information

Article

Progressive-Augmented-Based DeepFill for High-Resolution Image Inpainting

Muzi Cui [1], Hao Jiang [2] and Chaozhuo Li [3,*]

[1] College of Cyber Security, Jinan University, Guangzhou 511436, China; cuimuzi@stu2020.jnu.edu.cn
[2] Stevens Institute of Technology, Hoboken, NJ 07030, USA; hjiang23@stevens.edu
[3] Microsoft Research Asia, Beijing 100080, China
* Correspondence: cli@microsoft.com

Abstract: Image inpainting aims to synthesize missing regions in images that are coherent with the existing visual content. Generative adversarial networks have made significant strides in the development of image inpainting. However, existing approaches heavily rely on the surrounding pixels while ignoring that the boundaries might be uninformative or noisy, leading to blurred images. As complementary, global visual features from the remote image contexts depict the overall structure and texture of the vanilla images, contributing to generating pixels that blend seamlessly with the existing visual elements. In this paper, we propose a novel model, PA-DeepFill, to repair high-resolution images. The generator network follows a novel progressive learning paradigm, starting with low-resolution images and gradually improving the resolutions by stacking more layers. A novel attention-based module, the gathered attention block, is further integrated into the generator to learn the importance of different distant visual components adaptively. In addition, we have designed a local discriminator that is more suitable for image inpainting tasks, multi-task guided mask-level local discriminator based PatchGAN, which can guide the model to distinguish between regions from the original image and regions completed by the model at a finer granularity. This local discriminator can capture more detailed local information, thereby enhancing the model's discriminative ability and resulting in more realistic and natural inpainted images. Our proposal is extensively evaluated over popular datasets, and the experimental results demonstrate the superiority of our proposal.

Keywords: image inpainting; deep learning; generative adversarial networks

Citation: Cui, M.; Jiang, H.; Li, C. Progressive-Augmented-Based DeepFill for High-Resolution Image Inpainting. *Information* **2023**, *14*, 512. https://doi.org/10.3390/info14090512

Academic Editor: Marco Leo

Received: 11 August 2023
Revised: 13 September 2023
Accepted: 15 September 2023
Published: 18 September 2023

1. Introduction

Image inpainting refers to the process of synthesizing missing regions in images in a manner that is consistent with existing visual content. This field plays a crucial role in facilitating a variety of real-life applications such as image recovery [1], image editing [2], image relocation [3], and image registration [4]. The focus of image inpainting is to generate visually plausible regions with high resolutions while ensuring global semantic consistency. The key objective of image inpainting is to generate visually plausible regions with high resolution while ensuring global semantic consistency [5].

Traditional approaches to image inpainting often rely on infilling images with similar pattern blocks or manually-crafted shallow features. These methods include diffusion-based approaches [6–8] and patch-based approaches [9–12]. However, these heuristic methods tend to introduce bias when the remaining content is not closely related to the missing area. As a result, they often fail to maintain consistency between the generated visual features and the overall image semantics [13,14]. Recently, deep convolutional neural networks (CNN) [15,16] and generative adversarial networks (GAN) [17] exhibit great potential in image inpainting [18–20].

These techniques are capable of generating realistic new content in highly structured images.

Deep learning based approaches contribute to bridging the semantic gap between the low-level features and the high level semantics [21] to some extent. Unfortunately, existing methods often generate distorted structures with obvious boundary artifacts or even blurred textures when repairing images. As shown in Figure 1, the content repaired by several SOTA methods (CA [19], PConv [22], and Deepfillv2 [20]) are obviously distorted and blurred. The underlying reason of such undesirable performance is that existing methods heavily rely on the boundary information or the local visual features to make predictions. First, these methods tend to propagate boundaries to infill the blanks, but boundary signals are usually useless or even noisy, leading to the blurred images [18]. Second, existing works cannot effectively incorporate the global visual features from the remote image contexts to infer reasonable contents, resulting in the distorted structures [23,24].

(**a**) Original Image (**b**) Masked Image (**c**) CA (**d**) PConv (**e**) Deepfillv2

Figure 1. An illustration of different inpainting models. Given an image (**a**) and its masked version (**b**), CA [19] (**c**), PConv [22] (**d**), and Deepfillv2 [20] (**e**) generate distorted visual features.

To address the mentioned challenges, in this paper we propose a novel image infilling model, dubbed **P**rogressive-**A**ugmented-based **DeepFill (PA-DeepFill)**. Different from existing approaches, we have integrated the progressive and two-stage image inpainting strategies. Considering it is intractable and challenging to predict the missing pixels directly, we propose to first resolve the missing content in the coarse-grained granularity, a much easier task while preserving the uncertainty over more difficult generations. After that, the fine-grained visual pixels are further generated to ensure high resolution and coherence. Meanwhile, to capture the global informative contexts and long-distance interest patterns, a novel gathered attention block is integrated into the generator to learn the global attention mechanism weight for each pixel and each channel on the feature maps. By using Gaussian filtering operations, we can guide the model to segment the real pixels and synthesized pixels in the image in a soft manner. The multi-task guided model assigns corresponding weights to each synthesized region. Our proposal is thoroughly evaluated over several datasets, including CelebA [25] and COCO [26], and PA-DeepFill achieves SOTA performance. The major contributions are summarized as follows:

- We propose a novel progressive image inpainting paradigm, which is capable of generating high-resolution content for the missing area under an end-to-end training framework.
- We design a novel gathered attention block, which learns the global importance of different pixels and channels in the semantic feature maps of various scales.
- We have designed a discriminator for image inpainting tasks that effectively separates real pixels and synthesized pixels in the completed image and learns different weights for each synthesized region block based on the matching degree of the synthesis, thereby improving the inpainting effect of the model.
- We conduct extensive experiments, and the results demonstrate the superiority of our proposal in image reconstruction accuracy and visual authenticity.

2. Related Works

Image inpainting aims to restore image defects with reasonable content. In recent years, this field has received significant attention from the research community, primarily due to its valuable applications. Image inpainting algorithms can be broadly categorized into two main categories. One category includes traditional methods that rely on similar pattern blocks or other shallow features. These algorithms utilize the correlation between

image pixels and the similarity of content to fill in the gaps. The second category consists of deep learning-based methods, which leverage advanced techniques to learn meaningful features from a knowledge base. These methods utilize deep neural networks to generate high-quality inpainted images by understanding the underlying structure and context of the image.

2.1. Traditional Image Inpainting Methods

Traditional image inpainting methods primarily rely on the correlation between image pixels and the similarity of content. These methods can be further categorized into diffusion-based methods and patch-based methods, based on different optimization criteria.

2.1.1. Diffusion-Based methods

Diffusion-based methods in the field of image inpainting aim to propagate information from the surrounding areas of the damaged or missing regions to fill in the gaps [6–8]. The core idea behind these methods is to model the diffusion process, where the image information is gradually spread from known regions to the unknown regions. Among these methods, the most classic one is the Bertalmio–Sapiro–Caselles–Bellester (BSCB) model proposed by Bertalmio et al. [6]. This model is based on partial differential equations and uses pixels as the fundamental unit. It extends the isophote direction of the missing boundary of the image by diffusing the Laplacian feature of the known image into the missing area. It achieves image completion through continuous iterative repair. The above partial differential equation and variational method can be derived equivalently through the variational principle. This method is particularly effective for processing small-scale damaged images. However, when dealing with larger missing areas or areas with complex texture information, the resulting repair effect tends to be blurry. There are several main reasons for this:

- The algorithm models the image in the variational space, and regards the image as a piecewise smooth function, which can achieve continuous structure but does not contain any texture information.
- The algorithm is essentially a process of diffusion from the edge of the missing area to the interior. Once the area to be repaired is large or the texture is complex, it will fail [27].

2.1.2. Patch-Based Methods

Patch-based methods in the field of image inpainting aim to fill in missing or damaged regions by searching for and copying patches from the surrounding areas that are visually similar to the missing region [9–12,28–30]. The core idea behind these methods is to exploit the redundancy and self-similarity present in natural images. The most representative method is the Criminisi algorithm, proposed by Criminisi et al. in 2003 [9]. It is the most classical method for texture synthesis. However, patch-based algorithms do have certain drawbacks. These methods require a large amount of computation during the search process, and they rely on external databases or networks. When there are no similar and reasonable image patches available in the external database or network, these methods can lead to incorrect repairs.

