2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 978-1-6654-6946-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/CVPR52688.2022.00191

# MISF:Multi-level Interactive Siamese Filtering for High-Fidelity Image Inpainting

Xiaoguang Li[1,*], Qing Guo[2*], Di Lin[3], Ping Li[4],
Wei Feng[3†], Song Wang[1†]

[1]University of South Carolina, USA, [2]Nanyang Technological University, Singapore, [3]College of Intelligence and Computing, Tianjin University, China, [4]The Hong Kong Polytechnic University, Hong Kong

## Abstract

*Although achieving significant progress, existing deep generative inpainting methods still show low generalization across different scenes. As a result, the generated images usually contain artifacts or the filled pixels differ greatly from the ground truth, making them far from real-world applications. Image-level predictive filtering is a widely used restoration technique by predicting suitable kernels adaptively according to different input scenes. Inspired by this inherent advantage, we explore the possibility of addressing image inpainting as a filtering task. To this end, we first study the advantages and challenges of the image-level predictive filtering for inpainting: the method can preserve local structures and avoid artifacts but fails to fill large missing areas. Then, we propose the semantic filtering by conducting filtering on deep feature level, which fills the missing semantic information but fails to recover the details. To address the issues while adopting the respective advantages, we propose a novel filtering technique,* i.e.*, Multi-level Interactive Siamese Filtering (MISF) containing two branches: kernel prediction branch (KPB) and semantic & image filtering branch (SIFB). These two branches are interactively linked: SIFB provides multi-level features for KPB while KPB predicts dynamic kernels for SIFB. As a result, the final method takes the advantage of effective semantic & image-level filling for high-fidelity inpainting. Moreover, we discuss the relationship between MISF and the naive encoder-decoder-based inpainting, inferring that MISF provides novel dynamic convolutional operations to enhance the high generalization capability across scenes. We validate our method on three challenging datasets,* i.e.*, Dunhuang, Places2, and CelebA. Our method outperforms state-of-the-art baselines on four metrics,* i.e.*, $L_1$, PSNR, SSIM, and LPIPS.*

*Xiaoguang Li and Qing Guo are co-first authors and contribute equally.
†Wei Feng (wfeng@ieee.org) and Song Wang (songwang@cec.sc.edu) are the corresponding authors. Please try the released code and model in `https://github.com/tsingqguo/misf`.

## 1. Introduction

Image inpainting is a fundamental problem in computer vision and artificial intelligence applications. The main goal is to fill missing pixels in an image and make it identical to the clean one. Recent works mainly address the task by modeling it as a generation task [17, 19, 24, 30]. As a result, they can employ cutting-edge deep generative techniques (*e.g.*, generative adversarial network [10, 22]) to realize high-quality restoration on challenging datasets. However, the generative network-based inpainting encodes the input image to a latent space and then decodes it to a new image. Such a process neglects the explicit prior, *i.e.*, smoothness across neighboring pixels or features, and the fidelity of inpainting fully relies on the data and training strategy.

Note that, different from the generation task, image inpainting has its specific challenges: *First*, the image inpainting requires the completed images to respect the clean image (*i.e.*, to produce high-fidelity images) and to be natural. These requirements make image inpainting different from the pure image generation task that mainly focuses on naturalness. *Second*, the missing areas' shapes may be different and the background scenes are diverse. These facts require the inpainting method to have high generalization capability across missing areas and scenes. Although deep generative networks achieve significant progress on image inpainting, they are far from solving the above challenges. For example, the recent work RFRNet [17] conducts feature reasoning on the encoder-decoder network and achieves state-of-the-art performance on public datasets. Nevertheless, given different faces with different missing areas, it is hard to produce high-fidelity inpainting results. Moreover, the artifacts appear in the results. As shown in Fig. 1, for the top-left example with small missing areas, RFRNet can generate a natural face. However, when comparing with the ground truth, we see that the local structures around the arrows are distorted. For the bottom-left example with larger missing areas, RFRNet even fails to produce a natural face. When handling other natural scenes (*e.g.*, two examples on the

978-1-6654-6946-3/22/$31.00 ©2022 IEEE
DOI 10.1109/CVPR52688.2022.00191

**Figure 1.** Four examples of using state-of-the-art methods (*i.e.*, RFRNet [17] and JPGNet [12]) and the proposed method for image inpainting. Our method is able to complete the missing pixels and produce realistic and high-fidelity images. We highlight the main differences via green arrows.

right), RFRNet also introduces small artifacts.

