Hidden Object Masking using Deep Learning

Qing Bu
CETC Les Information System Co., Ltd
39020765@qq.com

Wei Wan
CETC Les Information System Co., Ltd
1271130252@qq.com

Ivan Leonov
Belarusian State University
Faculty of Applied Mathematics and Computer Science
Minsk, Belarus
ivanleonov.eu@gmail.com

***Abstract*—Image inpainting, the process of filling in missing or damaged regions within images, has witnessed a significant evolution in recent years, driven primarily by deep learning methodologies. This paper provides an overview of modern architectures used for image inpainting, and addresses how they can be applied to protect sensitive information.**

Keywords—Image Inpainting, WGAN, Generative adversarial network, WGAIN, Image Imputation

I. Introduction

Image inpainting is one of the most important tasks in computer vision(CV), which is equivalent to image completion. It finds application in various domains, from science to industry. The fundamental task of image inpainting is to restore damaged or occluded regions so that the proposed patch seamlessly completes regions.

The main task of image inpainting is to fill missing areas by the information present on the image and can be perceived as inverse problem[13]. Conventional approaches work well on small damages by focusing on statistics and pattern matching[14], which has limits in terms of robustness when a more complex scene and contextual representation is lacking. Another challenge for the classical approach is a large gap. A more recent and popular approach to the problem is convolutions neural networks(CNNs).

In the following sections, we will delve into the key components of image inpainting, including data-driven approaches, the role of convolutional neural networks (CNNs), generative adversarial networks (GANs), and the critical issue of evaluating inpainting results. We will also discuss practical applications of image inpainting across diverse domains, underscoring its role as an enabling technology in contemporary society.

II. Related Work

A. Classical approaches

Classical approaches appeared in the early 2000s after Bertalmio et al.[16] introduced the basic algorithmic approach based on the techniques used by professional art restorators, based on distance fields. Over the next decade, many more approaches were introduced which were based on statistics of the images. [5], [7], [20]

However, it has limitations with larger gap sizes. A break through in efficiency and performance was done by Barnes et al. [14] with a tool PatchMake which optimized the performance of patch filling. Despite covering larger areas, it still fails with context-aware patches. Despite all the advancements traditional methods still fail with semantic information of the images and that is where deep learning approaches surpass them.

B. Deep learning approaches

The deep learning field has witnessed rapid growth since the introduction of AlexNet in 2012. The breakthrough for image inpainting was done by Pathak et al. [4], after which the number of works in the field increased exponentially from year to year. The base principle is the presence of an encoder which captures the content representation of the scene into latent feature representation and a decoder which subsequently decodes it into a restored image. There are two main classes of deep learning models used for image inpainting, which are CNNs and GANs, although other architectures like recursive neural networks (RNNs) are also sometimes introduced.[9]

CNNs are a fundamental building block in many inpainting architectures. They are used to extract features from both the known and surrounding regions of the image. For inpainting, you can mask the missing region in the input image and use the encoder-decoder architecture to generate the missing content. Attention mechanisms are often integrated into deep inpainting networks to allow the model to focus on relevant parts of the image when generating missing content. Self-attention mechanisms, like those used in Transformer architectures, can help capture long-range dependencies and improve inpainting quality.

After generating the inpainted image, post-processing techniques can be applied to enhance the final result, by blending the completed region with the color of the surrounding pixels. In particular, the fast marching method [20], followed by Poisson image blending [21] demonstrates promising results.

Deep learning-based image inpainting has made significant strides in producing realistic and visually pleasing results. Researchers continue to explore novel architectures and training strategies to further advance the state-of-the-art in this field.

III. Modern Architectures

The rapid advancements in the realm of AI-generated content have brought about new techniques, revolutionizing the way we approach image inpainting problems. In this section, we delve into the modern algorithms that utilize deep learning models, allowing high-quality realistic image generation. The ability to understand the global context of an image and successfully impute missing regions characterize these cutting-edge models.

A. Irregular Mask Inpainting

NVIDIA's Inpaiting model [1] was specifically designed for image inpainting tasks involving irregular masks. The introduction of partial convolutions is the core innovation of this model. These convolutions were developed to allow the network to effectively process irregular masks. They ensure