Traditional image inpainting methods excel at repairing small missing areas with simple structures and textures. However, they may fall short when it comes to semantically repairing large missing areas in complex scenes. This is primarily due to the heavy reliance of patch-based methods on matching patches based on low-level features. These techniques do not effectively synthesize content that fits seamlessly within the known image context.

2.2. Deep Learning-Based Methods

Deep learning-based image inpainting models have shown significant improvements over traditional techniques in synthesizing realistic content for complex scenes and large damaged areas. By leveraging the power of deep neural networks, these models can

effectively capture high-level features and semantic information to generate more visually plausible inpainted results. This allows for more seamless integration of synthesized content within the surrounding context of the image.

Pathak et al. [29] proposed that Context Encoder (CE) is a start-up for image inpainting based on deep learning. The author used image impainting loss and anti-loss as constraints to improve the image inpainting effect [31,32]. This work has provided inspiration and a foundation for subsequent research. Iizuka et al. [30] preserves the discriminators in the Context Encoder network as local discriminators and adds global discriminators for the entire image area. The collaboration of two discriminators, each focusing on different regions, allows the repair network to effectively handle images with arbitrary irregular shapes and large defect areas. This approach addresses the issues of image boundary distortion and local blurring commonly observed in CE repair images. Building upon this foundation, Yu et al. [19] proposed a network architecture consisting of Coarse Network–Refinement Network, along with the integration of the attention mechanism [33]. This integration further enhanced the effectiveness of image inpainting. In a similar vein, Zeng et al. [18] incorporated a non-local module named the attention transfer network to inpaint missing regions in the feature pyramid.

To further improve image fidelity, many recent works employ generative adversarial networks (GAN) [34–36]. In these models, the discriminator of the GAN is trained to distinguish between restored and real images, while the generator is optimized to synthesize realistic images that can deceive the discriminator. By leveraging the adversarial game theory between the generator and discriminator, GAN-based inpainting models can generate more realistic textures. Notably, one successful GAN-based image inpainting model, called PatchGAN [37], has achieved remarkable success in image translation tasks. More specifically, PatchGAN's discriminator is specifically designed to differentiate between patches extracted from real images and patches generated by the inpainted images. To enhance the stability of GAN training, spectral normalization is applied to each layer of the discriminator. This normalization technique helps to regularize the network's weights, leading to more stable and reliable training process for the GAN-based image inpainting model. Isola et al. [37] utilized PatchGAN's discriminator and conditional generation adversarial network to accomplish the image-to-image translation task, which has provided inspiration for subsequent image inpainting tasks. Yu et al. [20] improved the mask update process of partial convolutional network [22] under the discriminator structure inherited from PatchGAN. Thus, they applied spectral-normalized discriminator on dense image patches, which the image inpainting effect can be further improved. Zeng et al. [18] proposed multi-path convolution to replace the common single convolution, different dilation rate parameters are set for each convolution path, and then the features obtained by convolution of each path are aggregated so that each pixel can obtain different scale features. It effectively solves the problems of obtaining context reasoning from image holes' distant contexts and synthesis of fine-grained texture of large area missing region. Guo et al. [38] introduced a Bi-directional Gated Feature Fusion (BGFF) module to facilitate the exchange and fusion of structure and texture information. This module enables the effective integration of structural and textural features, allowing the model to capture and leverage complementary information from both domains. By incorporating the BGFF module, the model can achieve enhanced performance through the effective combination of structural and textural cues in the image.

Additionally, there have been efforts focused on generative facial inpainting. Yeh et al. [31] searched for the closest encoding in the potential space of the damaged image and decoded it to obtain a complete image. Li et al. [32] introduced additional loss based on facial analysis to perform facial inpainting. However, these methods often require post-processing steps, such as color blending, to enhance color consistency near the boundaries of the inpainted regions.

3. Methodology

As shown in Figure 2, PA-DeepFill consists of two generator networks and one discriminator network. The first generator adopts a progressive learning paradigm, gradually extracting different levels of contextual semantic features by reconstructing the corresponding top layers to enhance the inpainting performance. Specifically, the first-stage generator is composed of multiple gathered attention block, including pixel attention block and channel attention block, which learn global attention weights for different pixels and channels in the feature map. These modules help capture informative contexts and distant interest patterns, facilitating context reasoning. The results of the first-stage inpainting are further enhanced by a designed image enhancement module. The second generator takes the original input image with a mask and the output image processed by the image enhancement module to synthesize fine-grained textures. The discriminator is responsible for supervising the generator to generate textures that conform to contextual semantics. With the well-designed mask prediction, the discriminator effectively improves the synthesized textures, making them more realistic. Through the joint optimization of several carefully designed loss functions, PA-DeepFill can synthesize images with contextual semantics and clear textures in the large missing areas of high resolution.

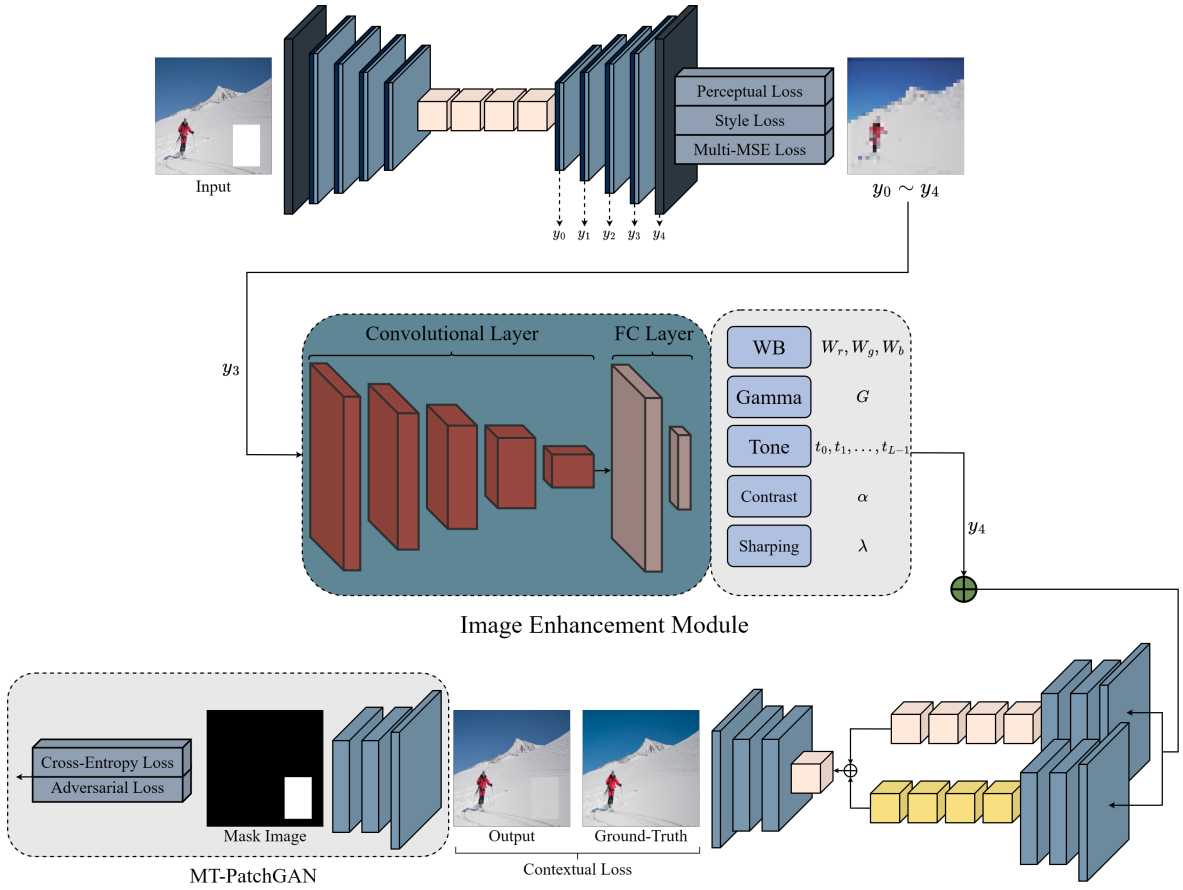


Figure 2. The overview of the proposed PA-DeepFill model.

