Image inpainting algorithm based on convolutional neural network structure and improved Deep Image Prior

Junshu Wang[1,2], Yuxing Han[1,2*]

(1. *College of Electronic Engineering, South China Agricultural University, Guangzhou 510642, China*;
2. *Lingnan Guangdong Laboratory of Modern Agriculture, Guangzhou 510642, China*)

Abstract: In previous studies, researchers believed that the reason for the excellent performance of convolutional neural networks was that they could learn hidden information from special-purpose datasets, and the emphasis was on learning. Recently, the authors of Deep Image Prior proved that the generator structure itself (using convolutional neural network) could extract image prior information and be used for the image inpainting task. In this paper, based on Deep Image Prior, four improvements (mix input, network noise, weight decay, and burning mean output) are proposed for preventing overfitting and improving output stability. Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) in two stepwise comparative experiments showed that our image inpainting algorithm surpassed the original algorithm and state-of-the-art algorithms after adding the proposed improvements in sequence. In large hole inpainting, the PSNR of our algorithm was 3.23 dB higher than in the original Deep Image Prior. Then, in a binary Bernoulli inpainting experiment, our algorithm achieved better performance in most classical image inpainting, proving that the algorithm could use the same set of parameters for each image in the task. In addition, this experiment also illustrated the performance of burning mean output in stabilizing the output and reducing the influence of meaningless noise in the early stage of the iteration on subsequent image inpainting.

Keywords: convolutional neural network, Deep Image Prior, image prior information, image inpainting, overfitting, large hole inpainting, binary Bernoulli inpainting
DOI: 10.33440/j.ijpaa.20200304.135

Citation: Wang J S, Han Y H. Image inpainting algorithm based on convolutional neural network structure and improved Deep Image Prior. Int J Precis Agric Aviat, 2020; 3(4): 65–73.

1 Introduction

In the last decade, benefiting from the development of the semiconductor industry, the computation power of graphic processing units (GPUs) has grown[1], which has promoted the development of high-performance computing. As a result, deep learning-based image processing algorithms, which depend on computing resources, have been a focus of research[2]. Another key point that makes deep learning-based algorithms perform well is the invention and application of convolutional neural networks (CNNs)[3]. With the success of AlexNet in the 2012 image network competition, convolutional neural networks have become more popular and have been used image processing tasks, such as image classification[4], object detection[5], crowd counting[6], semantics segmentation[7], etc.

Image collection mainly relies on various sensors. Nowadays, many external factors affect the quality of image transmission, the bandwidth of the transmission network, the status of the network, the stability of the network, etc., causing distortion of important images or key frames in the video, so image reconstruction technology is worth researching.

Furthermore, convolutional neural network has also demonstrated its effectiveness in image reconstruction tasks[8]. Especially in the design of the network structure, many researchers use the autoencoder structure in image reconstruction tasks, in which mappings of image space to feature space and feature space to image space are used in image reconstruction. Among them, convolutional neural networks are often used in the design of encoders and have been proven to achieve reasonable results. In previous studies, many scholars noted that the success of CNN is mainly due to its nonlinear activation function and multiple hidden layer structure, which contains millions of parameters; the structure can learn the hidden relation between input and output by training with an appropriate dataset, and its core process is iterative learning in backward propagation[9]. However, some researchers have questioned this in their latest research; that is, they question whether learning is the only reason why convolutional neural networks perform well. For instance, the authors of[13] recently showed that the same image classification network that generalizes well when trained on genuine data can also overfit when presented with random labels.

In a latest research, the authors of Deep Image Prior show that the convolutional neural network structure itself can obtain low-level semantic information from a degraded image through network parameter iterative learning, and the learned semantic information is enough to complete tasks such as image inpainting, denoising, and super-resolution, etc.[10]. It is noteworthy that the input supporting these deep learning-based image processing applications can only be a degraded image; furthermore, no additional dataset is needed to support learning. However, in Deep Image Prior, in order to achieve the same image reconstruction effect as the state-of-the-art algorithms in comparative experiments, it is necessary to adjust the network structure and parameters for each specific image reconstruction

Received date: 2020-10-26 **Accepted date:** 2020-12-10
Biography: Junshu Wang, PhD candidate, research interests: deep learning, precision agriculture, Email: junshuwang@stu.scau.edu.cn;
*** Corresponding author: Yuxing Han**, PhD, Professor, research interests: precision agriculture. Mailing Address: South China Agricultural University, Wushan Road, Tianhe District, Guangzhou City, Guangdong, 510642, China. Email: yuxinghan@scau.edu.cn.

task, even each image. Therefore, this method is cumbersome and cannot be applied to streamlining image inpainting tasks.

As the main contribution of this paper, four main improvements are proposed: mix input, network noise, weight decay, and burning mean output. The reason for adding network noise and weight decay is to prevent the influence of overfitting as much as possible (because overfitting will cause the hourglass architecture to generate an image exactly the same as the degraded image x_0). The reason for adding the mix input strategy is to prevent the appearance of large areas of dead neurons that cannot be updated iteratively. The benefit of burning mean output is to stabilize the output image and prevent meaningless noise in the previous iteration from affecting the subsequent image repair performance. In two stepwise comparative experiments, large hole inpainting and binary Bernoulli inpainting, were performed. In comparative experiments with the state-of-the-art algorithms, after adding the improvement strategy step-by-step, our algorithm achieved a gradual optimization effect in the evaluation standard PSNR and SSIM. In the large hole inpainting of image *library,* compared with the original Deep Image Prior algorithm, the algorithm in this paper improved the PSNR by 3.23 dB after adding all of the improvement strategies. In the binary Bernoulli inpainting experiment, we used the same set of parameters in the inpainting task for each classic image. Finally, our algorithm achieved higher PSNR value and better visual performance than the original Deep Image Prior and other state-of-the-art algorithms in this stepwise comparative experiment.

