DIP Project Report: Image Inpainting

Chong Li
Wuhan University, EIS
chongli@whu.edu.cn

Wenli Xiao
Wuhan University, EIS
xiaowenli@whu.edu.cn

Yifan Xia
Wuhan University, EIS
xiayifan@whu.edu.cn

Abstract— Image inpainting is the process of completing or recovering the missing region in the image or removing some object added to it. With the improvement of image processing tools and the flexibility of digital image editing, automatic image inpainting has found important applications in computer vision and has also become an important and challenging topic of research in image processing. In this paper, we have tried sequential-based method, CNN-based method and GAN-based method to do image inpainting, and also compared their result and suitable application area. What's more, we made a GUI to make visual operation possible.

Keywords: Image inpainting, CNN, GAN

I. Introduction

The task of image inpainting is to fill missing areas of an image, which is one of the important steps in many image processing tasks. For example, image inpainting can be used to fill holes left after removing some content from an image. Human vision has the incredible ability to eliminate inconsistencies. Therefore, the filled area after image inpainting must be visually reasonable. In addition to this, attention needs to be paid to the filling of fine structures during the image inpainting process, especially when the rest of the image contains more clear details.

Like most computer vision problems, image inpainting problems occur earlier than the widespread use of deep learning techniques. Broadly speaking, traditional image inpainting methods can be roughly divided into two types: diffusion-based and patch-based. Diffusion-based methods often follow the relevant formulas of differential operators. On the other hand, the patch-based method is to fill the missing areas with patches in the source image collection, thereby maximizing the patch similarity. However, these methods are not effective in dealing with problems related to repairing details, especially when there are many details near the inpainted area.

The application of deep learning in image inpainting has made a great success. These methods fill the missing pixel values by learning a large number of data distributions. They can produce a coherent structure in the defect area, which is impossible for traditional technology. In the following we will introduce some classical algorithm of image inpainting.

Telea [1] proposed an image inpainting technique based on the fast marching method. The algorithm estimates the image smoothness as a weighted average of the known image neighborhoods of pixels, treats the missing areas as a level set, and uses the fast marching method (FMM) to spread the image information. The algorithm is simple and easy to implement, with faster image processing speed and good results.

The inpainting algorithm based on partial differential equations is suitable for small-scale defects to be inpainted images, such as the BSCB [2] model. The model is essentially thermal diffusion, and the main inpaint principle is to let the information in the known area gradually diffuse along the direction of the isotherm to the unknown field as much as possible, and to maintain the structural consistency in the image inpaint results through anisotropic diffusion.

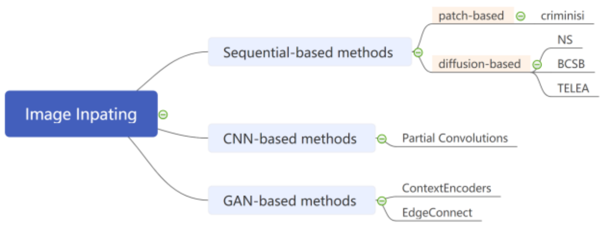


Fig. 1. Typical algorithms in image inpainting

For the inpainting of large damaged areas, a patch-based image inpainting algorithm is usually used, such as the Criminisi [3] algorithm. The algorithm fills the area to be inpainted block by block by searching the best matching block of the target block in the known area, searches the target block globally and judges the priority according to the known information amount and structure information of the target block. This algorithm has a good inpaint effect for inpainting large damaged areas, it is very simple and effective, but the calculation time is longer, and the inpaint effect is poor for high-frequency information.

Context encoder [4] is the first algorithm which use CNN and GAN together in image inpainting. It consists of an encoder capturing the context of an image into a compact latent feature representation and a decoder which uses that representation to produce the missing image content.

The Partial Convolution algorithm [5] uses partial convolutions and automatic mask update steps to achieve image repair. It is the first algorithm to prove the effectiveness of the training image inpainting model on irregularly shaped holes. Among them, the convolution is masked and re-normalized, only on the condition of effective pixels, and the automatic mask is used as part of the forward pass to automatically generate an updated mask for the next layer. The method can achieve state-of-the-art inpainting results.

The edge connection algorithm [6] divides the image inpainting into two stages: edge generation and image completion. Edge generation only focuses on illusory edges in missing areas. The image completion network uses illusory edges and estimates the RGB pixel intensity of the missing area. Both stages follow an adversarial framework to ensure that the illusion edge and RGB pixel intensity are visually consistent. Both of these network projects incorporate loss functions based on deep features to achieve perceptible and realistic results.

II. Related Works

Image inpainting is the process of completing or recovering the missing region in the image or removing some object added to it. So, the operation of inpainting depends on the type [7]. As there are many works about image inpainting, we focus on some of them in classes. And now the methods of image inpainting can be classified into three kinds, they are: sequential-based, CNN-based, and GAN-based.