Guo *et al*. [12] have noticed above issues of generative-based inpainting methods [12,17,23,25] and propose JPGNet that uses image-level predictive filtering to alleviate the artifacts. The image-level predictive filtering reconstructs pixels via their neighboring pixels. The filtering kernels are adaptively estimated according to the inputs. As a result, JPGNet can recover the local structure while avoiding the artifacts, thus helping RFRNet achieve significant quality improvement. Nevertheless, many details are smoothed, while the real structures fail to be recovered (See Fig. 1).

Inspired by the inherent advantages of predictive filtering on adaptiveness and restoration, we propose a novel framework to handle the two challenges. Specifically, we make three main efforts: *First*, we study the advantages and challenges of adopting the existing predictive filtering method for image inpainting, that is, the image-level predictive filtering can restore local structures and avoid artifacts but cannot fill large missing areas. *Second*, we extend the image-level filtering to the deep feature level and propose the *semantic filtering*, which can complete large missing areas but loses details. *Third*, to address the issues, we propose a novel filtering technique, *i.e.*, *Multi-level Interactive Siamese Filtering (MISF)*, which contains two branches: kernel prediction branch (KPB) and semantic & image filtering branch (SIFB). These two branches are interactively linked at semantic & pixel levels. SIFB provides multi-level features for KPB while KPB predicts dynamic kernels for SIFB. MISF can utilize the smoothness prior across neighbors explicitly and reconstruct clean pixels or features by linearly combining the neighbors. As a result, the final method takes the advantage of effective semantic & pixel-level filling for high-fidelity inpainting. As shown in Fig. 1, our method can generate natural and high-fidelity images under different scenes with different missing areas. In addition, we conduct an insightful discussion about the relationship between our method and the naive generative network, inferring that our method corresponds multi-level *dynamic convolutional operations* that adjusts the convolutional parameters according to different inputs and brings in generalization. We conduct extensive experiments on three challenging datasets (*i.e.*, Place2, CelebA, and Dunhuang) and achieve much better scores than the competitive methods on the public datasets in terms of four quality metrics.

## 2. Related Work

**Deep generative networks for image inpainting.** The conventional image inpainting methods [2, 3, 5, 7, 8, 18, 27] focus on finding useful patches for recovering the damaged image regions. However, the semantic information of image regions is out of consideration in these methods, thus yielding unsatisfactory results in complex scenes.

More recent methods employ deep generative adversarial networks [10] to learn semantic information from data for better inpainting. Pathak et al. [24] and Iizuka et al. [15] use the conditional GAN [22], along with the powerful information encoder, for a better polish of the image details in the inpainting result. Moreover, Iizuka et al. [15] enforce the local and global consistency of the recovered regions. Li et al. [17] propose the recurrent reconstruction of the corrupted image on the convolutional feature of an image. In addition, Yan et al. [28] and Yu et al. [31] propose the contextual model for capturing the correlation between the long-range regions. Liu et al. [19] and Yu et al. [32] focus on repairing the irregular shape of image damage, by capturing the spatial deformations of the damaged regions. In addition to the semantic information, some works employ the geometric information like the edge and contour of image regions for effective image inpainting [23, 25]. However, the above methods generally formulate the inpainting task as the generation. Although the generated images look natural and realistic, they are less identical to the ground truth.

**Predictive filtering based image restoration.** The predictive filtering has been widely used in image restoration tasks, *e.g.*, denoising [1, 21], deraining [13], shadow removing [9], and blur synthesis [4]. The predictive filtering allows more focused learning of the surrounding information for each pixel. However, the image-level filtering can hardly address the tasks relying on the semantic understanding of the scene. Different from previous methods, for the first

