and 2 indicate the forward process, where x_0 is the input image and x_t is the input image at timestep t. With larger timestep t, the noisier the image is. N is the Gaussian noise with mean=0, and variance=1. $\sqrt{1-\beta_t}$ and $\sqrt{\beta_t}$ are weights for x_t and N respectively. We usually use eqn. 2 in most of the time, because in eqn. 2, forwarding to timestep t only requires one calculation step while eqn. 1 requires t calculation steps to reach timestep t. This is achieved by letting $\alpha_t = (1-\beta_t)$ and $\bar{\alpha}_t = \prod_{t=1}^{T} \alpha_t$. Note that the sum of accessing Gaussian distribution N is still within mean=0 and variance=1. Thus in eqn. 2, the normal distribution N is also with mean=0 and variance=1, just the same as eqn. 1.

Eqn. 3 indicates the reverse process, also known as the denoising process. By multiple steps of denoising, DDPM can gradually generate a real-looking image from a Gaussian noise input, where $\mu_\theta(x_t, t)$ is the mean of the input image while $\sigma_t N$ stands for the variance. They can be denoted as,

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) \quad (4)$$

$$\sigma_t \mathrm{N} = \sqrt{\frac{\beta_t(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}}\ N \quad (5)$$

where $\epsilon_\theta(x_t, t)$ is the predicted noise and we use U-net [14] to do this part. In every reverse step, an intermediate noisy image and timestep t are inputted into a U-net which was trained to possess features of a set of data, and eqns. 3, 4 and 5 are used for denoising until timestep t countdown to 0. For variance $\sigma_t N$, because of the Gaussian noise N, it gives randomness during the denoising process thus the results vary with different trials of running. Note that DDPM works on reversing the noise-adding process with the help of the trained U-Net as noise predictor. A trained U-Net reflects the characteristics of the data forming the U-Net and implies almost an infinite number of images that hides behind the latent space. In each step, an intermediate image is refined and smoothen out with the noise pattern generated by the U-net to reflect more the characteristics of the original feature set. Hence, performance of the DDPM is based on the parameters trained in U-Net.

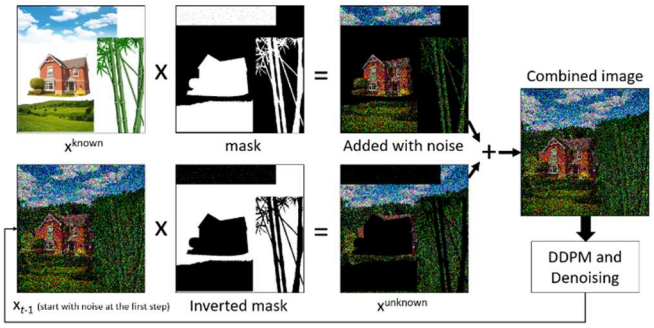

Fig. 2. Flow diagram of using masking in DDPM reverse process.

B. Masking

To fulfill the task of filling in the gap left over after the user who has placed the objects in a "paper", masking is needed for keeping the objects while generating prediction for the unknown area at the same time, such as other inpainting models do [12, 15, 16]. Painting filled by user is annotated as x^{known} while unfilled part is $x^{unknown}$. The aim for our intelligent painter is to predict $x^{unknown}$ with the information of x^{known} and to form a consistent picture. The use of masking is to make sure the content of x^{known} not be changed by the generative model while allowing the model to make change on $x^{unknown}$.

Fig.2 gives a flow diagram to illustrate the use of masking for a single step of the picture production. The step starts with x^{known}, with its known part being identified by the mask and then a new Gaussian noise is added (for denoising) to it in this t^{th} timestep. At the same time, an inverted mask is used to extract the generated result in the t^{th} step. With the start of Gaussian noise at first, it is then multiplied with the inverted mask to produce a gap-only image. These two steps are combined (added) to form a new result ready for denoising, $x^{predict}$. The combined image is then proceeded to do denoising. The result after denoising will be multiplied with inverted mask again for the next iteration.