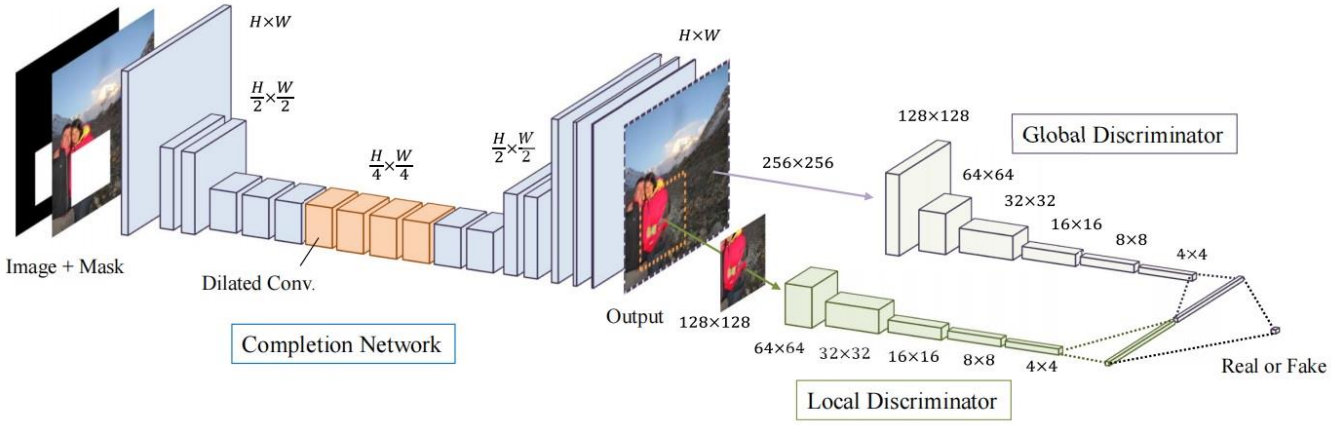

Fig. 1 Overview of an architecture with two discriminators. The global discriminator network takes the entire image as input, while the local discriminator network takes only a small region around the completed area as input. Both discriminator networks are trained to determine if an image is real or completed by the completion network, while the completion network is trained to fool both discriminator networks.

that only valid information from visible parts of the image is used to generate the inpainting for the masked region.

Let W be the convolution kernel and b its corresponding bias. Assume X be the pixels for the convolution windows and M -- the corresponding binary mask. The partial convolution is defined as:

$$x' = \begin{cases} W^T(X \odot M)\dfrac{sum(1)}{sum(M)}, & \text{if } sum(M) > 0 \\ 0, & \text{otherwise} \end{cases}$$

Where $\odot$ denotes the element-wise product. The convolution results depend only on unmasked inputs. The scaling factor $\frac{\text{sum}(1)}{\text{sum}(M)}$ applies scaling corresponding to the varying amount of unmasked pixels.

The model has a U-Net-like architecture, where partial convolutions replace every convolutional layer. The last partial convolution layer's input will contain the concatenation of the original input image with hole and original mask, making it possible for the model to copy non-hole pixels.

The significance of the loss function cannot be overstated, as it plays a foundational role in shaping the results of image inpainting. Loss functions are a mathematical metrics that quantify the dissimilarity between the imputed image and the ground truth. In addition to their role in model training, they can be designed to enforce the generation of realistic and locally consistent inpaintings. Furthermore, the choice of loss function allows task-specific optimizations. Depending on the objective, the loss function can prioritize various aspects, such as contextual coherence or style.

The proposed loss function targets both per-pixel reconstruction as well as seamlessness of the resulting image. In the paper, they use a variety of loss function in order to calculate the total loss. First, the per-pixel losses $\mathcal{L}_{hole}, \mathcal{L}_{valid}$ are calculated. These are the L^1losses for the hole and valid pixel respectively, calculated on the final inpainting. The total variation loss $\mathcal{L}_{TV}$ [3] acts a smoothing penalty on a 1-pixel dilation of the mask region. Last but not the least, the the style loss ($\mathcal{L}_{style}$) and perceptual loss($\mathcal{L}_{percept}$)[3] are calculated:

$$\mathcal{L}_{total} = \mathcal{L}_{valid} + 6\mathcal{L}_{hole} + 0.05\mathcal{L}_{percept} + 120\mathcal{L}_{style} + 0.1\mathcal{L}_{TV}$$

Nowadays, state-of-the-art models may include adversarial loss and Image-Dependent Markov Random Field (ID-MRF) loss terms. By combining these loss functions, the model can learn to produce images that not only exhibit spatial coherence and smoothness but also capture the fine-grained details and content of the input image, resulting in visually convincing results.

The loss term weights are determined by performing a hyperparameter search on a subset of validation images.

Holes present a problem for Batch Normalization since the mean and variance will be computed for masked regions. However, as we progress through each layer, the missing pixels are steadily filled, typically becoming entirely absent once we reach the decoder stage. Therefore, we can either perform two-phase training: train with Batch Normalization, then freeze batch normalization layers in the encoding part and fine-tune the model. Moreover, removing batch normalization at all is also an option, since such models train on big datasets, meaning small batch size.