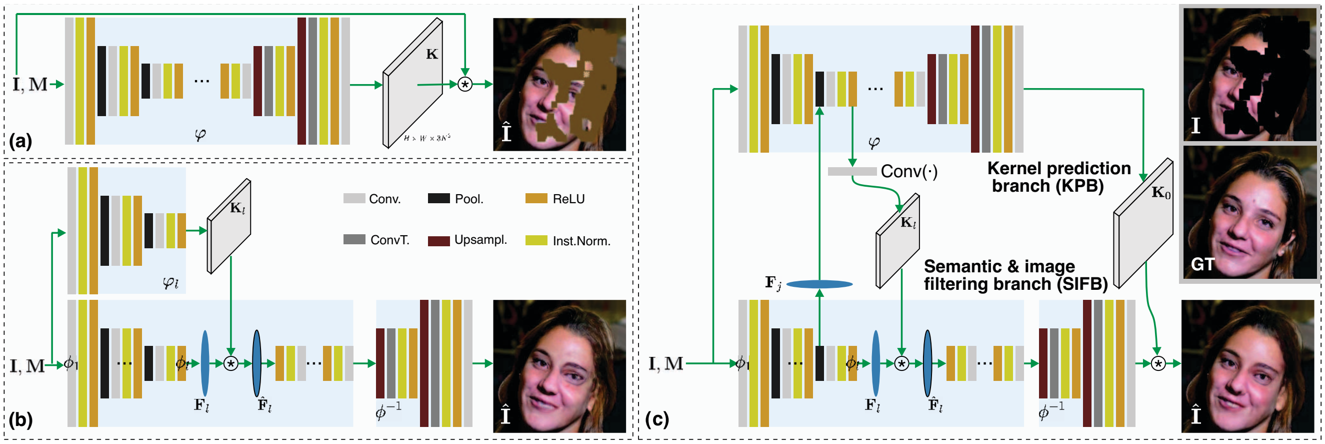


**Figure 2.** Three filtering-based image inpainting methods. (a) represents the predictive image-level filtering introduced in Sec. 3.1. (b) shows the proposed semantic filtering-based inpainting in Sec. 4.1. (c) is the multi-level interactive siamese filtering(MISF) in Sec. 4.2.

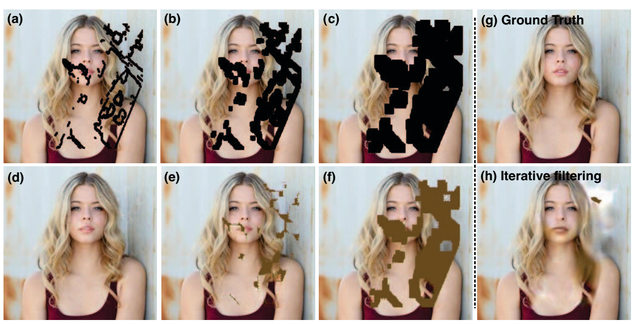


**Figure 3.** Using predictive filtering for image inpainting under three mask sizes. The images (a), (b), and (c) are fed to the predictive filtering and we get (d), (e), and (f), respectively. We also try to complete the image (c) via the predictive filtering recurrently and obtain (h).

attempt, we propose a novel filtering method by extending the image-level filtering to the deep feature level. As a result, we can conduct semantic filtering and realize effective inpainting. Moreover, to take both advantages of image-level and deep feature-level filtering, we propose the multi-level semantic & image filtering, which completes the semantic information as well as rich details.

## 3. Discussion and Motivation

### 3.1. Predictive Filtering for Image Inpainting

Predictive filtering is a widely used image restoration technique and can address image denoising [21] and deraining [13] tasks. Here, we formulate the image inpainting as the pixel-wise predictive filtering task

$$\hat{\mathbf{I}} = \mathbf{I} \circledast \mathbf{K}, \tag{1}$$

where $\mathbf{I} \in \mathbb{R}^{H\times W}$ is the corrupted image and $\hat{\mathbf{I}} \in \mathbb{R}^{H\times W}$ is the completed counterpart. The tensor $\mathbf{K} \in \mathbb{R}^{H\times W\times N^2}$ contains $HW$ kernels for filtering all pixels. The operation '$\circledast$' denotes the pixel-wise filtering. We can expand the above equation as

$$\hat{\mathbf{I}}[\mathbf{p}] = \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} \mathbf{K}_\mathbf{p}[\mathbf{q}-\mathbf{p}]\mathbf{I}[\mathbf{q}]. \tag{2}$$