2 Related works

2.1 Image inpainting and image prior information

The prior information of the degraded image is a key factor to decide the performance of image inpainting[11], because even if the image has been degraded, it still contains features that need to be referenced in image inpainting. In the field of natural image processing tasks, image inpainting is a typical reverse problem, therefore the solution of the problem is not fixed. From the perspective of statistics, the solution process of the image inpainting task conforms to Bayes' theorem, in which the prior can be represented as a prior probability. In order to reduce the solution space of the problem while better approximating the real solution, some constraints are needed. In the inpainting task, the prior can be considered as a constraint factor making the restored image obtain the basic semantic features of the original image as much as possible. From this point of view, the image inpainting task can be represented by the following formula:

$$x = argmin_x f(x, x_0) + p(x) \quad (1)$$

where, x is the original image; x_0 is the degraded image; $p(x)$ is the prior item of the degraded (noisy/low-resolution/occluded) image, and $f(x, x_0)$ is mapping between x and x_0. In this paper, image inpainting can also be presented by a standard regularization, which is similar to Formula (1):

$$x = argmin_x E(x, x_0) + R(x) \quad (2)$$

where, x is the original image; x_0 is the degraded image; $R(x)$ is a regularizer, and the choice of task-dependent loss $E(x, x_0)$ is often directly dictated by the application. In this research, as mentioned before, the structure of the convolutional neural network itself proved to be able to learn prior information from a degraded image, therefore the inpainting operation performs image reconstruction from a fixed random tensor $z \in \mathbb{R}^{C' \times H' \times W'}$. In this paper, the relationship between the neural network and the degraded image is expressed in a parameterized way: $x = f_\theta(z)$. Here, $x \in \mathbb{R}^{3 \times H \times W}$, and the network maps the parameter θ, comprising the weights and bias of filters in the network structure, to x. Therefore, in this paper, the image inpainting task can be expressed by parameterization method, in the following formula:

$$\theta^* = argminE(f_\theta(z), x_0) + R(f_\theta(z)) \quad (3)$$

where, θ^* is the parameter of the convolution neural network when the image inpainting achieves the best performance in practice, and $R(f_\theta(z))$ is the prior information hidden in the network structure.

2.2 Convolutional Neural Network

When studying the visual system of cats, biologists Hubel and Wiesel found that the transmission of visual information from the retina to the brain was accomplished through the activation of multiple levels of receptive fields[12]. Based on this, they proposed the concept of receptive field. Inspired by this concept, convolutional neural network was proposed. The optimization of convolutional neural network is mainly due to the concept of weight sharing and the convolutional–downsampling layer combination. When convolutional neural networks are used in tasks such as super-resolution and image denoising, generally they are used to construct an end-to-end mapping (between the degraded and restored image). In this construction, the feature extraction of degraded images is completed by convolutional neural networks in the process of downsampling. Therefore, this is also one of the main purposes of using convolutional neural networks in the image inpainting task.

By using the CNN structure, one can establish a mapping between a large area in low-dimensional space and a certain value or small area in high-dimensional space; furthermore, by adjusting the depth of the convolutional layer and the size of the filter, the image space is mapped to a three-dimensional (width, height, depth) feature space, and this feature space is composed of feature maps formed by operation of the convolutional kernel (also called a filter), the input of which is the output of the previous layer. The operation of the convolution layer is shown in the formula:

$$x_j^l = f(\sum_{i \in M_l} x_i^{l-1} * k_{ij}^l + b_i^l) \quad (4)$$

where, x_j^l is the jth feature map of the lth layer of a deep convolution structure; f represents the activation function; M is the set of input feature maps; * represents the convolution operation; k is the convolution kernel, and b represents the bias term. In the image inpainting deep convolutional neural network, the size of the receptive field and the complexity of the features that can be extracted by a convolutional neural network will change with the deepening of the convolutional layer. In low-dimensional convolutional layers that are close to the input image, the network may only extract some low-level features such as edges, curves, and corners and other low-level semantic information; with deepening of the convolution layer, the network then can learn more complex and high-level semantic features in the deeper layer near the output (generally the feature vectors generated by the encoder). Furthermore, the basic tool for establishing the mapping mentioned above is a neuron with learnable weight and bias. Each neuron can receive the output from the previous layer, and then use the activation function to establish a nonlinear relationship between neurons. The weights and bias in neurons can be learned by the backward propagation process, aimed at minimizing the task-dependent loss function. Deep Image Prior based image inpainting proved that prior information can be learned from the convolutional structures, so other characteristics of convolutional neural networks should be mentioned.

Due to overfitting, the image inpainting model based on the

principle of Deep Image Prior will eventually generate an image exactly the same as the degraded image, thus losing the meaning of inpainting. Therefore, the characteristics of the convolutional neural network structure that can prevent overfitting should also be mentioned. Generally, the more parameters in the model, the more likely there will be overfitting. In terms of reducing the size of the parameter, the sparse connectivity mechanism of the convolutional neural network is one method. In the backward propagation (BP) neural network, the neuron nodes of each layer obtaining a linear one-dimensional structure, and the neurons between layers are fully connected. On the contrary, a convolutional neural network uses the local correlation between layers to make the neurons of each adjacent layer only connect with the upper neuron nodes that are close to it (the local connection mechanism). Generally, in image inpainting tasks, the performance of a pixel after inpainting mainly depends on its neighboring pixels; this also corresponds to the sparse connectivity mechanism of the convolutional neural network, which greatly reduces the parameter scale of the network.

At the same time, weight sharing also reduces the parameter amount. In CNN, each filter of the layer repeatedly acts on each receptive field when the input is being convolved. Then, the convolution result constitutes the updated feature maps, and the expression of features is also updated accordingly. When convolution operation is performed on a feature map, for each receptive field, a filter with the same parameters will be used, including the same weight and bias. The advantage of weight sharing is that the location of local features in the input is not taken into account when extracting features from images; meanwhile, this sharing will greatly reduce the parameter amount. For example, when the size of an input image is 192×192, if the weight sharing mechanism of the 3×3×32 convolution kernel of the first layer is removed, the number of parameters will become 192×192×32, which is about 1.17 million parameters, or 4096 times the original 288 parameters.