Sequential-based methods can also be divided into diffusion-based and patch-based. About diffusion-based, Telea [1] proposed an algorithm that diffuse by weighted average of known neighborhoods with FMM. Bertalmio et al. [2] proposed an algorithm based on partial differential equations which is essentially thermal diffusion. Jin et al. [8] proposed a novel algorithm for sparsity-based image inpainting detection and revealed the potential connection between sparsity-based inpaint and canonical correlation analysis (CCA). It's proved to be useful for text and object distortion. As for patch-based, its principle is to fill in the missing region patch-by- patch by searching for well-matching replacement patches in the undamaged part of the image and copying them to corresponding locations. Criminisi et al. [5] proposed an algorithm by searching the target block globally and judging the priority according to the known information amount and the structure information of the target block. Jin and Ye [9] proposed a patch-based image inpainting method using a low-rank Hankel structured matrix completion approach, and performs well for random kind of distortion.

CNN-based methods are trying to be used to solve the problem of capturing the global structure remains in sequential-based method. Alilou and Yaghmaee [10] proposed a non-texture image inpainting method using GRNN neural network, restoring images with scratch, text, noise very well. Plus, Guilin et al. [5] proposed the use of partial convolutions, where the convolution is masked and renormalized to be conditioned on only valid pixels, which focus on mask distortion.

GAN-based methods use a coarse-to-fine network and contextual attention module gives good performance and is proven to be helpful for inpainting. Pathak et al. [4] proposed Context Encoder, it is the first time CNN and GAN are combined in image inpainting. Context encoder trained to generate the contents of an arbitrary image region conditioned on its surroundings, which is suitable for random distortion. Because of image restoring algorithms existed now usually have vague and blurry results with huge amount of time to train the models, Wang et al. [12] based on the construction of Context Encoders, continue to use the strategy of combining the encoders and generative adversarial networks (GANs). Also, Kamyar et al. [6] proposed a two-stage adversarial model Edge Connect that comprises of an edge generator followed by an image completion network, doing a better job of reproducing filled regions exhibiting fine details.

III. Theory And Implementation

A. *diffusion-based method: BSCB*

For the images to be inpainted with small-scale defects (such as stains or scratches), the inpainting algorithm based on partial differential equations works well [13]. There are some typical models, such as BSCB, TV and other models. The following is main theory of BSCB model.

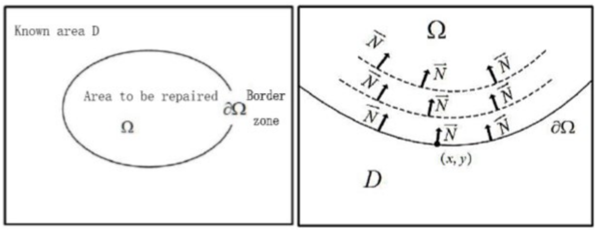


Fig. 2. BSCB model symbol map

The essence of the BSCB model is thermal diffusion. The main principle is to allow the information in the known area to diffuse to the unknown area along the direction of the isosceles as much as possible, and iterate until it is stable.

Let's denote $I^0(i,j)$ the image to be repaired. The iteration formula of the BSCB model is defined as:

$$I^{n+1}(i,j) = I^n(i,j) + \Delta t I_t^n(i,j) \qquad \forall (i,j) \in \Omega$$

where, the superscript n represents the current iteration number n times, Δt represents the iteration step size, $I_t^n(i,j)$ represents the improvement of $I^n(i,j)$, when the value of $I_t^n(i,j)$ is zero, it means that the iteration has converged.

The BSCB model introduces anisotropic diffusion to avoid the intersection of isometric lines. The diffusion equation is:

$$\frac{\partial I}{\partial t} = g_\epsilon(x,y)k(x,y,t)|\nabla I(x,y,t)| \qquad \forall (x,y) \in \Omega^\epsilon$$

where Ω^t is the neighborhood of radius ϵ around Ω, $k(x,y,t)$ is the indicated curvature of the image pixel contour, and $g_\epsilon(x,y)$ is the smoothing function. To meet the conditions:

$$g_\epsilon \begin{cases} 1, & (x,y) \in \Omega \\ 0, & (x,y) \in \Omega^\epsilon - \Omega \end{cases}$$

This model is mainly used for repairing small-scale damage, especially scratches, but it is not suitable for repairing large-scale damaged areas.

B. *patch-based method: Criminisi*

Algorithms based on partial differential equations are usually used to repair small-scale damaged areas, and large fuzzy areas will appear for large damaged areas. For large damaged areas, patch-based image inpainting algorithms are often used, such as Criminisi algorithm.

The symbolic diagram of the Criminisi algorithm is shown in Fig.3 below. In Fig.3, Φ represents the known pixel information, Ω represents the area to be inpainted, $\delta\Omega$ represents the boundary of the area to be repaired, point p is the boundary point, and Ψ is the target block in the neighborhood of the center point.