Here, $\mathbf{p}$ and $\mathbf{q}$ are the coordinates of pixels in the image while the set $\mathcal{N}_\mathbf{p}$ contains $N^2$ neighboring pixels of $\mathbf{p}$. The matrix $\mathbf{K}_\mathbf{p} \in \mathbb{R}^{N\times N}$ is the reshaped $\mathbf{p}$th vector of $\mathbf{K}$ and determines the weights for all pixels in $\mathcal{N}_\mathbf{p}$, which is also known as the kernel for the pixel $\mathbf{p}$. Intuitively, the filtering is to reconstruct the pixel $\mathbf{p}$ by linearly combining its neighboring pixels. For image inpainting, the pixels at the boundary of missing areas are reasoned by their neighboring pixels. The principle is that the missing pixels do not break the local structure. Meanwhile, the related pixels can be used to reconstruct the missing pixels. However, the local structures around missing pixels are diverse and may distinguish them from each other. To adapt to the context variations, we can train a predictive network to estimate the kernels for all pixels according to the input image and missing areas

$$\mathbf{K} = \varphi(\mathbf{I}, \mathbf{M}), \tag{3}$$

where $\varphi(\cdot)$ is the predictive network and $\mathbf{M}$ is the binary mask indicating the missing areas. We set $\varphi(\cdot)$ as an encoder-decoder network (See Fig. 2 (a)) and train it via the image quality loss (*i.e.*, $L_1$) [13], GAN loss [23], Style loss [26], and the perceptual loss [16]. We will detail these loss functions in Sec. 4.4. The pipeline is shown in Fig. 2 (a).

### 3.2. Challenges and Motivations

The above predictive filtering for inpainting is non-trivial and needs to be carefully studied. We can train the function $\varphi(\cdot)$ with an image inpainting dataset like CelebA dataset [20]. Then, we evaluate it on a series of images with the missing areas becoming larger and thicker. We show an example in Fig. 3 and observe that: ❶ For the thin and small missing areas, the predictive filtering can complete the


**Figure 4.** Two examples of the encoder-decoder network (En-Decoder), semantic filtering (Sem-Filter), and multi-level interactive siamese filtering (MISF). We highlight the main differences via green arrows.

missing pixels effectively and lead to high-fidelity results (See Fig. 3 (a) and (d)). Nevertheless, when the missing areas become larger and thicker, the pixels away from the missing areas' boundaries cannot be recovered. This is because the large missing areas break the local structure. Thus, the image-level filtering cannot achieve the reconstruction goal anymore.❷ Different scenes require the predicted kernels to adapt to the semantic variations. Nevertheless, the image-level filtering can only reconstruct pixels according to their local contexts and cannot understand the whole scene. For example, when the missing area is significantly large (See Fig. 3 (c)), the image-level filtering cannot guess what pixels should be filled to make the face realistic with high fidelity.

A naive solution for the challenges is to conduct the filtering recurrently. Specifically, we can perform filtering on the inpainting result, again and again, that is, we use the estimated missing pixels to reconstruct the pixels inside the missing areas. We show the result of such a strategy in Fig. 3 (h) for inpainting (c). The completed pixels become vague around the center of the missing areas. It is mainly because the large missing areas break the local structure. Thus, only the pixels near the boundary are reconstructed but have low fidelity. The reconstruction errors are accumulated during the recurrent filtering process. Recently, Guo *et al.* [12] incorporate the predictive filtering and generative network to address this issue. However, such a solution can equitably introduce some artifacts of the state-of-the-art generative network-based method. As a result, a novel technique is necessary to address the challenges.

# 4. Methodology

## 4.1. Semantic Filtering for Image Inpainting

As explained in the Sec. 3.2, the image filtering-based inpainting is not that effective since the large missing areas break the local structure information that lays the foundation of filtering-based restoration. To address this issue, we propose to extend the filtering from the image level to the deep feature level that contains semantic information. The intuitive idea is that semantic information can be preserved even a large area of the image is lost. As the case in Fig. 3 (c), even though the large areas of the girl's face are missed, a human can fill the missing regions according to the understanding of the face. To achieve the semantic filtering, we first employ an encoder-decoder network where the encoder is to extract features from the corrupted image (*i.e.*, $\mathbf{I}$) and the decoder is to map the features to the completed image. We have the following formulation for the encoder