Pooling is a nonlinear downsampling method in the CNN structure (shown in Figure 1). In the image inpainting model, which uses the autoencoder structure, after obtaining image feature information through convolution, the model will use these features to perform an upsampling operation to achieve image reconstruction. However, not all features need to be presented separately. The convolution layer can reduce the dimension of convolution features by a pooling operation. In the process of pooling, a feature map will be divided into several $n \times n$ disjoint regions, and after dimensionality reduction, the maximum (or average) value of these regions will be used to represent the feature. Among the advantages of the pooling operation, it can reduce the size of the image, increase the receptive field size of the convolution kernel, and reduce the computational complexity while preserving the features as much as possible. it is noteworthy that maximum pooling has shift invariability; even if the image has a small displacement, the extracted features will remain. In this efficient sampling method, which reduces the data dimension, the pooling operation also enhances the robustness of the model and reduces the useless information, which is helpful for feature extraction. Therefore, in the image inpainting task, the prior information of image texture features can be extracted more effectively. On the other side, average pooling can preserve background information. Both pooling methods can reduce the amount of network parameters to prevent overfitting, and allowed us use the interrupted iteration method to output a repaired image more effectively.

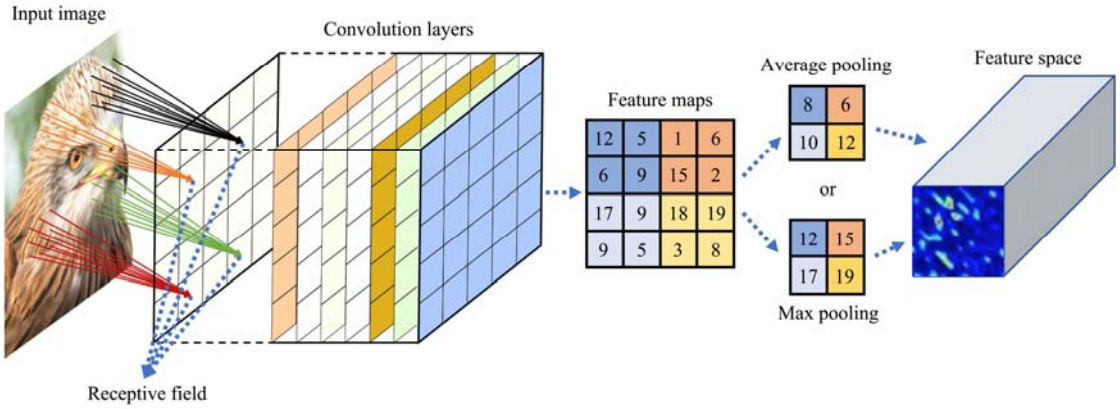


Figure 1 Typical convolutional neural network

In the Deep Image Prior article, it can be seen from the process of model fitting that in the image reconstruction tasks (such as super-resolution, large hole inpainting, image denoising, etc.), the longer the fitting process, the more helpful it is to find a suitable time to interrupt the iteration, which means that the model needs to combat the overfitting problem to the greatest extent. Generally, the model overfitting problem refers to excessively high performance in fitting the training data. The result is that the model cannot be effectively applied to unfamiliar data. In this study, the realization of image reconstruction tasks relies on interrupting the iterative process at an appropriate time. Therefore, once the overfitting phenomenon occurs, the model will generate an image exactly the same as the degraded image x_0; once this happens, the inpainting model will lose its meaning.

The model for the super-resolution reconstruction task was trained in order to show the degradation caused by the overfitting problem in the research of this paper; meanwhile, according to the number of iterations, images of different fitting stages are generated for observation. It can be observed from the figure that image reconstruction starts from random noise $z \in \mathbb{R}^{32 \times H \times W}$. In the initial stage, due to the high Mean Square Error (MSE) value and poor learning of image features by the parametric network, the image cannot be reconstructed reasonably (50 iterations). However, the low-level semantic information in the degraded image, such as color distribution and partial contour, can be obtained, while the background and key contents of the image are distinguished in some areas. As the iteration continues, in the underfitting stage, more semantic information is learned by the parameterized network (the image outline is clearer and the color

distribution is further determined). However, shown in Figure 2, in the image generated after 120 iterations, it can also be found that the inpainting performance is still not ideal, the image features x_0 have not been fully learned, and the effect of super-resolution reconstruction has not been achieved. In the just-right stage (about 2500 iterations), the parameterized network $f_\theta(z)$ achieves the best image super-resolution reconstruction performance, the image contour is sharper and clearer, and the color partition is also obvious enough, which shows that the convolutional neural network structure has learned enough semantic information and can give priority to output an image with higher resolution. Finally, the blur on the contour and some meaningless noise will be learned as high-level semantics in the overfitting stage (over 60,000 iterations). At this stage, through the mapping of the parameterized network, the model can only generate an image that is exactly the same as the input (a degraded image), so thus far, the parameterized network has no normalization ability and image inpainting function.

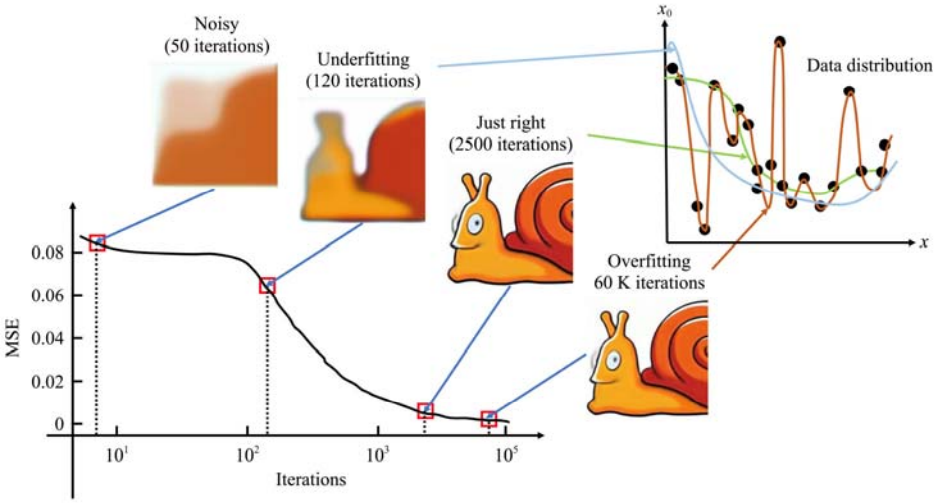


Figure 2 Stepwise generated images in super-resolution reconstruction task

3 Materials and methods

3.1 Overfitting

According to the advantages of convolutional neural network in preventing overfitting and image generation after model overfitting, overfitting must be prevented. In this paper, the improved Deep Image Prior image inpainting model uses regularization to prevent overfitting. Then an appropriate iteration time is selected in the model fitting stage, and the iteration will be interrupted. Finally, a restored image will be generated by using this parametrized model. Since in this paper MSE is used as the loss function, we use the *L2* loss to illustrate the relationship between regularization and loss function. In common tasks, regularization based on the mean square error can be expressed by the following formula:

$$L(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)^2+r(\theta) \tag{5}$$

where, the first half is an approximation item; θ is the parameter; $h_\theta(x_i)$ is the predicted value; y_i is the label value of the sample, and $r(\theta)$ is the regularization item. The additional regularization item can prevent the loss function from being too close to 0, so the parameters need to be limited.

3.2 Regularization and network noise

In this research, noise-based regularization is used. In the Deep Image Prior based image inpainting model, in order to prove that the prior information of the image is obtained by the CNN structure itself, rather than from the image x_0, in each iteration an additive normal noise is added to the input z with zero mean and standard deviation. This shows that during training, the input of each iteration changes randomly, and during the initialization phase of the network, the weight parameters of the network are initialized randomly, therefore it is only the deep convolutional network structure itself that obtains the prior information. In the research, in order to reduce the degradation caused by overfitting as much as possible while proving the network structure's prior information extraction capability, in each iteration, additive normal noise with zero mean and standard deviation was also added to the network weight parameter θ. Therefore, in this model, the loss function calculation process in each iteration can be expressed by the following formula:

$$l_\theta=(f_{\theta+G_{noisy\_\theta}}(z+G_{noisy\_z})-x_0)^2 \tag{6}$$

where, l_θ is the loss function when the network parameter is θ; G_{noisy_θ} is the random noise perturbing the parametrized network (with zero mean and standard deviation), and G_{noisy_z} is the random noise perturbing the input z (with zero mean and standard deviation). The parameter relationship and prior information section of the reconstructed image can be expressed by the following formula:

$$\theta^*=argminE(f_{\theta+G_{noise\_\theta}}(z+G_{noise\_z}),x_0)+R(f_{\theta+G_{noise\_\theta}}(z+G_{noise\_z})) \tag{7}$$

where, θ^* is the parameter of the convolutional neural network when the image inpainting achieves the best performance in practice (trained with random noise G_{noisy_θ} and G_{noisy_z}). The second half of the formula is the image prior information of the trained network structure by adding random noise to the network parameters.

3.3 Mix input

From the experiment with Deep Image Prior, it can be seen that parameterization offers high impedance to noise, therefore image reconstruction using noise as input is the slowest in the fitting process of parameter training. In the image inpainting task, we considered two types of degradation, large hole inpainting and binary Bernoulli inpainting (the image is sampled to drop 50% of pixels at random). In order to ensure the gradient descent speed of parameterization, the mix input type (a combination of image and noise) was used in the experiment. Compared with the

$z \in \mathbb{R}^{32\times H\times W}$ noise-based reconstruction method, the combined image and noise input method in this paper makes the parameterized network $f_\theta(z)$ converge faster. This input method has another advantage for large hole inpainting. In this type of inpainting task, the missing part of the degraded image will form many non-informative and non-gradient areas. If rectified linear unit (ReLU) is used as the activation function, many unlearnable areas where the derivation result is zero will be formed, containing many dead neurons[14], and they will become a burden in the network structure because the weights of such areas cannot be updated even if they undergo multiple iterations. Due to the limitation of the ReLU activation function, it is difficult for the image inpainting model to reconstruct such areas with good performance. Fortunately, we found that after noise is added to the image, the missing areas will become relatively "soft". In the convolution structure, the activation function values in the area will not be zero, therefore the weights of the large holes can be continuously updated during the iteration process.

3.4 Weight decay

When using the naive gradient descent, *L*2 regularization has the same effect as weight decay, because the influence of the regularization item on the weight is to make the weight attenuate a certain value in each iteration. However, when using Adam, the learning rate will gradually decrease, which makes the model converge better. If *L*2 regularization is used during Adam optimization, the effect of the regularization item will vary with the learning rate, because when calculating the gradient, the subtraction item needs to be divided by the sum of the gradient squares, which makes the item too small to realize the original definition of weight decay; the greater the weight, the greater the decay. This is one of the reasons why the performance of Adam optimization is sometimes worse than the performance of stochastic gradient descent (SGD) with momentum. The weight decay updates all weights with the same coefficient, and the penalty value is related to the value of the parameter; the greater the weight, the greater the penalty, as expressed by the weight decay formula:

$$J = J_0 + \frac{\lambda}{2m}\sum_{i=1}^{m}\theta^2 \tag{8}$$

where, λ is the decay coefficient and θ is the weight parameter. The reason for multiplying by 1/2 is to facilitate differential calculation; J is the cost function and J_0 is the cost function before weight decay. After derivation, the following formula can be obtained:

$$\frac{\partial J}{\partial \theta} = \frac{\partial J_0}{\partial \theta} + \frac{\lambda}{m}\theta \tag{9}$$

The following complete formula can be obtained by introducing this result into the weight decay formula:

$$\theta = (1-\lambda)\theta_0 - \alpha\left(\frac{\partial J_0}{\partial \theta_0} + \frac{\lambda}{m}\theta_0\right) \tag{10}$$

where, θ is the weight value after the decay; θ_0 is the original weight value, and α is the learning rate. From the formula, it can be found that after the original gradient decreases, the weight parameter needs to subtract an additional value, $\frac{\lambda}{m}\theta_0$, which is positively related to the weight value. Therefore, the greater the weight, the more decay, which can effectively reduce the cost function. In addition, after the weight is attenuated a little, another benefit is that the entire neural network will not be too sensitive to noise. If the weight is too large, a little change in the corresponding input value will have a large effect that will significantly change the output.