C. Resampling

We chose DDPM for our Intelligent Painter because of its good performance. Meanwhile, Repaint [12] was the first proposed state-of-the-art inpainting method developed from DDPM. The contribution of this model is to use resampling. This technique allows for the development of more harmonized textures, which can be used to create a more realistic output image. Therefore, resampling is a great method when it comes to image repair and scene reconstruction. During the image generation, when the model is denoised j steps, it adds noise again to the intermediate result by j steps using eqn.1. For instance, when j =10, and the image is denoised from steps 200 to 190. Noise is added again to step 200 using eqn. 1 by ten times and the model continues the reverse process.

The major idea of resampling is to reuse the intermediate result for better consistency, and resampling can help the known part and unknown part combining together smoothly. Without resampling, the DDPM cannot fill unknown areas well making use of the known part. The texture can be considered

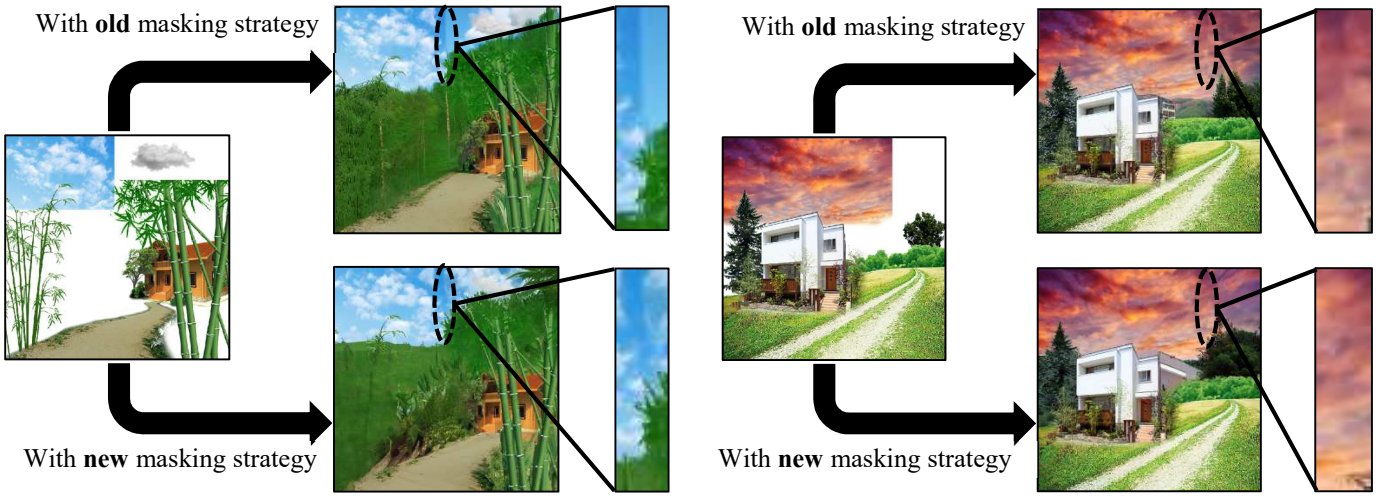

Fig. 3. Figure shows the difference between using old masking strategy (masking every timestep) and new masking strategy (masking fits the time condition). It shows that while using the new masking strategy, the connection part between x^{known} and $x^{unknown}$ is smoother than those using the old masking strategy.

fit, but the image may not look reasonably well. Anyway, with the known texture obtained previously by prediction, the DDPM can do better prediction through the resampling process.

III. Methodology

A. Intelligent Painter

For our intelligent painter, we use DDPM masking and resampling just like RePaint [12], but add an early stopping for clearer and better results [11]. The painter starts denoising at 250, resampling 10 times per 10 steps, and stops resampling at step 100.

As shown in Fig. 3, there is a stitching problem between x^{known} and $x^{unknown}$ and it is more obvious when the object is the sky. Two examples are shown in Fig. 3, (see the dotted circles). The upper result is generated using the masking method we mentioned in Section 2.2. Let us make an analysis on this masking technique. First, as we discussed in Section 2, x^{known} is firstly multiplied with the mask and new Gaussian noise is added in every timestep *t*; meaning that in every reverse step, the known part may not be consistent with unknown part during the denoising, since they are separately processed in every step. The prediction based on x^{known} in timestep *t* is not inherited into *t*-1 and *t*-2 etc. This is the first inconsistent point. Second, as x^{known} is fixed and the only changing part is $x^{unknown}$, this is for sure inconsistent if we compared it with x^{known} which has also been enabled to change with the prediction from the model.