B. Wasserstein Generative Adversarial Imputation Network

Training Generative Adversarial Networks for image inpainting poses several challenges. GANs are known for mode collapse, training instability, and convergence issues, often resulting in poor image quality and mode dropping. Wasserstein Loss offers a more stable and informative objective function compared to traditional GAN losses like the Jensen-Shannon divergence, making it suitable for GAN training. Additionally, usage of gradient penalty via techniques like gradient clipping or norm clipping helps prevent discriminator gradients from exploding, thereby providing stable training.[19] This regularization technique encourages the generator to produce more diverse and realistic samples, mitigating mode collapse issues.

The Wasserstein Generative Adversarial Imputation Network(WGAIN) implements this approach. [18] They used Wasserstein GAN as a generator with norm clipping to satisfy the Lipschitz constraint. During the training phase, three types of missingness were used: noise, single square in the center and randomly located multiple squares. This combination of different mask types allows us to effectively apply the trained model for hiding private information in an image. Thus, the user provides a mask made with multiple square regions, and

the trained WGAIN model repaint areas containing sensible information.

C. Globally and Locally Consistent Image Inpainting

Achieving both a global and local consistency is crucial for image inpainting since it ensures that the completed regions seamlessly blend into the overall context of the image while preserving the fine-grained details and textures. Without both aspects, inpainted regions may stand out as unnatural, disrupting the overall visual experience and failing to meet the expectations of viewers. Therefore, a successful image inpainting method must strike a balance between preserving the global context and maintaining local details and textures.

A simple modification of GAN architecture allows to address both aspects at the same time. Two discriminators are used in order to achieve global and local consistency. (Fig. 1) The global discriminator focuses on capturing the larger context of the image to ensure global consistency. It generates an initial estimate of the completed image. Meanwhile, the local discriminator, which refines the initial estimate by focusing on the details and textures within the image. This network helps achieve local consistency and ensures that the completed regions blend seamlessly with the existing content. The results from both discriminators are used to make the final decision.

In order to improve training stability, the some modification to the generator were enforced. The generator input consists of the masked image and the mask. Therefore, the training procedure is more stable, since random noise doesn't play any role in the generation. The generator may be conditioned on the known parts of the input image and the mask indicating the regions to be inpainted. This conditioning helps the generator focus on the specific task of completing the missing regions while considering the context.

D. Inpaint Anything

Nowadays, the state-of-the-art (SOTA) image inpainting models, like LaMa[10], Repaint[11], MAT[12], ZITS[15] have demonstrated exceptional performance. These models are capable of effectively inpainting large regions, handling complex patterns, and working well on high-resolution images. However, they usually rely on detailed masks.

Segment Anything Model(SAM) is a SOTA model from Meta AI that can create segmentation masks for any object, in any image. It can be used to generate accurate masks for all objects in an image. Thus, using an ensemble of SAM and SOTA inpainters we can create a model for removing any object from an image.

Inpaint Anything [16] allows users to easily remove objects from an image with a single click. Moreover, the proposed ensemble provides an opportunity to fill the selected region with realistic computer-generated images. In addition, the SOTA inpaiter can be replaced with a different SOTA model. To illustrate, combining SAM with Stable Diffusion (SD)[17] results in a "Fill Anything" model, giving the end user more control over the final inpainting.

This approach allows us to address a vast variety of computer vision problems: content restoration, privacy protection, and real-time image manipulation. When a portion of an image is removed, the Inpaint Anything model can restore the missing region seamlessly. Therefore, we can address the protection of private data. In scenarios where sensitive information needs to be protected, such regions can be selected using the SAM model and later imputed with SOTA inpainter, such as LaMa. Thus, the Inpaint Anything model and be used intelligently to inpaint over ”secret” areas to protect privacy.

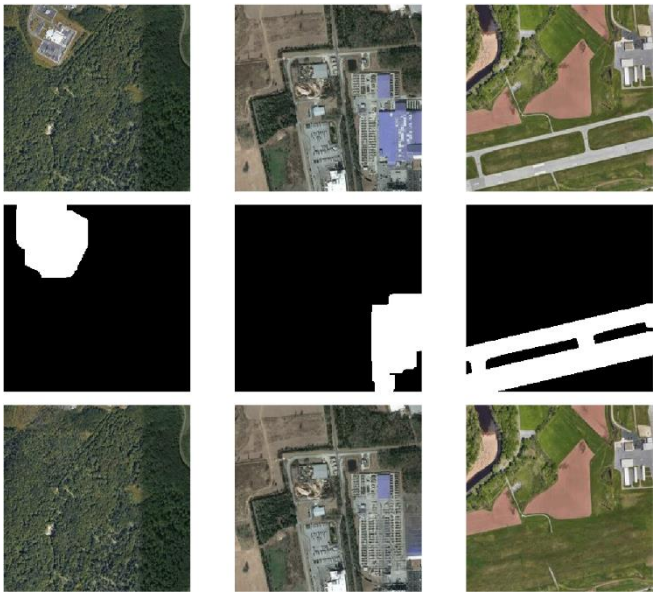

Fig. 2 SAM + LAMA results.The first row demonstrates original images, the second row -- masks, obtained from the SAM model. The third row shows the resulting inpaintings. The results were obtained on pretrained models, and can be enhanced with fine-tuning.