The Criminisi algorithm searches the best matching block of the target block to be repaired in the known area to fill the area to be repaired block by block. When selecting the point to be repaired, the priority of the point to be repaired should be judged, and the known information and structural information of the point to be repaired should be considered comprehensively.

$$P(p) = C(p) \times D(p)$$

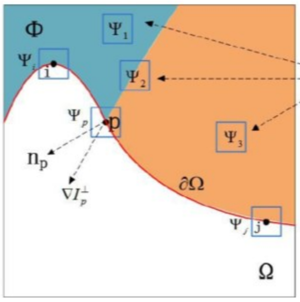


Fig. 3. Criminisi repair algorithm symbol schematic

where $C(p)$ is a confidence term, which measures the proportion of known information in the target block. $D(p)$ is a data item, which measures the structural information in the target block. After that, the best matching block is searched globally and then assigned to the current target block, and the confidence of the unknown pixels is updated, and the area to be repaired is continuously iterated until the area to be repaired cannot be detected. The algorithm is simple and effective, and it is effective when dealing with repair problems in large damaged areas. But the algorithm runs slower, especially for areas with large damage. In addition, for slightly complicated structures, such as curved structures, the repair effect is poor.

C. CNN-based method: Partial Convolutions

Partial convolution algorithm uses the superimposed partial convolution operation and the mask update step to complete the image restoration task. Next, the convolution and mask update mechanisms are introduced separately, and then the model structure and loss function are discussed.

Partial convolution layer Partial convolution layer is divided into partial convolution operation and mask update function. Set $\mathbf{W}$ as the convolution filter weight of the convolution filter, b as the corresponding deviation, x as the eigenvalue (pixel value) of the current convolution (sliding) window, and $\mathbf{M}$ as the corresponding binary mask. The partial convolution at each position is defined as:

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

where $\odot$ is the multiplication between elements. The scale factor $\mathrm{sum}(\mathbf{1})/\mathrm{sum}(\mathbf{M})$ adjusts the number of valid inputs by appropriate scaling. After partial convolution operation, if a position takes at least one valid input as an output condition, the position is valid and it is part of the forward path. The expression is as follows:

$$m' = \begin{cases} 1, & \text{if } \mathrm{sum}(\mathbf{M}) > 0 \\ 0, & \text{otherwise} \end{cases}$$

Implementation Part of the convolutional layer can be obtained by improving the custom layer in time and space, or it can be obtained by expanding the existing pytorch. By defining a binary mask with a size of $C * H * W$, the size is the same as the size of the related image or feature, which is the simplest and most direct implementation method. After that, a fixed convolutional layer is used to update the mask. The kernel size is the same as the size of the partial convolution operation, but the weights are all set to 1, without deviation.

Network design Use a structure similar to UNet, replace all convolutional layers with partial convolutional layers, and then use nearest neighbor upsampling in the decoding stage. Connect two feature maps and two masks as the next partial convolutional layer feature and mask input. Finally, the connection data containing the original input image and the missing regions and masks are input into the final partial convolutional layer.

Partial convolution filling Use partial convolution combined with an appropriate mask to repair the image boundary to avoid the image at the boundary from being affected by extraneous pixels outside the image.

Loss function *Define perceptual loss as follows:*

$$L_{perceptual} = \sum_{p=0}^{P-1} \frac{\left\| \psi_p^{\mathbf{I}_{out}} - \psi_p^{\mathbf{I}_{gt}} \right\|_1}{N_{\psi_p^{\mathbf{I}_{gt}}}} + \sum_{p=0}^{P-1} \frac{\left\| \psi_p^{\mathbf{I}_{comp}} - \psi_p^{\mathbf{I}_{gt}} \right\|_1}{N_{\psi_p^{\mathbf{I}_{gt}}}}$$

$\mathbf{I}_{comp}$ is the original image $\mathbf{I}_{out}$, but the non-aperture pixels are defined as real images. $N_{\psi_p^{\mathbf{I}_{gt}}}$ is the number of ingredients in $\psi_p^{\mathbf{I}_{gt}}$. The perceptual loss function calculates the distance between $\mathbf{I}_{out}$ and $\mathbf{I}_{comp}$ and the real image. For the activation graph of the p-th selection layer given the original input $\mathbf{I}^*$, three pooling layers are constructed to compensate for the loss.

In addition, the valid loss term, style loss term, hole loss term and total-variation loss term are constructed, and the final loss is a combination of these loss terms.

$$L_{total} = L_{valid} + 6L_{hole} + 0.05L_{perceptual} + 120\left(L_{style_{out}} + L_{style_{comp}}\right) + 0.1L_{tv}$$

D. GAN-based methods: Context Encoder

Context Encoders: a convolutional neural network trained to generate the contents of an arbitrary image region conditioned on its surroundings, it can not only earn a representation that captures not just appearance but also the semantics of visual structures. The overall architecture is a simple encoder-decoder pipeline, as Fig.4. network structure of Context Encoders shows. The encoder takes an input image with missing regions and produces a latent feature representation of that image. The decoder takes this feature representation and produces the missing image content. And there is also a channel wise fully-connected layer, as it is important to connect the encoder and the decoder through it, which allows each unit in the decoder to reason about the entire image content.