$$\mathbf{F}_L = \phi(\mathbf{I}, \mathbf{M}) = \phi_L(\ldots \phi_l(\ldots \phi_2(\phi_1(\mathbf{I}, \mathbf{M})))) \tag{4}$$

where $\phi(\cdot)$ is the encoder and $\mathbf{F}_l$ is the deep feature extracted from the $l$th layer, *i.e.*, $\mathbf{F}_l = \phi_l(\mathbf{F}_{l-1})$. For example, $\mathbf{F}_L$ is the output of the last layer of $\phi(\cdot)$ (*i.e.*, $\phi_L(\cdot)$). The decoder can be formulated as

$$\hat{\mathbf{I}} = \phi^{-1}(\mathbf{F}_L), \tag{5}$$

where $\phi^{-1}(\cdot)$ is the decoder. Then, we conduct the semantic filtering on extracted features like the image-level filtering

$$\hat{\mathbf{F}}_l[\mathbf{p}] = \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} \mathbf{K}_{l,\mathbf{p}}[\mathbf{q}-\mathbf{p}]\mathbf{F}_l[\mathbf{q}], \tag{6}$$

where $\mathbf{K}_{l,\mathbf{p}}$ is the kernel for filtering the $\mathbf{p}$th element of $\mathbf{F}_l$ via the neighboring elements, *i.e.*, $\mathcal{N}_\mathbf{p}$. We use the matrix $\mathbf{K}_l$ to include all element-wise kernels (*i.e.*, $\mathbf{K}_{l,\mathbf{p}}$). After that, we replace the $\mathbf{F}_l$ with $\hat{\mathbf{F}}_l$ in Eq. (4) and conduct the subsequent operations. To let the kernels adapt to different scenes, we also employ a predictive network to predict the kernels like the image-level predictive filtering (*i.e.*, Eq. (3))

$$\mathbf{K}_l = \varphi_l(\mathbf{I}, \mathbf{M}), \tag{7}$$

where $\varphi_l(\cdot)$ is the predictive network to produce $\mathbf{K}_l$. We can conduct the semantic filtering on any deep features. In the following, we use a 3-layer convolution network for $\varphi_l(\cdot)$, and only perform semantic filtering at the 3th layer of $\phi(\cdot)$ for an intuitive discussion.

We show the semantic filtering-based image inpainting in Fig. 2 (b) and train the networks (*i.e.*, $\phi(\cdot)$, $\phi^{-1}(\cdot)$, and $\varphi_l(\cdot)$) via the $L_1$, GAN, Style and perceptual loss functions like the predictive filtering. We present the inpaiting examples in Fig. 2 and Fig. 4 and have the following observations: ❶ Compared with the image-level predictive filtering, semantic filtering can fill all missing pixels and recover the semantic information effectively. As the cases in Fig. 4, the structures of the missed left eyes and faces are recovered. As a result, the inpainting results are more realistic and have higher fidelity. ❷ Although the main structures are recovered, the results lose details. In the first case of Fig. 4, the girl's forehead and left eye still contain artifacts and her mouth is blurred. We have similar observations on other cases.

## 4.2. Multi-level Interactive Siamese Filtering

Semantic filtering fills the missing semantic information at the deep feature level that has a low spatial resolution.