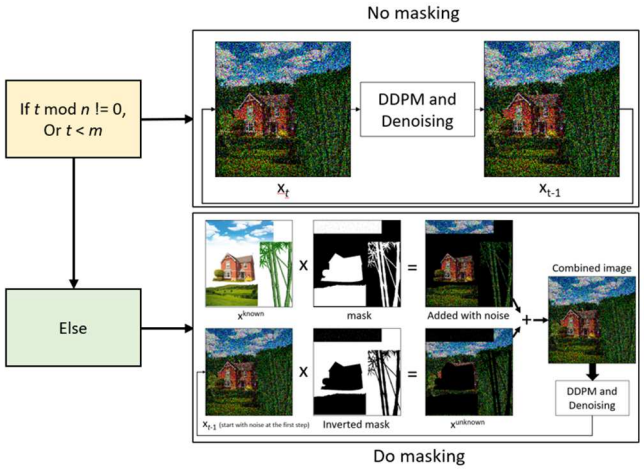

Fig. 4. Flow diagram of using our new masking strategy.

B. New masking strategy

Due to the inconsistency brought by the above masking strategy, we propose another new masking strategy that can better suit both x^{known} and $x^{unknown}$. Fig. 4 shows the diagram of our new masking strategy. This new masking strategy allows the model to make use of the previous predictions and enables x^{known} to change with $x^{unknown}$ at the same time by minimizing the time of using masking. When *t* goes through *m* steps, the model uses masking to refill the content of our painting and completely stops masking at *n* step. When the conditions are not fulfilled, no masking is used and the denoising process of the painter is just like what normally does by the DDPM.

C. Pre-self-referencing

From the above evidence, we conclude that the resampling is in fact doing self-reference by the intermediate result. Yet, the resampling method can highly improve the image quality and can do the job of inpainting successfully. But with the number of resampling increased, the number of steps of the model to generate one result is also enormously increased. It is very time consuming for some applications. Since we consider this as self-referencing, let us propose a pre-process method to help the model instead of using many resampling steps.

Our idea is to shift the objects that user placed to its nearby area and put Gaussian noise in the unknown part. The pre-process diagram is shown in Fig. 5. It is a simple yet effective idea to replace hundreds of resampling steps into 1 pre-process step.

D. Overall

With the use of this new masking strategy and pre-processing to help self-referencing, we substantially have improved the painting quality and increased the speed of operation. In Section 4, we will show our experimental results and some comparisons.

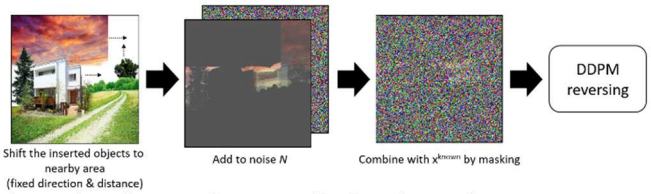

Fig. 5. Flow diagram of our pre-self-referencing. It does a one-step pre-process to replace 1400 steps of resampling.

IV. Experimental Results and Discussion

A. Experiment details

We used a pre-trained model which was trained by the dataset Place2 [17] to do testing on a single RTX3090 GPU using Pytorch. Diffusion started at timestep 250, we did 10 resampling per 10 timesteps and stopped resampling at step 100. The number of total steps for one image generation was 2770 steps and the generation time was 220s.

B. Metric

Let us compare the image quality between results using the previous method and our new method with the new masking strategy and self-referencing. We set both *m* and *n* to 10 for the new masking strategy. In addition, with the use of our pre-self-referencing method, we can reduce half of the resampling steps, and the total number of steps is reduced to 1370. Table I lists the average score with metrices MANIQA [18] and NIQE [19] based on 30 results generated from 10 input paintings. Three generation results were obtained for each input. The 3 results were generated with different sets of noises randomly set by the computer.