The trained loss function is composed of reconstruction (L2) loss, which is responsible for capturing the overall structure of the missing region and coherence with regards to its context, but tends to average together the multiple modes in predictions, and adversarial loss, which tries to make prediction look real, and has the effect of picking a particular mode from the distribution.

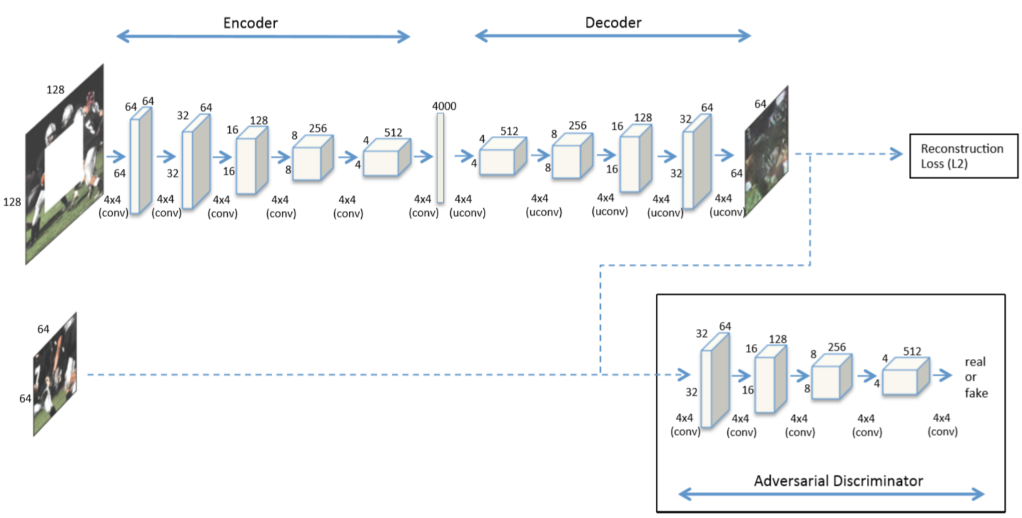


Fig.4. network structure of Context Encoders

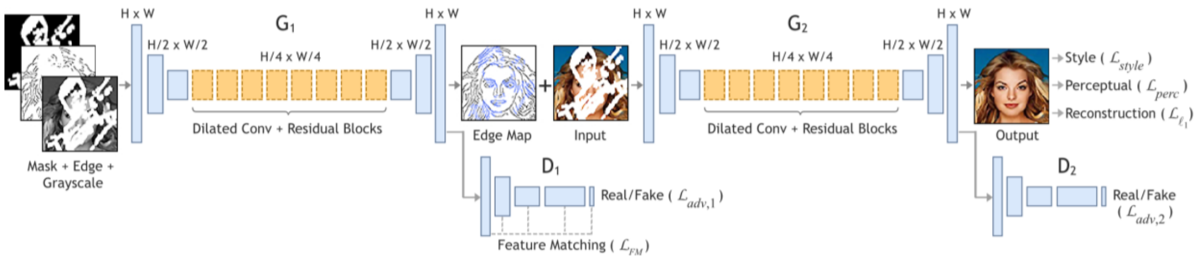


Fig.5. network structure of Edge Connect

The reconstruction loss is shown below, where x is original colorful image, context encoder is F, $\widehat{M}$ is a binary mask corresponding to the dropped image region with a value of 1 wherever a pixel was dropped and 0 for input pixels.

$$L_{rec}(x) = \left\| \widehat{M} \cdot \left(x - F\left(\left(1 - \widehat{M}\right) \cdot x \right) \right) \right\|_2^2$$

However, while reconstruction loss encourages the decoder to produce a rough outline of the predicted object, it often fails to capture any high frequency detail. Therefore, adversarial loss is added as below, Where D is adversarial discriminator.

$$L_{adv} = \max_D \left\{ E_{x \in X}\left[log\left(D(x)\right)\right] + E_{z \in Z}\left[log\left(1 - D\left(F\left(\left(1 - \widehat{M}\right) \cdot x\right)\right)\right)\right]\right\}$$

In conclusion, the whole loss function is:

$$L = \lambda_{rec} L_{rec} + \lambda_{adv} L_{adv}$$

E. GAN-based methods: Edge Connect

As deep learning techniques have yielded significant improvements in image inpainting, many of these techniques fail to reconstruct reasonable structures as they are commonly over-smoothed and/or blurry. Therefore, a two-stage adversarial model Edge Connect has been proposed. It is made up with two parts, part one is edge generator, part two is image completion network. And there function respectively is generating the edge of whole figure with incomplete grayscale image and edge map, mask and completing inpainting work with predicted edge map and incomplete color image, like Fig.5, and in the later, we call edge generator as G_1 and image completion network as G_2.