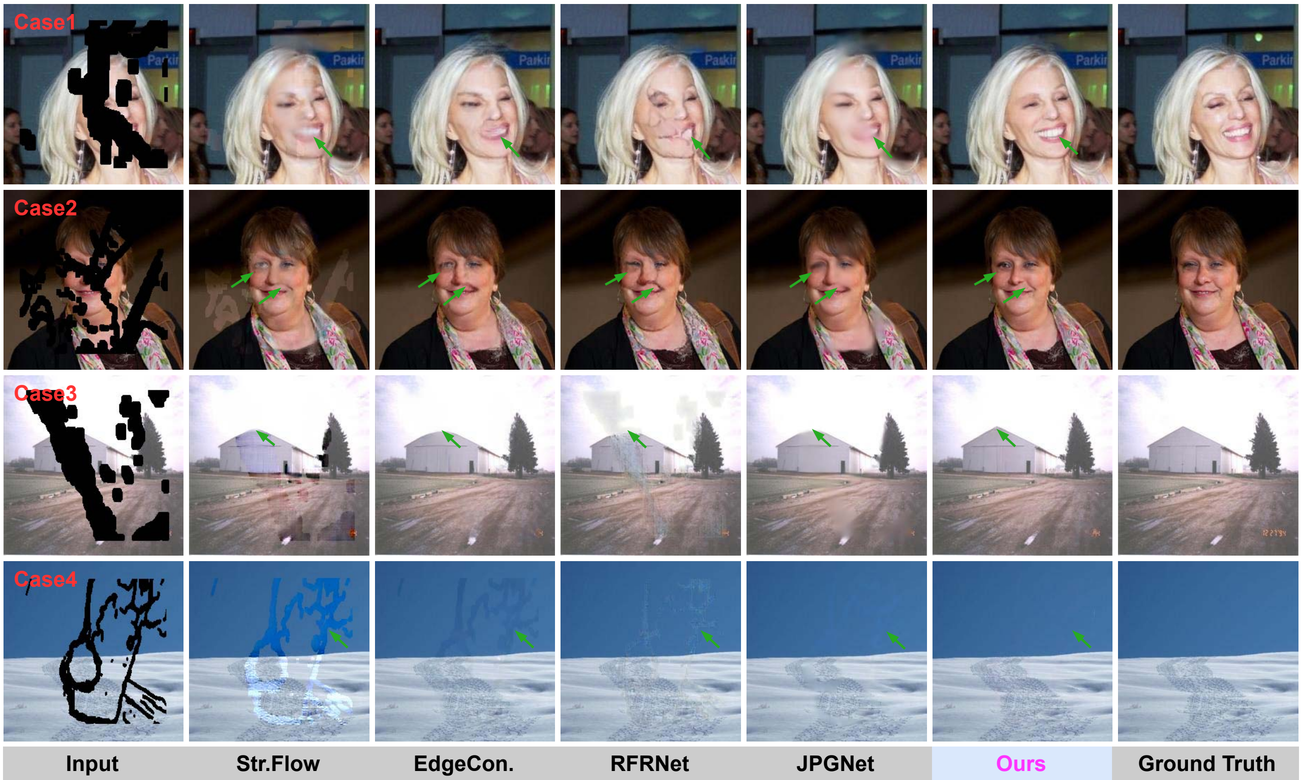


**Figure 5.** Four visualization results of Str.Flow [25], EdgeCon. [23], RFRNet [17], JPGNet [12], and our method. Case1 and Case2 are from the CelebA dataset, Case3 and Case4 are from Places2 dataset. We highlight the main differences via green arrows.

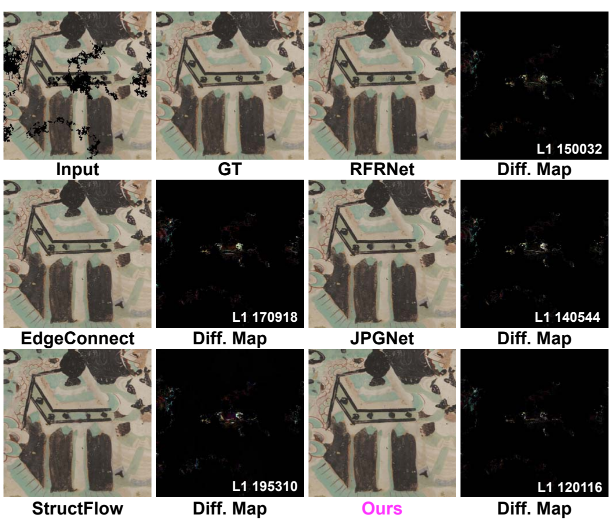


**Figure 6.** An example from the Dunhuang dataset. In addition to the inpainting results of five methods, we show the difference maps and L1 norm between the predicted results and the ground truth (GT).