3.5 Leaky ReLU

Rectified linear unit (ReLU) is the most commonly used activation function in neural networks. ReLU retains the biological inspiration of the step function (the neuron is activated only when the input exceeds the threshold), but when the input is positive, the derivative is not zero, which allows gradient-based learning (although at x=0, the derivative is undefined). Using ReLU can improve computational efficiency, because neither the function nor its derivative contains complex mathematical operations. However, when the input is negative, ReLU will directly invalidate the neuron, which is called dead ReLU, because if the input is negative, the gradient will be zero, so the neuron's weight cannot be updated and it will remain silent during the remaining iterations, which is called dead neuron. The performance in the large hole inpainting task is not ideal. In order to solve the shortcoming of the ReLU function, a leakage value is added to the negative half of the function, so it is called leaky ReLU, as shown in Figure 3.

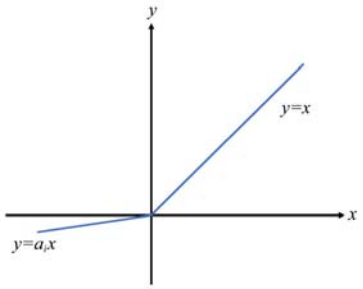


Figure 3 Leaky rectified linear unit (ReLU)

The formula of leaky ReLU[15] is as follows:

$$y = \begin{cases} x, & if\ x \geq 0 \\ a_i x, & if\ x < 0 \end{cases} \tag{11}$$

In the formula, a_i is a fixed parameter within [0,1). The leaky ReLU function is a variant of the ReLU activation function. This function has a small slope for negative inputs. Since the derivative is always non-zero, this can effectively reduce the appearance of silent neurons and make its parameters available for gradient-based learning (although it will be relatively slow). As a result, the problem of neuron silence caused by the ReLU function entering the negative interval is solved.

3.6 Burning mean output

In the training optimization strategy, a new weighted output is proposed, called burning mean output, and in the comparative experiment, we proved that it performs better in most cases. In the original Deep Image Prior algorithm, the author provided two types of output: directly output the result, or combine the previous and current output images with different weights, which we call average mean output. However, in actual image inpainting tasks, especially in the initial stage of training, the output images are all messy or have very low PSNR. The weighted output of such images is meaningless and may even cause a certain degree of interference to the images generated in subsequent training. So, in our algorithm, an adaptive weighted intervention output method is proposed. We found that in the inpainting application, after 5000 iterations, the image will gradually tend to be stable, so we chose to carry out weighted intervention on the output at 5000 iterations.

3.7 Model outline

In terms of the design of the network structure, this study uses the same structure as Deep Image Prior, which uses the

encoder-decoder structure in large hole inpainting and a U-Net-like "hourglass" architecture with skip connections in binary Bernoulli inpainting.

As shown in Figure 4, in the inpainting, starting from a mix input (noise + degraded image x_0), we iteratively update the parameters in order to minimize the data term $E(f_\theta(z), x_0)$. At every iteration the weights θ are mapped to an image $x=f_\theta(z)$. The image x is used to compute the task-dependent loss $E(x, x_0)$. The gradient of the loss related to weights θ is then computed and used to update the parameters.

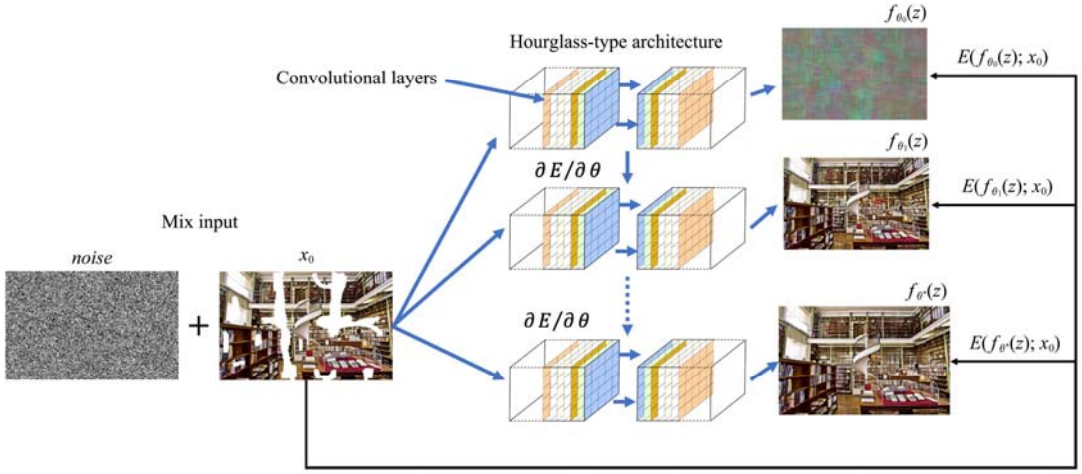


Figure 4 Model outline.

4 Results and discussion

In the contrast experiment, two commonly used standards of evaluating image inpainting performance were used, peak signal-to-noise ratio (PSNR)[16] and structural similarity (SSIM)[17]. PSNR is the most common and widely used objective evaluation method for image reconstruction. However, the PSNR value cannot be completely consistent with the quality of human vision, so in this comparative experimental design, reconstructed images were shown to verify the improved reconstruction performance. In order to show the main improvements in image inpainting (network noise, mix input, weight decay, burning mean output) compared with Deep Image Prior, in the task of large hole painting, an image library was selected to achieve the step-by-step experiment, add improvement strategies in turn, and record the PSNR and SSIM after each improvement, proving the effectiveness of the improved strategy. The improvement strategy was applied sequentially, the PSNR and SSIM values after each improvement were recorded separately, and the effectiveness of the improvement strategy was proved accordingly. In the experiment we also compared the final result with the state-of-the-art algorithms, generating the images shown in Figure 5; PSNR and SSIM are shown in Table 1.

In the binary Bernoulli inpainting experiment, we used the same classic images as Deep Image Prior. Compared with the Deep Image Prior method, which adjusts the parameters of each task separately, this experiment was intended to prove that our model can reconstruct all images in the task with same set of training parameters and performs better than the original algorithm in most inpainting of degraded images. We limited the number of iterations to 20000 times and recorded the PSNR results of each inpainting strategy (using average mean output, burning mean output, average mean output + mix input, burning mean output + mix input + weight decay + network noise), then compared with the state-of-the-art algorithms, and generated the images shown in Figure 6; PSNR is shown in Table 2. On the other hand, since the author has compared the parameters in Deep Image Prior, the best performance one of which was used in this study.