3.1. Gathered Attention Block

The gathered attention block is the fundamental component of the generator, which learns attention weights for pixels and channels in semantic feature maps of various scales. Specifically, the gathered attention block consists of two major modules: the pixel attention block and the channel attention block.

3.1.1. Pixel Attention Block

The generated content is expected to be conform to the context semantics consistency, while directly incorporating the entire image features might introduce extra noise [39,40]. Thus, we propose the pixel attention block to adaptively focus on the most informative global pixel-level patterns of interest.

As shown in Figure 3, a pixel attention block is composed of the global pooling layer, the 1×1 and the 3×3 convolution layers. The major steps are as follows:

- Perform 1×1 convolution on the input feature image;
- The 3×3 convolutional layer reduces the dimension of the feature map channel, which is equivalent to sparse coding of the feature map;
- Each pixel point of the input feature map is multiplied by the corresponding pixel attention weight to obtain the final feature map.

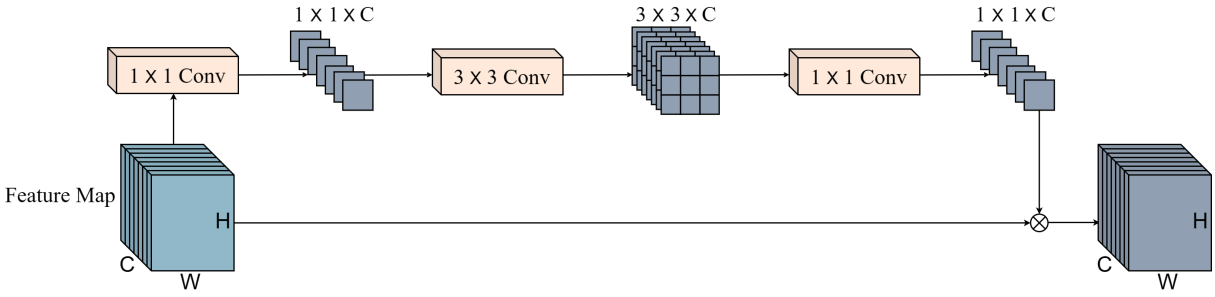


Figure 3. The framework of pixel attention block.

Through the above three steps, the pixel attention block empowers the generator to capture the pixel-level semantic features, improving the context reasoning ability.

3.1.2. Channel Attention Block

Different semantic features may play different roles in image inpainting process. However, previous works [20] incorporate all the information equally, leading to the fuzzy textures. To identify the informativeness of various semantic features, we further design a channel attention block to learn the importance of different feature channels. As shown in Figure 4, the channel attention block contains the following three steps:

- The input feature map is compressed into a one-dimensional vector after passing through the global max-pooling layer;
- Employ 1×1 convolution to learn an attention weight for each channel dimension of the original feature map;
- Multiply each channel by the corresponding attention weight to obtain the final feature map.

With the channel attention blocks, the generator network is capable of incorporating helpful channel-level global features while avoiding the potential noise.

The outputs from these two attention-based modules are combined together by element-wise addition as the final output. Gathered attention blocks capture the informative global visual semantics, which contribute to reducing the dependency on the boundary information by precisely incorporating the distant features of different levels as complementary.

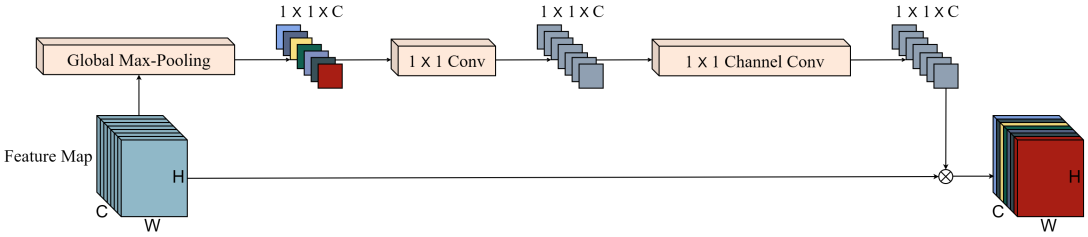


Figure 4. The framework of channel attention block.

3.2. Image Enhancement Module

The Image Enhancement Module is divided into two parts as shown in Figure 2: the adaptive parameter control network and the multi-dimensional image enhancement network.

3.2.1. Adaptive Parameter Control Network

The adaptive parameter control network consists of five convolutional blocks and two fully connected layers. Each convolutional block consists of a 3×3 convolutional layer with a stride of 2 and a Leaky ReLU activation function. The input to the model is a 256×256 resolution image. By using low-resolution images, the model can predict the parameter information required by the multi-dimensional image enhancement network while significantly reducing computational cost [41]. The final output of this network consists of the hyperparameters required by the multi-dimensional image enhancement network.

3.2.2. Multi-Dimensional Image Enhancement Network

The multi-dimensional image enhancement network consists of five differentiable filters with adjustable hyperparameters, including white balance, gamma correction, contrast, tone, and sharpening. Following previous works [41,42], we represent white balance, gamma correction, contrast, and tone operations as pixel-wise filters and represent the sharpening operation as a sharpening filter.

Pixel-wise filters. In order to generate images that are more visually appealing to human perception, we use pixel-wise filters to apply specific operations that map the input pixel values $P_i = (r_i, g_i, b_i)$ to the output pixel values $P_o = (r_o, g_o, b_o)$, where (r, g, b) represent the red, green, and blue color channels, respectively. Table 1 shows the mapping function relationships for pixel-wise filters.

Table 1. Pixel-wise filter mapping function.

Filter	Parameters	Mapping Function
White Balance	W_r, W_g, W_b	$P_o = (W_r r_i, W_g r_g, W_b r_b)$
Gamma	G	$P_o = P_i^G$
Contrast	α	$P_o = \alpha \cdot En(P_i) + (1 - \alpha) \cdot P_i$
Tone	t_i	$P_o = (L_{t_r}(r_i), L_{t_g}(g_i), L_{t_b}(b_i))$

The design of a contrast filter includes an input parameter to set linear interpolation between the original image and the fully enhanced image. As shown in Table 1, the definition of $En(P_i)$ in the mapping function is as follows:

$$Lum(P_i) = 0.25r_i + 0.65g_i + 0.1b_i \tag{1}$$

$$EnLum(P_i) = \frac{1}{2}(1 - \cos(\pi \times (Lum(P_i)))) \tag{2}$$

$$En(P_i) = P_i \times \frac{EnLum(P_i)}{Lum(P_i)} \tag{3}$$

Based on previous work [41–43], we set the mapping function of the color tone filter as a monotonic piecewise linear function. Let L be the number of parameters for the color tone filter, then all the parameters can be represented as $t_0, t_1, \ldots, t_{L-1}$. The points on the color tone curve are represented as $(k/L, T_k/T_L)$, where $T_k = \sum_{i=0}^{k-1} t_l$. The mapping function of the color tone filter can be expressed as:

$$P_o = \frac{1}{T_L} \sum_{j=0}^{L-1} clip(L \cdot P_i - j, 0, 1) t_k \tag{4}$$

Sharpening filter. Image sharpening can better highlight the details of the inpainted image [44], thereby improving the image inpainting performance of the model. The image sharpening operation can be represented as:

$$F(x, \lambda) = I(x) + \lambda(I(x) - Gau(I(x))) \tag{5}$$

where $I(x)$ represents the input image, $Gau(I(x))$ represents the Gaussian filtering operation applied to the input image, and λ is a positive scaling factor. By optimizing the scaling factor λ, the degree of sharpening can be adjusted to obtain a restored image with richer details.

3.3. Generator Network

PA-DeepFill consists of two generator networks: the first-stage generator adopts a progressive generation strategy to generate rough restoration results; the second-stage generator is responsible for receiving the rough restoration results enhanced by image quality improvement, and further optimizing the restoration quality of the image to make the generated image more refined and realistic.