Edge generator G_1 First of all, I have to define the variables of figure. I_{gt} means original colorful figure, I_{gray} represents grayscale of original figure and C_{gt} means edge map of original colorful figure. G_1 is aiming at predicting the edge map of whole figure with three inputs. Input 1 is masked of gray scale $\tilde{I}_{gray} = I_{gray} \cdot (1 - M)$. Where $M = \{0,1\}$, when pixel belongs to inpating area, M will be 1, otherwise M will be 0. Input 2 is edge map of gray scale of original figure $\tilde{C}_{gt} = C_{gt} \cdot (1 - M)$. Input 3 is gray scale of original figure I_{gray}. The generator predicts the edge map for the masked region

$$C_{pred} = G_1\left(\tilde{I}_{gray}, \tilde{C}_{gt}, M\right)$$

According to the function of G_1, we trained with an objective comprised of an adversarial loss and feature-matching loss

$$\min_{G_1} \max_{D_1} L_{G_1} = \min_{G_1}\left(\lambda_{adv,1} \max_{D_1}(L_{adv}, 1) + \lambda_{FM} L_{FM}\right)$$

where $\lambda_{adv,1}$ and λ_{FM} are regularization parameters. The adversarial loss is defined as

$$L_{adv,1} = E\left(C_{gt}, I_{gray}\right)\left[\log D_1\left(C_{gt}, I_{gray}\right)\right] + E_{I_{gray}} \log\left[1 - D_1\left(C_{pred}, I_{gray}\right)\right]$$

Plus, the feature matching loss L_{FM} is defined as below, the feature-matching loss LFM compares the activation maps in the intermediate layers of the discriminator. L is the final convolution layer of the discriminator, N_i is the number of elements in the i^{th} activation layer, and D_1^i is the activation in the i^{th} layer of the discriminator.

$$L_{FM} = E\left[\sum_{i=1}^{L} \frac{1}{N_i} \left\| D_1^i\left(C_{gt}\right) - D_1^i\left(C_{pred}\right)\right\|_1\right]$$

Image Completion Network G_2 G_2 is to complete inpainting work with predicted edge map and incomplete color image. Therefore, its inputs are original colorful figure I_{gt} and combination of original edge map and G_1 predicting edge map named $C_{comp} = C_{gt} \cdot (1 - \mathrm{M}) + \mathrm{C_{pred}} \cdot M$. The network returns a colorful image I_{pred}

$$I_{pred} = G_2(I_{gt}, C_{comp})$$

This is trained over a joint loss that consists of an l_1 loss, adversarial loss, perceptual loss, and style loss. The adversarial loss is as below.

$$L_{adv,2} = E(I_{gt}, C_{comp})\left[logD_2(I_{gt}, C_{comp})\right] + E_{C_{comp}} \log\left[1 - D_2(I_{pred}, C_{comp})\right]$$

As the name suggests, L_{perc} penalizes results that are not perceptually similar to labels by defining a distance measure between activation maps of a pre-trained network. Perceptual loss is defined as below, where ϕ_i is the activation map of the i^{th} layer of a pretrained network.

$$L_{prec} = E\left[\sum_i \frac{1}{N_i}\left\|\phi_i(I_{gt}) - \phi_i(I_{pred})\right\|_1\right]$$

These activation maps are also used to compute style loss which measures the differences between covariance of the activation maps, style loss is computed by