In this research, as mentioned above, one of the key reasons to use the image inpainting algorithm based on Deep Image Prior is to prevent overfitting. Once overfitting occurs, in addition to the deterioration of inpainting performance in human visual judgment, another degradation is decreased PSNR value. In order to verify the improvement of the overfitting phenomenon by our model, the change of PSNR was recorded with the change of iteration times in a binary Bernoulli inpainting experiment. In the result analysis, the PSNR variation curves for three types of output (direct, average mean, burning mean) were drawn to illustrate the advantage of burning mean output (Figure 7).

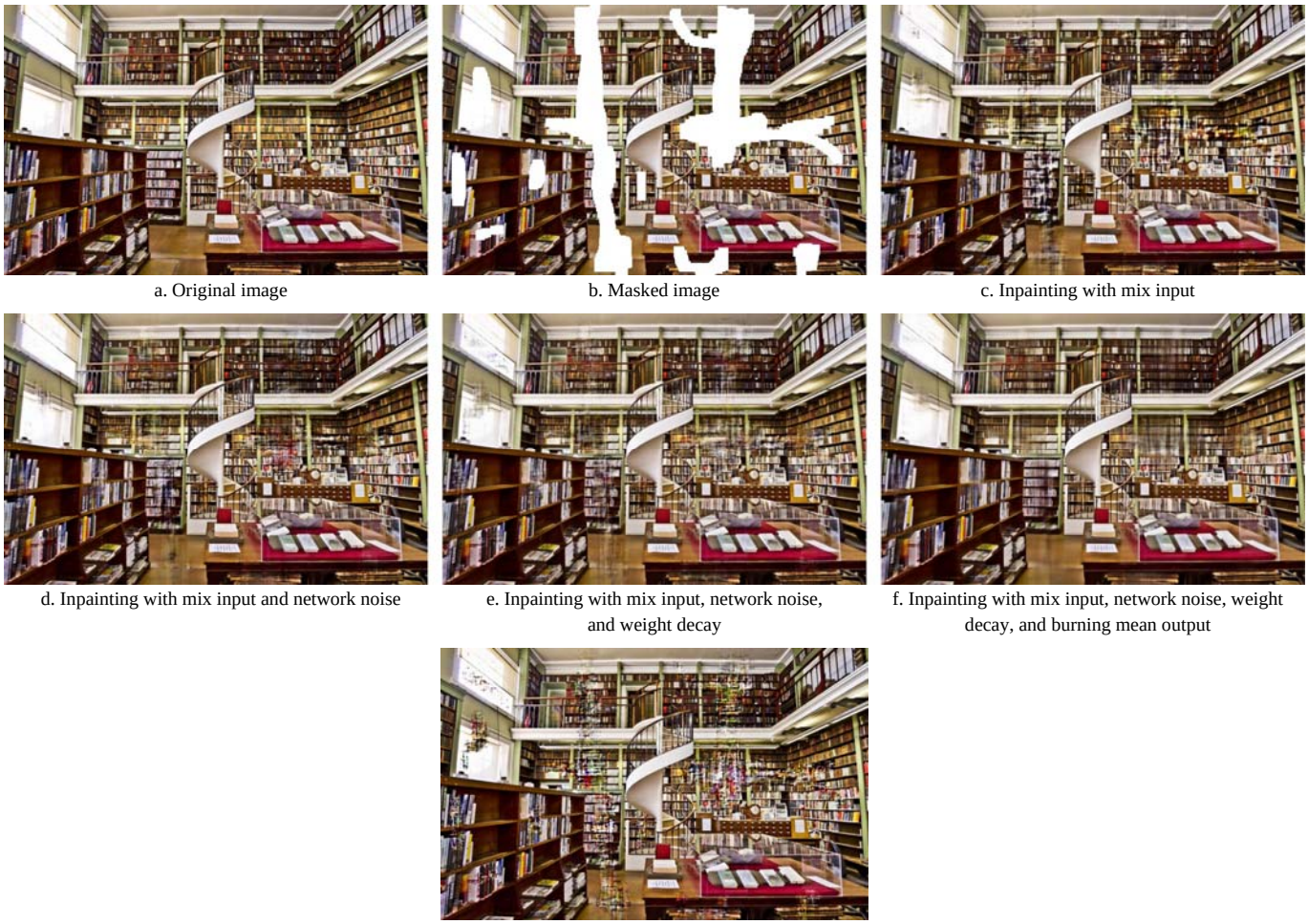
Figure 7 shows the PSNR variation curves of classic images in Deep Image Prior. PSNR increased rapidly before about 1500 iterations, peaked after 2000–2500 iterations, and then began to decline under the influence of overfitting, therefore the performance of image inpainting began to decline after the peak. In contrast, in our improved algorithm, there are no obvious peaks in the curves. Instead, there is a change from fast rising to slow climbing after about 2500 iterations; moreover, there is no trend of decreasing PSNR value in these output curves. Furthermore, from the perspective of output methods, whether in the original Deep Image Prior algorithm or the improved algorithm in this paper, the green curve representing burning mean output performed better; the curve was stable and had a higher PSNR value.