3.3.1. The First-Stage Generator

The generator in the first stage of PA-DeepFill adopts a progressive generation strategy, which avoids learning all features simultaneously like traditional GANs. Following the general idea of "from coarse to fine", this progressive paradigm subjectively allows us to first learn the coarse structural features (low resolution) of the image distribution, and then gradually learn the fine-grained details (high resolution) of the image after feature learning, rather than learning all scale features at the same time. As shown in Figure 3, the outputs from the last five layers of the generator are divided into y_0, y_1, y_2, y_3, and y_4. The size of the generated feature map is 32×32, 64×64, 128×128, 256×256, 512×512, and the original image size is downsampled to the corresponding feature map size. Then, multi-level MSE loss can be calculated. Ideally, the generator will generate infilling content of different resolutions following y_0, y_1, y_2, y_3 and y_4, respectively. However, due to the phased nature of network training, the current resolution is directly transferred to a higher-resolution network architecture, and the network may collapse due to the weight mismatch of newly added layers. To solve this problem, we further design the Smooth Transition (ST) module. As shown in Figure 5, when the resolution is transferred from 32×32 to 64×64, we control the ST module to achieve a gradual process to avoid sudden network collapse. As shown in Figure 5b, after generator G generates an image with a resolution of 32×32, in order to generate an upsampled image with a resolution of 64×64, it is divided into three branch paths. The first branch passes through a convolution layer and a sigmoid activation function to generate an attention weight α. The second branch uses a nearest neighbor interpolation algorithm to upsample the image with a resolution of 32×32 by $2\times$. In the third branch, we introduce a transposed convolution layer. After the transposed convolution, 32×32 images are also upsampled twice. The final up-sampled images are the weighted combination of the images from the three paths.

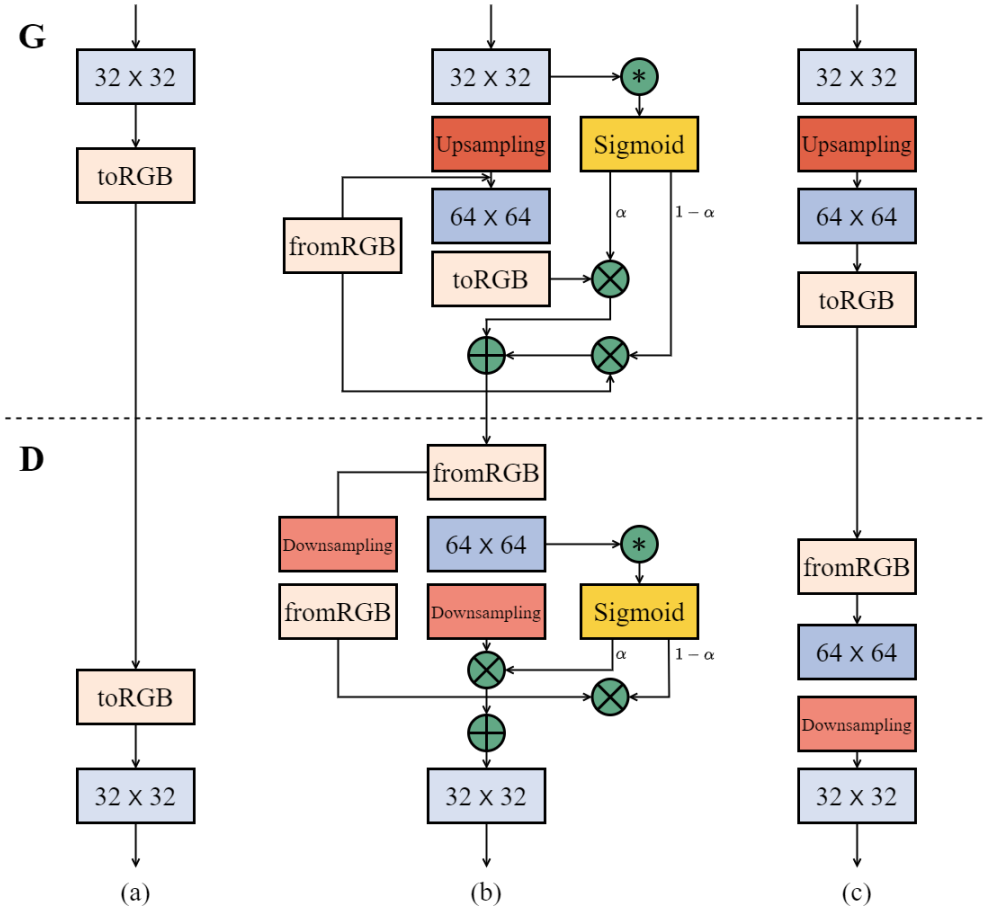


Figure 5. The frameworkof smooth transition (ST) module. Where (**a**) is 32 × 32, (**b**) is the transition layer from 32 × 32 to 64 × 64, and (**c**) is 64 × 64.

3.3.2. The Second-Stage Generator

The goal of the second-stage generator is to further optimize the restoration quality based on the rough restoration results, making the generated image more refined and realistic. To enhance the effectiveness of image inpainting, we employ the gated convolution proposed by Yu et al. [20] as a replacement for the conventional convolution operation in the network. The expression of gated convolution is as follows:

$$\begin{aligned} Gating_{y,x} &= \sum\sum W_g \cdot I \\ Feature_{y,x} &= \sum\sum W_f \cdot I \\ O_{y,x} &= \phi(Feature_{y,x}) \odot \theta(Gating_{y,x}) \end{aligned} \tag{6}$$

where θ represents the sigmoid function, so the output of the gated convolution is between 0 and 1, and ϕ represents the ReLU activation function in this paper. Compared to traditional convolution operations, gated convolution introduces a gating mechanism that adaptively controls the information flow in the network. This allows the network to better capture long-range dependencies and reduce interference from irrelevant information. In the task of image inpainting, using gated convolution can enhance the generator's ability to capture long-distance image features and synthesize more fine-grained textures, making the restoration results appear more natural and continuous, further enhancing the effectiveness of image inpainting. The fine restoration network also includes contextual attention blocks, which is used to learn contextual information from the non-hole regions of the input image. By minimizing the adversarial loss, pixel-level reconstruction loss and contextual loss, the fine restoration network can generate preliminary restoration results.

3.4. MT-PatchGAN

In order to further enhance the discriminator's ability to discriminate information from the inpainted regions, we propose an improved version of the mask-level local discriminator based on the work of Zeng et al. [18]. We refer to this approach as the **M**ulti-**T**ask guided mask-level local discriminator-based **PatchGAN (MT-PatchGAN)**. Specifically, this approach utilizes mask downsampling as the ground truth data for the image inpainting prediction task. Furthermore, a finer-grained training of the discriminator is achieved by employing a mask generated through Gaussian filtering. Instead of merely padding the output matrix of the PatchGAN with binary values, the discriminator is trained using a mask obtained through Gaussian filtering. This approach effectively separates the inpainted regions into real and inpainted parts, enabling the assessment of inpainting quality in the inpainted regions. It enhances the network's capacity to capture long-range features. Consequently, the strengthened discriminator promotes the generation of clearer and visually consistent textures. By introducing MT-PatchGAN, the network can better discriminate the inpainted regions and provide more informative feedback to the generator, leading to improved inpainting results.

3.5. End-to-End Optimization

We represent the corresponding binary completion mask as b, where 1 represents the known pixels and 0 represents the missing pixels. The real image is denoted as x, the generated image as z, the generator as G, and the discriminator as D. The $\odot$ operator denotes the element-wise multiplication operation. The result of image completion can be represented as:

$$z = x \odot b + G(x \odot b, b) \odot (1 - b) \tag{7}$$

To further optimize the visual realism of the images, the model employs six loss functions as optimization strategies, i.e., a multi-MSE loss, a style loss [45], a contextual loss, a perceptual loss [46], a cross-entropy loss, and an adversarial loss based on MT-PatchGAN. Specifically, the multi-MSE loss ensures the reconstruction area conform to the visual context in different levels. A single Mean Squared Error (MSE) loss is defined as: $L_{MSE} = \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$, in which n denotes the resolution of the feature map, y_i represents the pixel value distribution of the feature map, $\widehat{y}_i$ represents the pixel value distribution of the real picture. After combining the MSE losses from various levels, the proposed multi-MSE loss is formally defined as follows:

$$L_{multi-MSE} = \sum_{j=0}^{k} \lambda_i \times L_{j\_MSE}, \tag{8}$$

where k denotes the number of hidden layers in the generator, $\lambda_0 = 0.0001$, $\lambda_1 = 0.001$, $\lambda_2 = 0.01$, $\lambda_3 = 0.1$, and $\lambda_4 = 1$. Similarly, the contextual loss is measured by computing the pixel-level differences between the generated and real images to capture their similarity. The specific expression of the contextual loss is as follows:

$$L_{con} = \frac{1}{W} \sum_{i=1}^{W} \sum_{j=1}^{H} (z_{i,j} - x_{i,j})^2 \tag{9}$$

where x and y represent the generated image and the real image, respectively; W and H represent the width and height of the image; and i and j represent the indices of the pixels. This loss function calculates the squared sum of the differences at each pixel position, resulting in a scalar value that measures the similarity between the generated image and the real image.