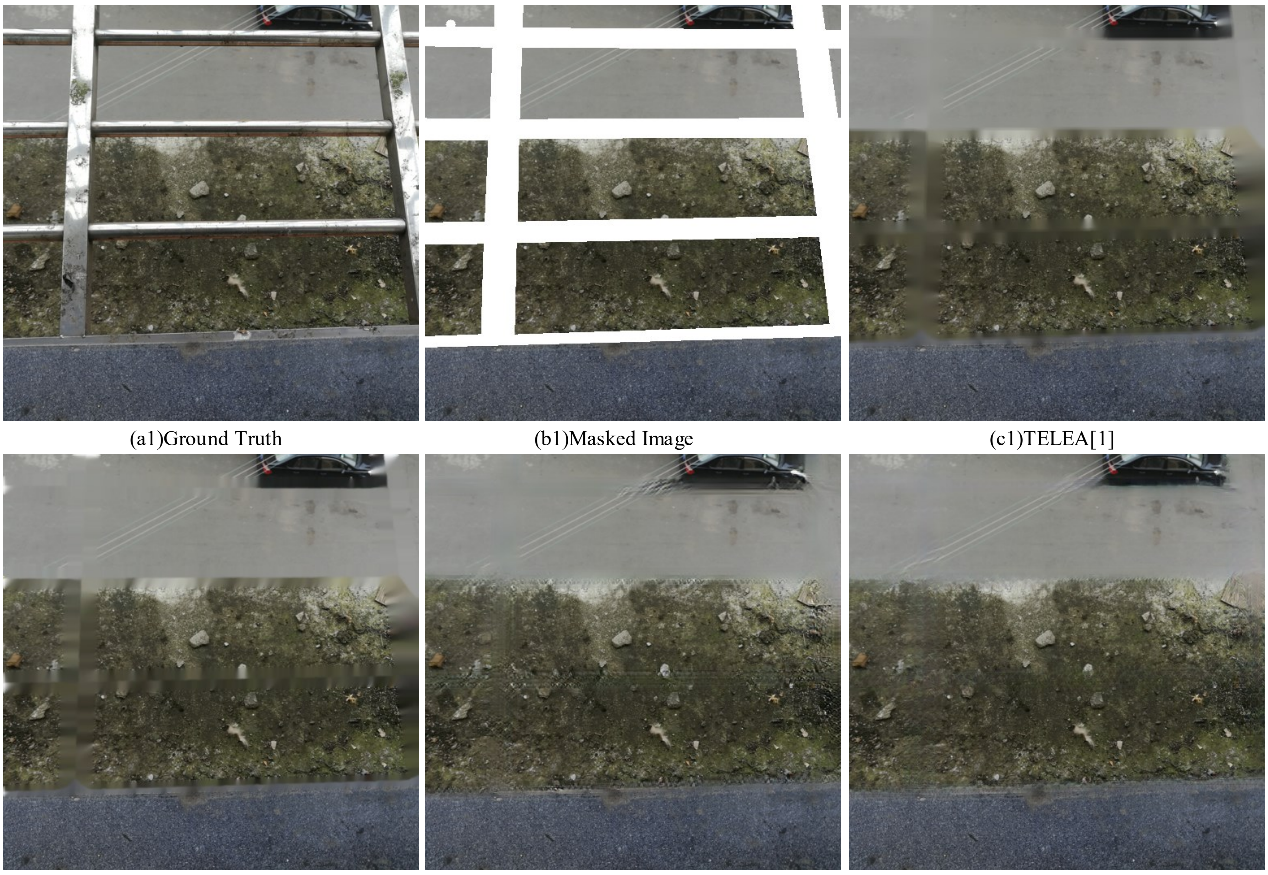
$$L_{style} = E_j\left[\left\|G_j^{\phi}(\tilde{I}_{pred}) - G_j^{\phi}(\tilde{I}_{gt})\right\|_1\right]$$

Finally, the overall loss function of G_2 is

$$L_{G_2} = \lambda_{l_1} L_{l1} + \lambda_{adv,2} L_{adv,2} + \lambda_p L_{perc} + \lambda_s L_{style}$$

IV. EXPERIMENT RESULT

Image inpainting can generally be used to remove obstructions in the image. As shown in Fig.6, we select 5 pictures, including lawns, buildings and other scenes, remove some of the objects, such as railings and people, and use four algorithms to repair the images. According to the repair results, the NS and TELEA algorithms have good repair effects on small damaged areas, but when repairing large damaged areas, they produce blurred images and phantom effects, such as the repair results of lake water and lawn. In addition, the two algorithms are obviously less effective in repairing high-frequency image areas and cannot reflect image details. The repaired images do not meet the perceptual semantics, such as the edges of house buildings. The effect of the P-Conv algorithm is significantly better than the previous two algorithms, especially the roadside pictures and lake water repair results, it is difficult for the human eye to identify the repair traces. But the algorithm has overfitting when the damaged image does have too much area. For example, after repairing the sky, there are red areas that should not exist. In addition, the algorithm still lacks the treatment of details. For example, the shadow of the lawn should be a straight line, but the curve after repair is not in line with the actual situation. The Edge Connect algorithm has the best effect, which can repair large irregular broken areas, and ensure that the repaired picture details are in line with the actual situation, it is difficult to distinguish the repair traces. But it is difficult to achieve perfect restoration in the low-frequency area, such as the restoration of the lake water, where exists a little blurring.

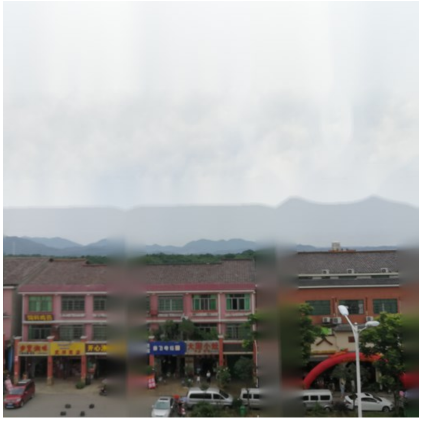


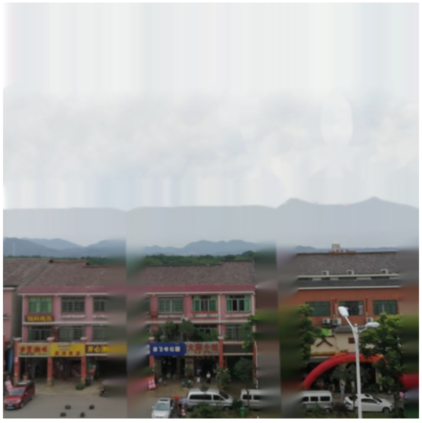
(a1)Ground Truth (b1)Masked Image (c1)TELEA[1]