Table 1 Peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) results of large hole inpainting

	ours with mix input	ours with mix input + network noise	ours with mix input + network noise + weight decay	ours with mix input + network noise + weight decay + burning mean output	original Deep Image Prior	CDD [18] (Curvature Driven Diffusions)	Shepard networks[21]
PSNR	19.34	20.54	20.57	**21.80**	18.57	14.52	12.79
SSIM	0.82	0.85	0.85	**0.86**	0.84	0.82	0.80

Table 2 PSNR and SSIM results of binary Bernoulli inpainting

	Barbara	Boat	Lena	Peppers	C. man	Couple	Finger	Hill	Man
GLCIC	12.33	12.90	13.51	13.99	12.58	14.39	12.16	14.39	14.44
Average mean output	28.58	29.76	32.75	29.67	26.66	29.62	30.70	30.10	29.64
Burning mean output	29.15	30.23	33.67	30.77	26.74	29.89	31.02	30.36	29.82
Average mean output + mix input	32.87	32.68	34.21	31.65	27.80	32.29	32.05	32.88	31.90
Papyan et al.	28.14	31.44	35.04	31.11	27.90	31.18	31.34	32.35	31.92
Original Deep Image Prior	32.22	33.06	36.16	33.05	**29.80**	32.52	**32.84**	32.77	32.20
Burning mean output + mix input + weight decay + network noise	**33.46**	**33.17**	**36.17**	**33.35**	28.90	**32.78**	32.71	**33.46**	**32.44**

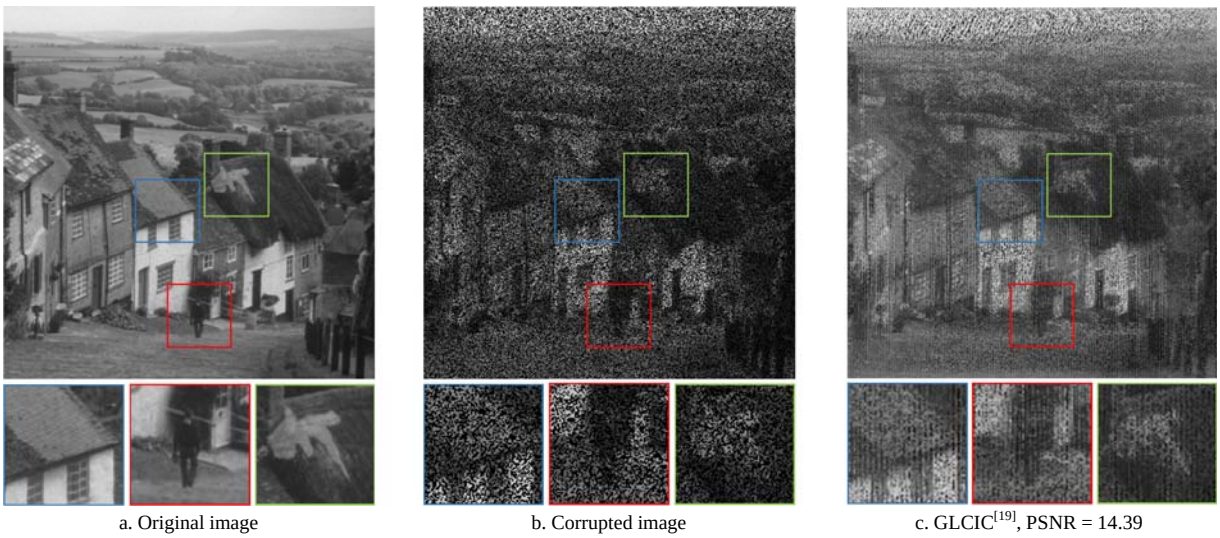


a. Original image b. Masked image c. Inpainting with mix input

d. Inpainting with mix input and network noise e. Inpainting with mix input, network noise, and weight decay f. Inpainting with mix input, network noise, weight decay, and burning mean output

g. Original Deep Image Prior inpainting result

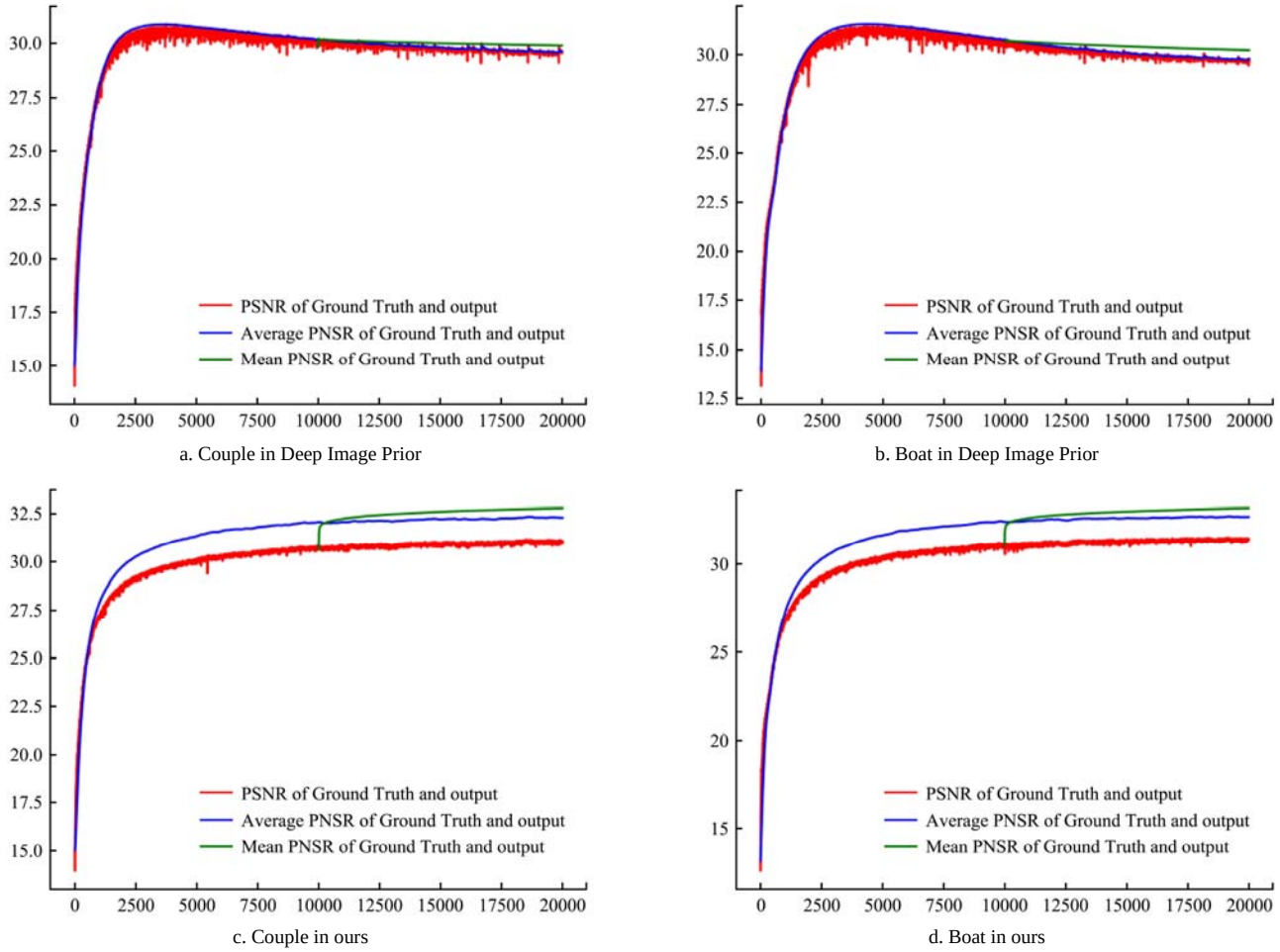
Figure 5 Results of stepwise large hole inpainting experiment