MANIQA is an automated image quality assessment method which uses a domain-specific learning approach to evaluate the quality of an image, while NIQE is a non-reference image quality assessment method which measures the naturalness of an image. MANIQA places more emphasis on analyzing certain elements, such as noise, sharpness, and color of an image, while NIQE evaluates the overall quality of an image, by considering its global features, such as luminance, contrast, and structure. All in all, MANIQA focuses more on local image quality while NIQE focuses more on the overall structure of an image.

As shown in Table 1, the new masking strategy increases the score of MANIQA by 0.0014 while the NIQE is decreased by 0.06936. These mean that the image quality is better in both nature quality and overall structure of the image. The new masking strategy helps more on the content details especially the stitching problem that we have discussed in Section 3.1 and shown in Fig. 3. Meanwhile, with the use of pre-self-referencing, not only half of the resampling steps can be saved, but the image quality is also increased. For doing pre-processing, the metrics show that it helps more on NIQE than help on MANIQA, meaning that the overall structure is much better. This fits the target of resampling, harmonizing the known area and unknown area. The improvement of NIQE proves that our pre-self-referencing has achieved the same result with resampling and does the job better and faster.

TABLE I. Metrics comparison between RePaint and our methods used.

Method (steps no.)	MANIQA (↑)	NIQE (↓)
RePaint (2770)	0.70276	4.82256
New masking (2770)	0.70416	4.75320
New masking (1370)	0.70459	4.65886
New masking + pre-self-referencing (1370)	0.70498	4.46310

C. Computation time

As we discussed in last Section that we have reduced the number of resampling steps by half, from 10 resampling per 10 steps to 5 resampling per 10 steps. The computation time also decreased from 220s to 110s, which means half of the time is shortened or the speed is increased by 50%. During our testing, we also generated some image results using 5 resampling per 10 steps but not doing any pre-process to prove that our pre-self-reference is really useful. Fig. 6 shows that without using the pre-processing, there are some unreasonable results while after using of pre-processing, this problem has gone. Pre-self-referencing can reduce the computation time and retain the image consistency at the same time.

V. Conclusion

We have developed a new usage of diffusion called intelligent painter, which can let users decide the components they want, and input and place with any tilt angle and size. Objects can partly over-lapped in a picture. We expect to give more freedom to the user to let them draw their own painting. Initially RePaing approach was directly used to carry out intelligent printing work. However, during our testing, we found the problem of bad connections between inserted objects and generated area. Moreover, the generation time is long as well. Therefore, we have proposed a new masking strategy to solve the this problem and use pre-self-referencing to shorten the computation time. Results from our experimental work prove that the improvement is significant. We have also done some work to compare our results with non-diffusion models such as AOT-GAN [20] and Big-LaMa [21] which cannot give high quality results especially for inputs with large empty space (details not provided, for limited paper space), whereas diffusion models can usually do better.

Acknowledgment

This work is partly supported by the Caritas Institute of Higher Education (ISG200206) and UGC Grant (UGC/IDS(C)11/E01/20) of the Hong Kong Special Administrative Region.

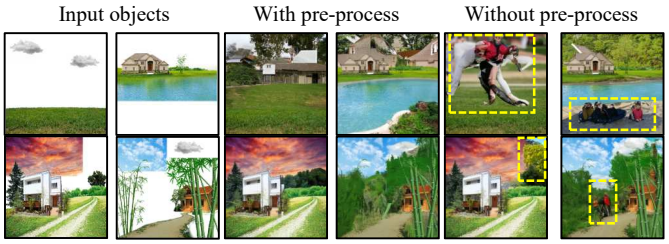

Fig. 6. Comparison results between using pre-process and not using it when running in 5 resampling per 10 steps, in a total of 1370 steps.

REFERENCES

[1] I. J. Goodfellow et al., "Generative Adversarial Networks," Transaction in ACM Transaction on Communications, vol 63, no.11, pp. 139-144, Nov. 2020

[2] Xiaolong Wang and Abhinav Gupta, "Generative image modeling using style and structure adversarial networks," Proceedings, pp. 318-335, Computer Vision–ECCV 2016: 14th European Conference, 11-14 Oct, Amsterdam, The Netherlands.