The perceptual loss and style loss, based on the pre-trained VGG-19 network [47,48], are used to capture the high-level features and structural similarity between the generated and real images. By extracting feature maps from the VGG-19 network, the differences

between the feature maps of the generated and real images are computed. Specifically, the expression for the perceptual loss is as follows:

$$L_{per} = \sum |F(z) - F(x)|. \tag{10}$$

The style loss is as follows:

$$L_{sty} = \sum |Gram(F(z)) - Gram(F(x))|, \tag{11}$$

where $F(\cdot)$ is the output feature map of a specific layer in the VGG-19 network, and $Gram(\cdot)$ is the Gram matrix of the feature map.

To further enhance the image inpainting capability, two loss functions are used to optimize the discriminator: the cross-entropy loss function and the adversarial loss function for MT-PatchGAN. The cross-entropy loss function is employed to evaluate the similarity between the inpainted region and the original region, enabling the discriminator to distinguish between real and generated images. The cross-entropy loss function for the discriminator is expressed as:

$$L_{ce} = -[(x \log(z) + (1 - x) \log(1 - z))] \tag{12}$$

Introducing an adversarial loss function can enhance the discriminator's focus on the missing regions and disregard the influence of the inpainting boundaries. The adversarial loss function for the discriminator can be expressed as:

$$L_{adv}^{D} = E_{z \sim p_z}\left[(D(z) - \delta(b))^2\right] + E_{x \sim p_{data}}\left[(D(x) - 1)^2\right] \tag{13}$$

where δ represents the processing of the mask used for training, including downsampling and Gaussian filtering. Correspondingly, the adversarial loss for the generator is denoted as:

$$L_{adv}^{G} = E_{P \sim p_z}\left[(D(x) - 1)^2\right]. \tag{14}$$

The overall loss function is the weighted combination of these six losses:

$$\begin{aligned} L = \lambda_{Multi-MSE} L_{Multi-MSE} + \lambda_{con} L_{con} + \lambda_{per} L_{per} + \\ \lambda_{sty} L_{sty} + \lambda_{ce} L_{ce} + \lambda_{adv} L_{adv}^{G}. \end{aligned} \tag{15}$$

where λ_i represents the weights of different losses. For our experiments, we empirically choose $\lambda_{Multi\_MSE} = 1$, $\lambda_{con} = 0.1$, $\lambda_{per} = 0.1$, $\lambda_{sty} = 250$, $\lambda_{con} = 0.1$, $\lambda_{ce} = 0.01$, and $\lambda_{adv} = 0.01$ for training.

4. Experiments

4.1. Experimental Settings

4.1.1. Datasets

We conduct experiments on three large public datasets, namely COCO [26], CelebA [25], and QMUL-OpenLogo [49]. Following previous works [26], we randomly select 10,000 images from each dataset as the validation and testing sets.

4.1.2. Baselines

Our proposal is extensively compared with popular SOTA baselines, including **CA** [19], **PConv** [22], **Deepfillv2** [20], **AOT-GAN** [18], and **CTSDG** [38].

4.1.3. Parameter Settings

We implement the first-stage generator network of PA-DeepFill using an encoder network composed of three convolutional layers, which performs four times downsampling on the input image. Correspondingly, we use a three-layer deconvolutional generator to

upsample the feature map generated by the encoder network. For the discriminator network, we build it by using four convolutional layers with a stride of two, stacked on a 70×70 PatchGAN network. We then construct the discriminator network. We perform Gaussian filtering on the discriminator network using a 70×70 Gaussian kernel.

4.2. Quantitative Analysis

To ensure a fair comparison, for each test image, we randomly select a mask from a set of freely-formed masks provided by Liu et al. [22] to serve as the testing mask. This random selection of masks ensures that the comparison is conducted under the same conditions.

From Tables 2 and 3, it can be observed that our proposed model outperforms other state-of-the-art inpainting models in terms of image inpainting with larger missing areas (i.e., the mask area is greater than 30%). We believe this is due to the adoption of a progressive strategy, which reduces the inpainting span and thus lowers the difficulty of the inpainting task. Additionally, our designed image enhancement module effectively improves the results of rough inpainting, further enhancing the subsequent inpainting performance. MT-PatchGAN forces the discriminator to focus on the inpainted regions, and the enhanced discriminative ability encourages the generator to synthesize finer texture details. Comparing Tables 2 and 3, it can be observed that all models exhibit worse inpainting results on the COCO dataset compared to the CelebA dataset. This is because the COCO dataset contains more complex image scenes with a higher amount of detailed textures.

Table 2. Quantitative comparison on CelebA. The best and the second best results are **bolded** and underlined.

Mask(%)		CA	Deepfillv2	PConv	CTSDG	AOT–GAN	PA–DeepFill
$L_1(10^{-2})\downarrow$	0–10	1.57	1.31	1.43	**0.55**	1.14	1.23
	10–20	2.19	2.01	2.09	**1.29**	1.89	1.81
	20–30	3.69	3.31	3.47	**2.31**	2.76	2.33
	30–40	4.56	4.11	4.17	3.44	3.91	**3.23**
	40–50	5.33	5.14	5.30	**4.84**	5.01	**4.84**
	50–60	8.33	7.97	8.03	7.65	7.88	**7.57**
PSNR↑	0–10	28.61	29.94	29.37	**34.15**	32.99	33.78
	10–20	25.92	26.77	26.13	**28.77**	28.47	28.49
	20–30	20.43	21.10	20.99	25.32	25.19	**25.49**
	30–40	18.96	19.73	19.30	23.03	22.94	**23.47**
	40–50	16.47	18.02	16.45	**21.17**	21.07	21.14
	50–60	14.33	14.57	13.69	18.43	**18.51**	18.44
$SSIM(10^{-1})\uparrow$	0–10	9.33	9.55	9.20	**9.75**	9.44	9.33
	10–20	8.94	9.08	8.94	**9.33**	9.19	9.11
	20–30	8.33	8.77	8.43	8.79	8.60	**8.96**
	30–40	7.99	8.15	7.94	8.22	8.11	**8.24**
	40–50	7.45	7.20	7.46	7.59	7.30	**7.61**
	50–60	6.00	6.25	6.33	6.70	6.37	**7.01**

Table 2. *Cont.*

Mask(%)		CA	Deepfillv2	PConv	CTSDG	AOT–GAN	PA–DeepFill
FID↓	0–10	4.25	3.89	3.91	**3.01**	3.14	3.11
	10–20	13.79	13.20	15.37	**8.89**	9.01	8.94
	20–30	22.38	22.05	27.37	17.09	15.79	**15.44**
	30–40	34.33	35.97	38.44	26.97	25.14	**24.75**
	40–50	53.66	50.19	58.47	**40.46**	44.37	43.74
	50–60	84.94	81.46	91.37	68.31	70.59	**64.58**

Table 3. Quantitative comparison on COCO.