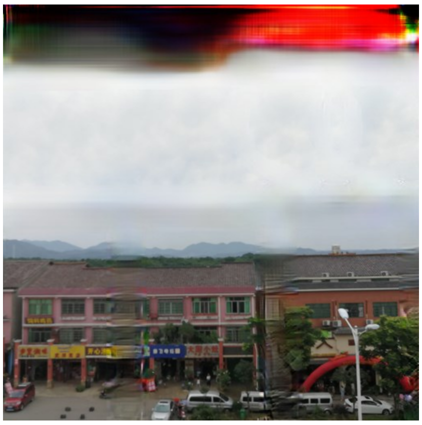
(d1)NS[11] (e1)P-Conv[5] (f1)Edge Connect[6]


(a2)Ground Truth


(b2)Masked Image


(c2)TELEA[1]


(d2)NS[11]


(e2)P-Conv[5]


(f2)Edge Connect[6]


(a3)Ground Truth


(b3)Masked Image


(c3)TELEA[1]


(d3)NS[11]

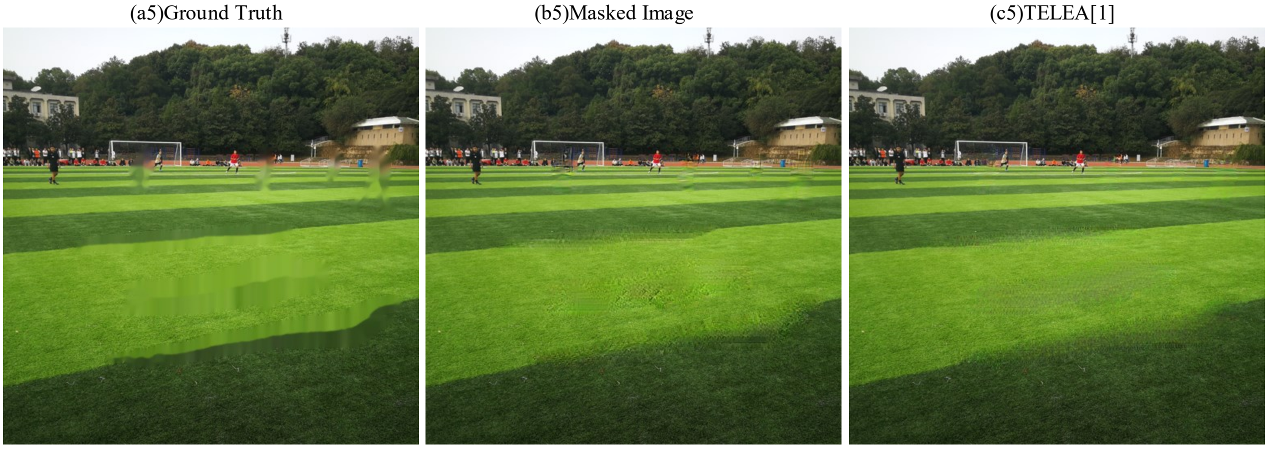

(e3)P-Conv[5]


(f3)Edge Connect[6]

(a4)Ground Truth (b4)Masked Image (c4)TELEA[1]

(d4)NS[11] (e4)P-Conv[5] (f4)Edge Connect[6]

(a5)Ground Truth (b5)Masked Image (c5)TELEA[1]



(d5)NS[11] (e5)P-Conv[5] (f5)Edge Connect[6]

Fig. 6. (a)original image, (b)masked image, result of (c)TELEA, (d)NS, (e)P-Conv, (f)Edge Connect algorithm

(a)Ground Truth (b)Masked Image (c)Criminisi[3]

Fig. 7. (a)original image, (b)masked image, result of (c)TELEA, (d)NS, (e)P-Conv, (f)Edge Connect algorithm

Because the patch-based algorithm Criminisi runs too slowly, we only test this algorithm in a lightly masked (2.89%) image. Although there are only 7583 pixels to be inpainted, it still runs almost 44 minutes. Fortunately, as shown in Fig.7, it's result is the best one in traditional algorithms. What's more, the operation time is roughly proportional to pixels to be painted and the size of image. Hence to run other examples will cost even several days.