The model can ensure the inpainted areas maintain visual consistency with the surrounding content, to create high-resolution and natural-looking results. Since there are mobile versions of the SAM, the Inpaint Anything model can used interactively using mobile phones. Therefore, this ensemble model can further develop image editing software.

IV. Conclusion

We have represented modern approaches to image inpainting. These models can be applied to solve a vast variety of problems, such as object removal and protection of private information. We argue that the simplest approach to use for this goal is the Inpaint Anything model. The paint anything has two phases: segmentation and inpainting. The segmentation phase is done through the state-of-the-art Segment Anything model. This model and be easily fine-tuned to better suit the given dataset. Thus, we can improve the performance of the resulting model. Meanwhile, the inpainted phase is covered via a SOTA inpainting model such as LaMa or the Stable Diffusion. The inpainting models can be fine-tuned as well. Meaning that we have control over the quality and style of the inpainting. Therefore, we can provide the end user with a suitable interface in order to give control over the final inpainting.

References

[1] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions” in European Conference on Computer Vision (ECCV), 2018, pp. 85–100.

[2] L.A. Gatys, A.S. Ecker, M. Bethge, “A neural algorithm of artistic style.” 2015, arXiv, preprint.

[3] J. Johnson, A. Alahi, L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution.” in: European Conference on Computer Vision. Springer, 2016 pp. 694–711.

[4] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell and A. A. Efros, "Context Encoders: Feature Learning by Inpainting,”*2016 IEEE*

Conference on Computer Vision and Pattern Recognitio, Las Vegas, NV, USA, 2016, pp. 2536-2544

[5] A. Criminisi, P. Perez and K. Toyama, "Object removal by exemplar-based inpainting," *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings.*, Madison, WI, USA, 2003, pp. II-II

[6] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image inpainting. In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., USA, pp. 417–424.

[7] M. Bertalmio, L. Vese, G. Sapiro and S. Osher, "Simultaneous structure and texture image inpainting," in *IEEE Transactions on Image Processing*, vol. 12, no. 8, , Aug. 2003, pp. 882-889

[8] A. Telea, "An image inpainting technique based on the Fast Marching Method," Journal of Graphics Tools, vol. 9, no. 1, 2004, pp. 23–34

[9] A. Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel Recurrent Neural Networks," CoRR, 2016.

[10] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions." in Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2022, pp. 2149–2159.

[11] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte and Luc Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11461–11471.

[12] Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: "Mask-aware transformer for large hole image inpainting." in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp 10758–10768

[13] A. Lucas, M. Iliadis, R. Molina, and A. K. Katsaggelos, "Using deep neural networks for inverse problems in imaging: Beyond analytical methods," IEEE Signal Processing Magazine, vol. 35, no. 1, 2018, pp. 20–36.

[14] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized patchmatch correspondence algorithm," Computer Vision – ECCV 2010, 2010, pp. 29–43

[15] Qiaole Dong, Chenjie Cao, and Yanwei Fu. "Incremental transformer structure enhanced image inpainting with masking positional encoding." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 11358–11368

[16] Tao Yu, Runseng Feng, Ruoyu Feng, Jinming Liu, Xin Jin, Wenjun Zeng, Zhibo Chen., "Inpaint Anything: Segment Anything Meets Image Inpainting.",2023, arXiv, preprint

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. "High-resolution image ¨synthesis with latent diffusion models." in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10684–10695.

[18] Vašata, D., Halama, T., Friedjungová, M. "Image Inpainting Using Wasserstein Generative Adversarial Imputation Network." in: Farkaš, I., Masulli, P., Otte, S., Wermter, S. (eds) Artificial Neural Networks and Machine Learning – ICANN 2021.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. 2017. "Improved training of wasserstein GANs." in Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, pp. 5769–5779.

[20] Alexandru Telea. "An Image Inpainting Technique Based on the Fast Marching" in Method. Journal of Graphics Tools 9, 1, 2004, pp 23–34.

[21] A. Criminisi, P. Perez, and K. Toyama. "Region Filling and Object Removal by Exemplar-based Image Inpainting."in IEEE Transactions on Image Processing 13, 9, 2004, pp. 1200–1212.)