Mask(%)		CA	Deepfillv2	PConv	CTSDG	AOT-GAN	PA-DeepFill
$L_1(10^{--2})\downarrow$	0–10	1.05	0.92	0.98	**0.54**	0.78	0.73
	10–20	1.35	1.29	1.30	**0.89**	1.14	0.91
	20–30	2.04	1.88	1.94	1.59	1.73	**1.43**
	30–40	2.99	2.71	2.85	2.37	2.35	**2.27**
	40–50	4.01	3.72	3.70	3.47	3.55	**3.36**
	50–60	6.58	6.35	6.44	6.13	**6.07**	6.19
PSNR↑	0–10	34.12	37.51	35.09	**39.48**	38.39	38.91
	10–20	31.32	33.05	32.74	34.15	33.60	**34.33**
	20–30	27.45	29.34	28.37	30.18	29.78	**30.43**
	30–40	24.27	26.35	25.91	27.04	26.73	**27.11**
	40–50	22.00	24.05	24.13	**24.67**	24.37	24.51
	50–60	18.97	20.12	19.87	**20.68**	20.50	20.44
$SSIM(10^{--1})\uparrow$	0–10	9.59	9.76	9.65	**9.84**	9.78	9.71
	10–20	9.40	9.54	9.55	**9.67**	9.57	9.57
	20–30	8.91	9.23	9.21	**9.31**	9.16	9.27
	30–40	8.51	8.84	8.80	8.92	8.88	**8.94**
	40–50	8.13	8.40	8.40	8.49	8.44	**8.64**
	50–60	7.31	7.60	7.41	**7.65**	7.55	**7.65**
FID↓	0–10	2.01	1.75	1.83	**1.21**	1.34	1.33
	10–20	4.16	3.87	4.01	**3.13**	3.37	**3.13**
	20–30	7.90	7.31	7.51	6.29	6.84	**6.22**
	30–40	13.96	11.37	13.37	10.01	10.75	**9.74**
	40–50	19.39	15.35	18.05	13.66	14.64	**13.43**
	50–60	31.33	24.25	29.54	**20.77**	22.67	20.74

Quality Analysis

To ensure a fair qualitative comparison, we randomly selected repaired images generated by different models and displayed them. In Figure 6, we present the results for various scenes, such as airplanes, fruits, and bridges, along with enlarged patches for closer examination. As depicted in these examples, our model demonstrates the ability to reconstruct more realistic structures and produce clearer textures for a variety of scenes.

(**a**) Mask Image

(**b**) The repair results of CA

(**c**) The repair results of Deepfillv2

(**d**) The repair results of PConv

(**e**) The repair results of CTSDG

(**f**) The repair results of AOT-GAN

(**g**) The repair results of PA-DeepFill

Figure 6. Qualitative comparisons of PA-DeepFill with CA [19], PConv [22], Deepfillv2 [20], CTSDG [38], and AOT-GAN [18] on COCO [26]. Each column shows the overall repair effect and local repair details. All the images are center-cropped and resized to 512 × 512.

4.3. Ablation Study

In this section, we conducted lots of ablation experiments to verify the effectiveness of the core components. We randomly select masks with different mask area ratios for each ablation experiment. As shown in Table 4, model performance presents significant decline after removing the pixel attention block (PAB) or the channel attention block (CAB).

Table 4. Ablation experiments on gathered attention block. Pixel attention block is denoted as PAB, and channel attention block is represented as CAB.

Model	PAB	CAB	$L_1(10^{-2})\downarrow$	PSNR↑	SSIM↑	FID↓
1	×	×	4.01	21.97	7.33	19.97
2	✓	×	3.59	22.74	7.93	21.78
3	×	✓	3.57	22.75	7.99	21.32
4(PA-DeepFill)	✓	✓	**3.23**	**23.47**	**8.24**	**24.75**

This is reasonable as these two types of attentions contribute to capturing different types of semantic correlations, and combining them together would provide comprehensive global information as complementary. In the qualitative analysis of Figures 7 and 8, it can be observed that the models incorporating attention blocks successfully synthesized higher-resolution textures in the missing regions of the images. This provides strong evidence for the effectiveness of the proposed attention blocks in image inpainting tasks.

(**a**) Mask Image (**b**) Repair Result without PAB (**c**) Repair Result with PAB

Figure 7. The qualitative analysis of Pixel Attention Block.

(**a**) Mask Image (**b**) Repair Result without CAB (**c**) Repair Result with CAB

Figure 8. The qualitative analysis of Channel Attention Block.

To evaluate the effectiveness of the proposed image enhancement (IE) module for image inpainting tasks, we conducted an ablation study on the module. Table 5 presents the quantitative comparison results of the image enhancement module. It can be observed that adding the image enhancement module between the rough inpainting network and the fine inpainting network significantly improves the performance of image inpainting.

Table 5. Quantitative analysis of the image enhancement module. Image enhancement module is denoted as IE module.

IE Module	$L_1(10^{-2})\downarrow$	PSNR↑	SSIM↑	FID↓
×	3.74	22.44	8.11	27.74
✓	**3.23**	**23.47**	**8.24**	**24.75**

In Table 5, it can be seen that adding the image enhancement module between the first-stage generator and the second-stage generator effectively enhances the image inpainting results.

(**a**) Mask Image (**b**) Repair Result without IE module (**c**) Repair Result with IE module

Figure 9. The qualitative analysis of image enhancement module.

In the qualitative analysis of Figure 9 , it can be observed that the models incorporating image enhancement module successfully synthesized higher-resolution textures in the missing regions of the images. We believe that the image enhancement module utilizes the structural priors of convolutional neural networks to capture structural information from images, effectively restoring images with complex textures and structures. It optimizes the output of the rough inpainting network and further utilizes this enhanced output as input for the fine inpainting network. This promotes the generation of inpainted images that are semantically consistent with the context, thereby enhancing the overall inpainting effect of the model.

To evaluate the effectiveness of the proposed improved discriminator for image inpainting tasks, we compared it with PatchGAN, SN-PatchGAN applied in Deepfillv2 [20], and MT-PatchGAN. The quantitative analysis results of the discriminator for the model are shown in Table 6.

Table 6. Quantitative analysis of different discriminators.

Model	$L_1(10^{-2})\downarrow$	PSNR↑	SSIM↑	FID↓
PatchGAN	3.67	22.13	8.07	41.44
SN-PatchGAN	3.34	23.07	**8.26**	33.47
MT-PatchGAN	**3.23**	**23.47**	8.24	**24.75**

From the data in Table 6, it can be observed that both our proposed improvement and SN-PatchGAN achieve better performance in image inpainting tasks compared to PatchGAN. This is because PatchGAN cannot perceive the real regions in the inpainted image and tends to classify all output image patches as fake. On the other hand, the other two improved discriminators have higher granularity and can effectively distinguish the real parts inherited from the original image from the inpainted parts generated by the model. Meanwhile, it can be concluded that MT-PatchGAN is more suitable for image

inpainting tasks compared to other approaches. We believe that this is because the Gaussian filtering used in this approach allows for the soft extraction of the masked regions, enabling pixel-level segmentation between the real and synthesized regions. This greatly reduces the obvious artifacts at the boundaries of the missing regions, caused by the presence of both real and synthesized pixels. Additionally, through multi-task guidance, the model can more accurately reconstruct the missing parts of the image, thus improving the overall image inpainting performance. Figure 10 presents a set of qualitative experiments comparing different discriminator approaches. It can be observed that the MT-PatchGAN is particularly suitable for image inpainting tasks. For example, PatchGAN can only use the surrounding pixels to fill in the gaps between the tiles. SN-PatchGAN produces some distortion when synthesizing the tile gaps, while MT-PatchGAN utilizes contextual information to complete the gaps between the tiles, generating a visually more appealing image.

(**a**) Mask Image (**b**) PatchGAN (**c**) SN-PatchGAN (**d**) MT-PatchGAN

Figure 10. The qualitative analysis of image enhancement module.

To further improve the quality of image inpainting, we conducted comparative experiments on the Multi-Task Guided Mask-Level Local Discriminator with different sizes of receptive fields (default size is 70×70). The experimental results are shown in Table 7. From the experimental results, it can be observed that using the default size of 70×70 for the receptive field yields the best performance in image inpainting tasks.

Table 7. Comparative experiment on the size of receptive field.