Tab. 1. Quantitative result with TELEA [1], NS [11], PConv [5], Edge Connect [6]

	Mask	TELEA	NS	PConv	Edge
SSIM	0-10%	**0.9881**	0.9880	0.9854	0.9863
	10-20%	0.9651	**0.9665**	0.9577	0.9641
	20-30%	**0.7711**	0.7671	0.7551	0.7599
	30-40%	0.7608	**0.7633**	0.7617	0.7602
	40-50%	**0.7480**	0.7466	0.6583	0.7392
PSNR(dB)	0-10%	**38.930**	38.602	38.498	36.934
	10-20%	**35.395**	35.303	34.831	34.783
	20-30%	**15.995**	15.897	15.902	15.914
	30-40%	**15.185**	15.229	15.455	15.375
	40-50%	10.923	**11.077**	10.419	11.025
Time(s)	0-10%	0.148	0.092	14.32	18.45
	10-20%	0.359	0.310	17.21	20.16
	20-30%	0.410	0.412	16.86	21.68
	30-40%	0.481	0.478	14.67	19.67
	40-50%	0.690	0.717	15.67	18.57

As shown in Tab. 1, use SSIM and PSNR objective evaluations to evaluate the image inpainting results of the four algorithms (TELEA, NS, PConv, Edge) under different Mask ratios, and record the time spent on inpainting. When using the SSIM indicator to evaluate the repair result graph, the two traditional algorithms of TELEA and NS under different Mask ratios are higher than the two CNN-based algorithms of PConv and Edge. In addition, the evaluation results of the PSNR measurement also show the repair results of the traditional algorithm has a higher PSNR measurement value. According to the actual image observation, the repair result of the traditional algorithm is obviously worse than that of the CNN-based image repair algorithms, but because the two types of CNN-based algorithms produce details that are far from the original image during the repair process, for example the area that is human in the original image replaced with a lawn image containing high-frequency information, resulting in a low objective evaluation value. Therefore, the use of objective indicators such as PSNR and SSIM to evaluate the repair results has a large error, and the evaluation effect is inaccurate. In the experiment of recording the processing time of each algorithm, it is obvious for the traditional algorithm that the processing time will increase correspondingly as the Mask ratio increases, but the Mask ratio has no significant effect on the processing time of the two types of algorithms based on CNN. In addition, due to the simple structure and fewer considerations, the traditional algorithm has a shorter processing time within 1 second, while the PConv algorithm has a processing time between 14 and 18 seconds, and the Edge algorithm is slightly longer between 18 and 22 seconds.

V. CONCLUSION

With the results above, we can find that the image inpainting effect of Partial-Convolution is better than NS, BCSB and Criminisi, which is reflected in that it can better repair the high-frequency part of the image, that is, the detail part, and the restored picture is more realistic. But it also has a problem that it may overfit in the low frequency part. So, The Edge Connect solves this problem. The effect of its repairing pictures is the best. It can not only repair large areas of irregular broken areas, but also ensure that the details of the repaired pictures conform to the actual situation, while there will be a little blur in the low frequency part.

In general, the GAN-based method works best, and it also takes the longest time. The second best is CNN-based method, the time is medium. The Sequential-based method is less effective, but the fastest.

REFERENCES

[1] Telea A . An image inpainting technique based on the Fast Marching Method[J]. Journal of Graphics Tools, 2004, 9(1):23-34

[2] Bertalmio M, Sapiro G, Caselles V, et al. Image inpainting[C]// SIGGRAPH '00 Proceedings of the 27th annual conference on Computer graphics and interactive techniques, July 23-28, 2000, New Orleans, LA, USA., 417-424(2000).

[3] Criminisi A , Perez P , Toyama K . Region Filling and Object Removal by Exemplar-Based Image Inpainting[J]. IEEE Transactions on Image Processing, 2004, 13(9):1200-1212.
[4] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. A. Efros, Context encoders: Feature learning by inpainting, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2536-2544.
[5] 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), September 2018.
[6] Nazeri K , Ng E , Joseph T , et al. EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning[J]. 2019.
[7] Elharrouss, O., Almaadeed, N., Al-Maadeed, S. et al. Image Inpainting: A Review . Neural Process Lett 51, 2007–2028 (2020). https://doi.org/10.1007/s11063-019-10163-0.
[8] X. Jin, Y. Su, L. Zou, Y. Wang, P. Jing, Z. J. Wang, Sparsity-based image inpainting detection viacanonical correlation analysis with low-rank constraints, IEEE Access 6 (2018) 49967-49978.
[9] K. H. Jin, J. C. Ye, Annihilating filter-based low-rank hankel matrix approach for image inpainting, IEEE Transactions on Image Processing 24 (11) (2015) 3498-3511.
[10] V. K. Alilou, F. Yaghmaee, Application of grnn neural network in non-texture image inpainting and restoration, Pattern Recognition Letters 62 (2015) 24-31.
[11] M. Bertalmío, Bertozzi A L , Sapiro G . *Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting*[C]// Null. IEEE Computer Society, 2001..
[12] H. Wang, L. Jiao, H. Wu, R. Bie, New inpainting algorithm based on simplified context encoders and multi-scale adversarial network, Procedia computer science 147 (2019) 254-263.
[13] 李俊柏. 数字图像修复技术的研究[D]. 浙江理工大学.