[3] Tero Karras, Samuli Laine and Timo Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," Proceedings, pp. 4396-4405, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 16-20 June, Long Beach, CA, USA.

[4] Diederik P. Kingma and Max Welling, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.

[5] Zhi-Song Liu, Wan-Chi Siu, Li-Wen Wang, Chu-Tak Li, Marie-Paule Cani and Yui-Lam Chan, "Unsupervised real image super-resolution via generative variational autoencoder" Proceedings, pp. 442-443, 2020 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPR), 16-18 June.

[6] Zhi-Song Liu, Vicky Kalogeiton, Marie-Paule Cani, "Multiple Style Transfer Via Variational Autoencoder", Proceedings, pp. 2413-2417, IEEE International Conference on Image Processing (ICIP 2021), 19-22 September, Anchorage, AK, USA.

[7] Jonathan Ho, Ajay Jain and Pieter Abbeel, "Denoising diffusion probabilistic models," Proceedings, pp. 6840-4851, 2020 Conference on Neural Information Processing Systems, 6-12 Dec.

[8] Prafulla Dhariwal and Alex Nichol, "Diffusion models beat gans on image synthesis." Proceedings, pp. 8780-8794, 2021 Conference on Neural Information Processing Systems, 6-14 Dec.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser and Bjorn Ommer, "High-resolution image synthesis with latent diffusion models," Proceedings, pp. 10684-10695, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 19-24 June, New Orleans, Louisiana, USA.

[10] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon and Sungroh Yoon, "ILVR: Conditioning Method for Denoising Diffusion Probabilistic Models," Proceedings, pp. 14347-14356, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 11-17 Oct, Montreal, QC, Canada.

[11] Wing-Fung Ku, Wan-Chi Siu , Xi Cheng and H. Anthony Chan, "Intelligent painter: picture composition with resampling diffusion model" arXiv preprint arXiv:2210.17106, 2022.

[12] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "RePaint: Inpainting using Denoising Diffusion Probabilistic Models," Proceedings, pp. 11461-11471, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 19-24 June, New Orleans, Louisiana, USA.

[13] YANG, Ling, et al. "Diffusion models: A comprehensive survey of methods and applications," arXiv preprint arXiv:2209.00796, 2022.

[14] O. Ronneberger, P. Fischer, and T. Brox, 'U-Net: Convolutional Networks for Biomedical Image Segmentation'. Proceedings, Part III 18, pp. 234-241, Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, 5-9 Oct, Munich, Germany.

[15] Roman Suvorov, et al., "Resolution-robust large mask inpainting with fourier convolutions," Proceedings, pp. 2149-2159, 2022 IEEE/CVF conference on applications of computer vision (CVPR), 19-24 June, New Orleans, Louisiana, USA.

[16] LI, Wenbo, et al. "Mat: Mask-aware transformer for large hole image inpainting," Proceedings, pp. 10758-10768, 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR) , 19-24 June, New Orleans, Louisiana, USA.

[17] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba, "Places: A 10 million Image Database for Scene Recognition," IEEE Transactions on pattern analysis and machine intelligent, vol. 40, no. 6, pp. 1452-1464, 2017.

[18] Sidi Yang, et al. "Maniqa: Multi-dimension attention network for no-reference image quality assessment," Proceedings, pp. 1191-1200, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 19-24 June, New Orleans, Louisiana, USA.

[19] Anish Mittal, Rajiv Soundararajan and Alan C. Bovik, 'Making a "Completely Blind" Image Quality Analyzer', IEEE Transactions on Signal Processing Letter, vol. 20, no. 3, pp. 209–212, Mar. 2013.

[20] Yanhong Zeng, Jianlong Fu, Hongyang Chao and Baining Guo, "Aggregated Contextual Transformations for High-Resolution Image Inpainting," in IEEE Transactions on Visualization and Computer Graphics, doi: 10.1109/TVCG.2022.3156949.

[21] Roman Suvorov et al., "Resolution-robust Large Mask Inpainting with Fourier Convolutions," 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2022, pp. 3172-3182