Size	$L_1(10^{-2})\downarrow$	PSNR↑	SSIM↑	FID↓
16×16	3.39	23.00	8.23	27.71
30×30	3.39	23.01	**8.24**	27.52
70×70	**3.23**	**23.47**	**8.24**	**24.75**
128×128	3.41	22.64	8.21	29.69

Figure 11 presents a qualitative analysis of different receptive field sizes. As shown in Figure 11d, using a receptive field size of 70×70 effectively synthesizes the missing pixels in the airplane tire and demonstrates better performance in preserving the obscured airplane body label compared to other sizes of receptive fields. The size of the receptive field has a significant impact on the performance of image inpainting tasks. If the receptive field is too large, the discriminator will focus more on global structure and overall image consistency. In image inpainting tasks, an excessively large receptive field may cause the discriminator to overlook the consistency of local textures, details, and edges. Consequently, the generated image may exhibit blurred details and textures, resulting in a lower quality restoration. On the other hand, if the receptive field is too small, the discriminator will primarily focus on the local textures and details of the input image. This weakens the emphasis on global structure and consistency, which in turn affects the generation of restoration results with correct global structure. In such cases, although the restoration results may have high-quality textures and details in local regions, there may be issues with the overall structure, such as object deformation or unclear edges. Therefore, finding an optimal receptive field size is crucial to balance the preservation of details and the overall structure in image inpainting tasks.

(**a**) Mask Image (**b**) 16 × 16 (**c**) 32 × 32 (**d**) 70 × 70 (**e**) 128 × 128

Figure 11. Qualitative analysis of receptive field size.

4.4. Application

In this section, we will evaluate the performance of our model, PA-DeepFill, on three real-world scenarios: face editing, logo removal, and object removal. Through these experiments, we aim to demonstrate the effectiveness of PA-DeepFill in practical applications.

4.4.1. Logo Removal

The automatic removal of logos is highly beneficial in logo design, media content creation, and other relevant fields. To evaluate the effectiveness of PA-DeepFill in logo removal, we employ the QMUL-OpenLogo dataset for both training and testing [49]. Figure 12 showcases the visual outcomes of logo removal achieved using our proposed PA-DeepFill. The results effectively demonstrate that PA-DeepFill is capable of removing logos from images and seamlessly filling the resulting gaps with realistic content.

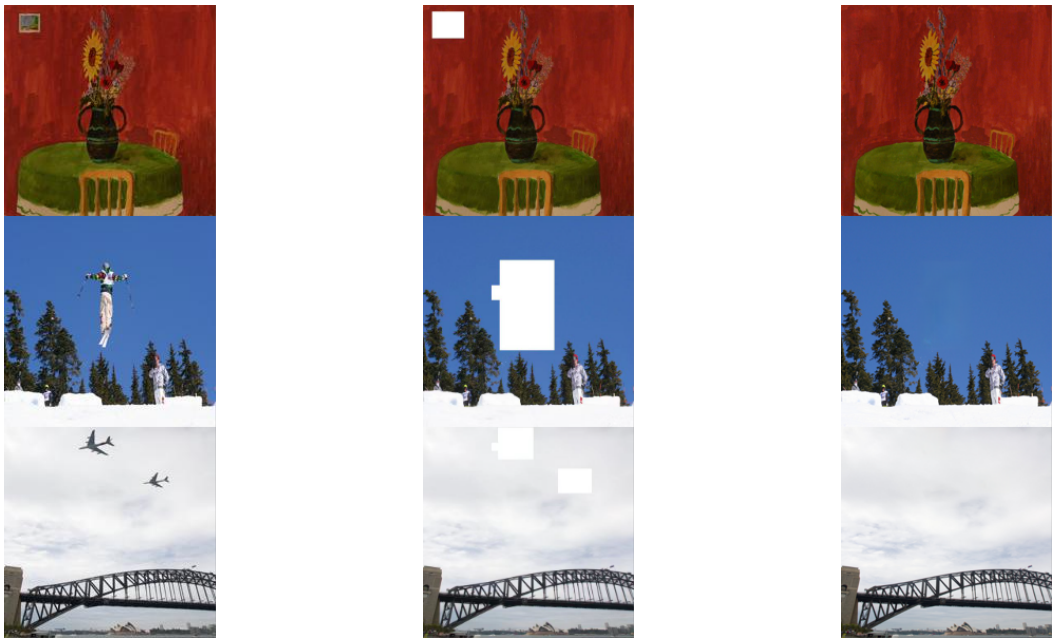
(**a**) Original Images (**b**) Masked Images (**c**) Repaired Images

Figure 12. Visual results of PA-DeepFill in logo removal.

4.4.2. Object Removal

Object removal is a widely used technique for image anonymization, with the aim of removing specific objects from an image and filling the resulting gaps with plausible content. To assess the performance of PA-DeepFill in object removal, we train the model on the COCO dataset [26] using user-defined masks.

Figure 13 displays the results of object removal achieved by our proposed PA-DeepFill. The visual outcomes demonstrate that PA-DeepFill is capable of effectively removing specific objects from diverse and complex scenes using user-provided masks.



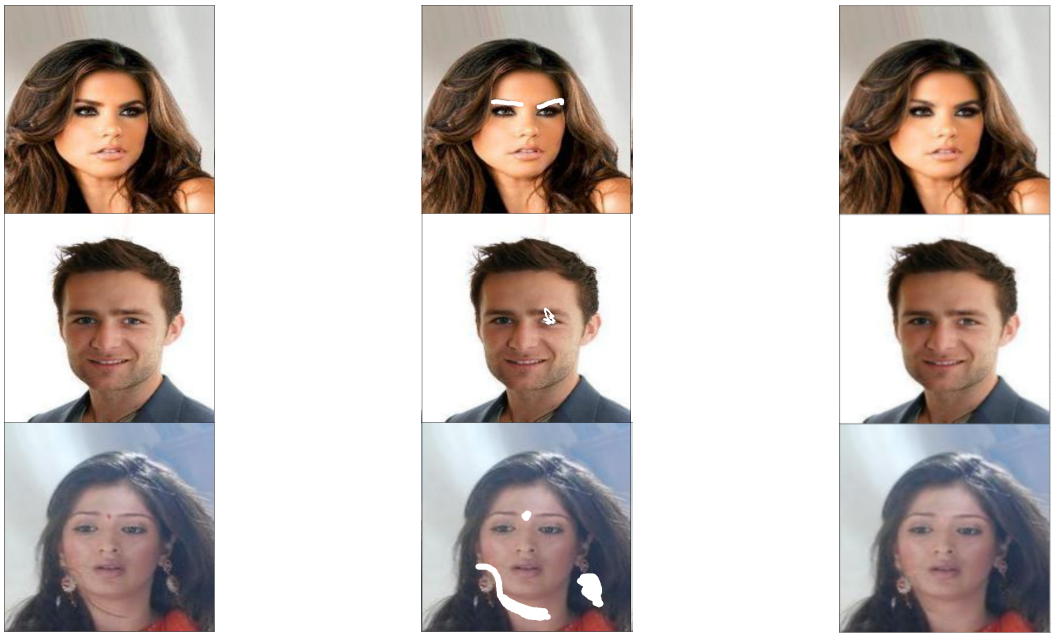
(**a**) Original Images (**b**) Masked Images (**c**) Repaired Images

Figure 13. Visual results of PA-DeepFill in object editing.

4.4.3. Face Editing

We test the effect of face editing on CelebA [25]. The visual results of the proposed PA-DeepFill face editing are shown in Figure 14.

The results show that our model can complete the consistent structure and generate clear facial texture.



(**a**) Original Images (**b**) Masked Images (**c**) Repaired Images

Figure 14. Visual results of PA-DeepFill in face editing.

5. Conclusions

In this paper, we introduce an image inpainting model that leverages the concept of progressive inpainting and incorporates a gathered attention block for high-resolution image inpainting. Our proposal is capable of adaptively incorporating the distant visual features as complementary to boost the infilling performance, and progressively generating visual content with different resolutions to avoid the distorted visual features. Our proposal demonstrates superior performance on different datasets and tasks.