a. Original image b. Corrupted image c. GLCIC[19], PSNR = 14.39

d. Papyan et al[20]., PSNR = 31.34 e. Deep Image Prior, PSNR = 32.84 f. Ours, PSNR = 33.46

Figure 6 Results of binary Bernoulli inpainting experiment



a. Couple in Deep Image Prior b. Boat in Deep Image Prior

c. Couple in ours d. Boat in ours

Figure 7 PSNR variation curves

5 Conclusions

In this paper, based on the original Deep Image Prior algorithm, the convolution structure of the generator was used to obtain image prior information and perform the image inpainting task. Four improvements are proposed (mix input, network noise, weight decay, and burning mean output) in order to prevent overfitting while stabilizing the output. Two stepwise comparative experiments, large hole inpainting and binary Bernoulli inpainting, were used to compare our algorithm, Deep Image Prior, and state-of-the-art algorithms. In the results of the large hole inpainting experiment, the PSNR of our algorithm was 3.23 dB higher than that of Deep Image Prior. In binary Bernoulli inpainting, we used the same set of parameters for all classic images and in most cases exceeded the original and state-of-the-art algorithms after applying all of the improvement strategies. This experiment also proved that burning mean output could stabilize the output and avoid the interference of meaningless noise generated in the previous iterations in subsequent image inpainting performance.

[References]

[1] Xiang Y, Yu B, Yuan Q, et al. GPU acceleration of CFD algorithm: HSMAC and SIMPLE. Procedia Computer Science, 2017, 108: 1982–1989. doi: 10.1016/j.procs.2017.05.124

[2] Pouyanfar S, Sadiq S, Yan Y, et al. A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys (CSUR), 2018, 51(5): 1–36. doi: 10.1145/3234150

[3] Li Z, Yang W, Peng S, et al. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. arXiv preprint arXiv:2004.02806, 2020.

[4] Li S, Song W, Fang L, et al. Deep learning for hyperspectral image classification: An overview. IEEE Transactions on Geoscience and Remote Sensing, 2019, 57(9): 6690–6709. doi: 10.1109/TGRS.2019.2907932

[5] Zhao Z Q, Zheng P, Xu S, et al. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 2019, 30(11): 3212–3232. doi: 10.1109/TNNLS.2018.2876865

[6] Liu L, Wang H, Li G, et al. Crowd counting using deep recurrent spatial-aware network. arXiv preprint arXiv:1807.00601, 2018.

[7] Garcia-Garcia A, Orts-Escolano S, Oprea S, et al. A survey on deep learning techniques for image and video semantic segmentation. Applied Soft Computing, 2018, 70: 41–65. doi: 10.1016/j.asoc.2018.05.018

[8] Gong K, Guan J, Kim K, et al. Iterative PET image reconstruction using convolutional neural network representation. IEEE transactions on medical imaging, 2018, 38(3): 675–685. doi: 10.1109/TMI.2018.2869871

[9] Wei B, Sun X, Ren X, et al. Minimal effort back propagation for convolutional neural networks. arXiv preprint arXiv:1709.05804, 2017.

[10] Ulyanov D, Vedaldi A, Lempitsky V. Deep image prior. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018: 9446–9454.

[11] Kurrant D, Baran A, LoVetri J, et al. Integrating prior information into microwave tomography Part 1: Impact of detail on image quality. Medical physics, 2017, 44(12): 6461–6481. doi: 10.1002/mp.12585

[12] Ranjan R, Singh A, Rizvi A, et al. Classification of Chest Diseases Using Convolutional Neural Network. Proceedings of First International Conference on Computing, Communications, and Cyber-Security (IC4S 2019). Springer, Singapore, 2020: 235–246.

[13] Zhang C, Bengio S, et al. Understanding deep learning requires rethinking generalization. In Proc. ICLR, 2017

[14] Dubey A K, Jain V. Comparative Study of Convolution Neural Network's ReLu and Leaky-ReLu Activation Functions. Applications of Computing, Automation and Wireless Systems in Electrical Engineering. Springer, Singapore, 2019: 873–880. doi: 10.1007/978-981-13-6772-4_76

[15] Zhang X, Zou Y, Shi W. Dilated convolution neural network with LeakyReLU for environmental sound classification. 2017 22nd International Conference on Digital Signal Processing (DSP). IEEE, 2017: 1–5.

[16] Liu N, Zhai G. Free energy adjusted peak signal to noise ratio (FEA-PSNR) for image quality assessment. Sensing and Imaging, 2017, 18(1): 11.

[17] Hore A, Ziou D. Image quality metrics: PSNR vs. SSIM. 2010 20th international conference on pattern recognition. IEEE, 2010: 2366–2369.

[18] Li L I. The inpainting model by Curvature-Driven Diffusions (CDD). Computer Knowledge and Technology, 2006.

[19] Iizuka S, Simo-Serra E, Ishikawa H. Globally and locally consistent image completion. ACM Transactions on Graphics, 2017, 36(4): 1–14.

[20] Papyan V, Romano Y, Elad M, et al. Convolutional Dictionary Learning via Local Processing. 2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017.

[21] Ren J，Xu L，Yan Q， Sun W. Shepard convolutional neural networks. In Proc. NIPS, pages 901–909, 2015.