Author Contributions: Conceptualization, M.C. and H.J.; methodology M.C.; software, H.J. and C.L.; validation, M.C.; formal analysis, M.C., H.J. and C.L.; investigation, C.L.; resources, C.L.; data curation, C.L.; writing—original draft preparation, M.C.; writing—review and editing, C.L. and M.C.; visualization, M.C. and H.J.; supervision, C.L.; project administration, C.L.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All data are presented in the main text.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wan, Z.; Zhang, B.; Chen, D.; Zhang, P.; Chen, D.; Liao, J.; Wen, F. Bringing old photos back to. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2747–2757.
2. Youngjoo, J.; Jongyoul, P. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1745–1753.
3. Lin, S.; Wang, X.; Xiao, G.; Yan, Y.; Wang, H. Hierarchical representation via message propagation for robust model fitting. *IEEE Trans. Ind. Electron.* **2021**, *68*, 8582–8592. [CrossRef]
4. Lin, S.; Xiao, G.; Yan, Y.; Suter, D.; Wang, H. Hypergraph optimization for multi-structural geometric model fitting. *Proc. Aaai Conf. Artif. Intell.* **2019**, *33*, 8730–8737. [CrossRef]
5. Lin, S.; Luo, H.; Yan, Y.; Xiao, G.; Wang, H. Co-clustering on bipartite graphs for robust model fitting. *IEEE Trans. Image Process.* **2022**, *31*, 6605–6620. [CrossRef]
6. Bertalmio, M.S.; Caselles, C.; Coloma, V.B. Image inpainting. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 23–28 July 2000; pp. 417–424.
7. Shen, J.; Chan, T.F. Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **2002**, *62*, 1019–1043. [CrossRef]
8. Sridevi, G.; Srinivas Kumar, S. Image inpainting based on fractional-order nonlinear diffusion for image reconstruction. *Circuits Syst. Signal Process.* **2019**, *38*, 3802–3817. [CrossRef]
9. Criminisi, A.P.; Kentaro, P.T. Object removal by exemplar-based inpainting. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 18–20 June 2003; IEEE: New York, NY, USA, 2003; Volume 2, p. II.
10. Cheng, W.-H.; Hsieh, C.W.; Lin, S.-K.; Wang, C.-W.; Wu, J.-L. Robust algorithm for exemplar-based image inpainting. In Proceedings of the International Conference on Computer Graphics, Imaging and Visualization, Beijing, China, 26–29 July 2005; pp. 64–69.
11. Xu, Z.; Sun, J. Image inpainting by patch propagation using patch sparsity. *IEEE Trans. Image Process.* **2010**, *19*, 1153–1165. [PubMed]
12. Le Meur, O.; Gautier, J.; Guillemot, C. Examplar-based inpainting based on local geometry. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–13 September 2011; IEEE: New York, NY, USA, 2011; pp. 3401–3404.
13. Yan, Z.; Li, X.; Li, M.; Zuo, W.; Shan, S. Shift-net: Image inpainting via deep feature rearrangement. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 1–17.
14. Lin, S.; Yang, A.; Lai, T.; Weng, J.; Wang, H. Multi-motion Segmentation via Co-attention-induced Heterogeneous Model Fitting. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *2013*, 1–13. [CrossRef]
15. Rawat, W.; Wang, Z. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **2017**, *29*, 2352–2449. [CrossRef]
16. Hao, Y.; Shuyuan, L.; Lin, C.; Yang, L.; Hanzi, W.D.; Zhang, P.; Chen, D.; Liao, J.; Wen, F. SCINet: Semantic cue infusion network for lane detection. *Proc. IEEE Int. Conf. Image Process.* **2022**, *2022*, 1811–1815.
17. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
18. Zeng, Y.; Fu, J.; Chao, H.; Guo, B. Aggregated contextual transformations for high-resolution image inpainting. *IEEE Trans. Vis. Comput. Graph.* **2022**, *29*, 3266–3280. [CrossRef] [PubMed]

19. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Generative image inpainting with contextual attention. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* **2018**, *2018*, 5505–5514.
20. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T.S. Free-form image inpainting with gated convolution. *Proc. IEEE/CVF Int. Conf. Comput. Vis.* **2019**, *2019*, 4471–4480.
21. Romero, A.; Castillo, A.; Abril-Nova, J.; Timofte, R.; Das, R.; Hira, S.; Pan, Z.; Zhang, M.; Li, B.; He, D.; et al. NTIRE 2022 image inpainting challenge: Report. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–22 June 2022; pp. 1150–1182.
22. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.-C.; Tao, A.; Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 85–100.
23. Wang, Y.; Li, C.; Liu, Z.; Li, M.; Tang, J.; Xie, X.; Chen, L.; Yu, P.S. An Adaptive Graph Pre-training Framework for Localized Collaborative Filtering. *ACM Trans. Inf. Syst.* **2022**, *41*, 1–27. [CrossRef]
24. Liu, H.; Jiang, B.; Xiao, Y.; Yang, C. Coherent semantic attention for image inpainting. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4170–4179.
25. Liu, Z.; Luo, P.; Wang, X.; Tang, X. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3730–3738.
26. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
27. Wang, X.; Niu, S.; Wang, H. Image inpainting detection based on multi-task deep learning network. *IETE Tech. Rev.* **2021**, *38*, 149–157. [CrossRef]
28. Hays, J.; Efros, A.A. Scene completion using millions of photographs. *Commun. ACM* **2008**, *51*, 87–94. [CrossRef]
29. Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A.A. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2536–2544.
30. Iizuka, S.; Simo-Serra, E.; Ishikawa, H. Globally and locally consistent image completion. *ACM Trans. Graph. ToG* **2017**, *36*, 1–14. [CrossRef]
31. Yeh, R.A.; Chen, C.; Yian, L.T.; Schwing, A.G.; Hasegawa-Johnson, M.; Do, M.N. Semantic image inpainting with deep generative models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, 21–26 June 2017; pp. 5485–5493.
32. Li, Y.; Liu, S.; Yang, J.; Yang, M.-H. Generative face completion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3911–3919.
33. Mnih, V.; Heess, N.; Graves, A. Recurrent models of visual attention. *Adv. Neural Inf. Process. Syst.* **2014**, *36*, 27.
34. Pang, B.; Li, C.; Liu, Y.; Lian, J.; Zhao, J.; Sun, H.; Deng, W.; Xie, X.; Zhang, Q. Improving Relevance Modeling via Heterogeneous Behavior Graph Learning in Bing Ads. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 3713–3721.
35. Zhang, X.; Wang, X.; Shi, C.; Yan, Z.; Li, X.; Kong, B.; Lyu, S.; Zhu, B.; Lv, J.; Yin, Y.; et al. De-gan: Domain embedded gan for high quality face image inpainting. *Pattern Recognit.* **2022**, *124*, 108415. [CrossRef]
36. Zhou, Y.; Barnes, C.; Shechtman, E.; Amirghodsi, S. Transfill: Reference-guided image inpainting by merging multiple color and spatial transformations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2266–2276.
37. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 June 2017; pp. 1125–1134.
38. Guo, X.; Yang, H.; Huang, D. Image inpainting via conditional texture and structure dual generation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 14134–14143.
39. Nazeri, K.; Ng, E.; Joseph, T.; Qureshi, F.Z.; Ebrahimi, M. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv* **2019**, arXiv:1901.00212.
40. Ren, Y.; Yu, X.; Zhang, R.; Li, T.H.; Liu, S.; Li, G. Structureflow: Image inpainting via structure-aware appearance flow. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 181–190.
41. Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; Zhang, L. Image-Adaptive YOLO for Object Detection in Adverse Weather Conditions. *Proc. Aaai Conf. Artif. Intell.* **2022**, *36*, 1792–1800. [CrossRef]
42. Hu, Y.; He, H.; Xu, C.; Wang, B.; Lin, S. Exposure: A White-Box Photo Post-Processing Framework. *ACM Trans. Graph.* **2018**, *37*, 26.1–26.17. [CrossRef]
43. Xu, Y.; Feng, K.; Yan, X.; Yan, R.; Ni, Q.; Sun, B.; Lei, Z.; Zhang, Y.; Liu, Z. CFCNN: A novel convolutional fusion framework for collaborative fault identification of rotating machinery. *Inf. Fusion* **2023**, *95*, 1–16. [CrossRef]
44. Polesel, A.; Mathews, V.; Ramponi, G. Image enhancement via adaptive unsharp masking. *IEEE Trans. Image Process.* **2000**, *3*, 9. [CrossRef]
45. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 2414–2423.

46. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 694–711.
47. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
48. Xu, Y.; Yan, X.; Sun, B.; Zhai, J.; Liu, Z. Multireceptive Field Denoising Residual Convolutional Networks for Fault Diagnosis. *IEEE Trans. Ind. Electron.* **2022**, *69*, 11686–11696. [CrossRef]
49. Su, H.; Zhu, X.; Gong, S. Open logo detection challenge. *arXiv* **2018**, arXiv:1807.01964.