Thus, it inevitably loses detailed information. A straightforward solution is to conduct filtering on multi-level features. For example, for all features extracted from the encoder (*i.e.*, $\{\mathbf{F}\}_{l=1}^{L}$), we can filter each of them via an exclusive predictive network as in Sec. 4.1, which, however, would lead to extra memory and time costs. Moreover, the $l$th predicted kernel (*i.e.*, $\mathbf{K}_l$) only depends on the $l$th feature (*i.e.*, $\mathbf{F}_l$) and cannot take the advantages of the features from other layers. To address this issue, we propose the *multi-level interactive Siamese filtering (MISF)* that consists of two branches with similar architectures, *i.e.*, *kernel prediction branch (KPB)* and *semantic & image filtering branch (SIFB)*, which are encoder-decoder networks containing several convolutional blocks. The two branches are interactively linked: KPB (*i.e.*, $\varphi(\cdot)$ in Fig. 2 (c)) takes the raw image, binary mask, and multiple features of SIFB as inputs and predicts multi-level kernels for SIFB. SIFB (*i.e.*, $\phi(\cdot)$ in Fig. 2 (c)) uses these kernels to filter the features at different levels. As a result, SIFB are dynamically changed according to the input. We show the whole framework in Fig. 2 (c).

Specifically, given a corrupted image $\mathbf{I}$ and the corresponding binary mask $\mathbf{M}$, we feed them to the SIFB that conducts filtering at the image level and semantic level (*i.e.*, filtering at the $l$th-layer feature) jointly. As a result, we can generate the completed image by

$$\hat{\mathbf{I}} = \phi^{-1}(\phi_L(\ldots\phi_{l+1}(\mathbf{F}_l \circledast \mathbf{K}_l))) \circledast \mathbf{K}_0, \tag{8}$$

where $\mathbf{F}_l = \phi_l(\ldots\phi_1(\mathbf{I}, \mathbf{M}))$. The kernels for deep feature and image (*i.e.*, $\mathbf{K}_l$ and $\mathbf{K}_0$) are predicted by the KPB

$$\mathbf{K}_l = \mathrm{Conv}(\varphi_l(\ldots\varphi_{j+1}([\mathbf{E}_j, \mathbf{F}_j]))), \tag{9}$$

$$\mathbf{K}_0 = \varphi_L(\ldots\varphi_{j+1}([\mathbf{E}_j, \mathbf{F}_j])), \tag{10}$$

where $\mathbf{F}_j = \phi_j(\ldots\phi_1(\mathbf{I}, \mathbf{M}))$ is the features from the $j$th layer of SIFB, and $\mathbf{E}_j = \varphi_j(\ldots\varphi_1(\mathbf{I}, \mathbf{M}))$ is from the $j$th layer of KPB. We add a convolutional layer (Conv($\cdot$)) to adjust the size of $\varphi_l(\ldots\varphi_{j+1}([\mathbf{E}_j, \mathbf{F}_j]))$ to meet the requirements of kernels. We show the whole framework in Fig. 2 (c). The kernels $\mathbf{K}_l$ and $\mathbf{K}_0$ are for the feature-level and image-level filtering, respectively. Intuitively, with Eq. (8), we conduct both semantic & image filtering in a single framework to fill large missing regions and enhance details. Moreover, with Eq. (9) and Eq. (10), all predicted kernels for semantic & image filtering are driven by the input image $\mathbf{I}$ and the deep feature $\mathbf{F}_j$, which contain all available spatial details and the understanding of the whole scene. As a result, both semantic information and detailed pixels can be properly reconstructed. With the new designs, our method achieves high-fidelity image inpainting. As shown in Fig. 4, MISF produces semantic face structures with rich details.

### 4.3. Relationship to Encoder-Decoder Network

In this section, we aim to explain the effectiveness of our method from the viewpoint of the encoder-decoder network. We can use the naive encoder-decoder network to perform the image inpainting directly. For example, we feed the corrupted image into an encoder and use a decoder to reconstruct the image. This process can be represented through Eq. (4) and Eq. (5). We can train the encoder-decoder network via the same loss functions like filtering.

From the perspective of the encoder-decoder network, our semantic filtering is an improved encoder-decoder network that contains an extra 'dynamic convolution layer' (See Fig. 2 (b)). MISF further makes the dynamic process conditional on the multi-level features. As a result, the parameters of the dynamic convolution are element-wise and dynamically tuned according to different images and their semantic meaning through the predictive network. The advantages of dynamic convolution have been evidenced in many works [6, 11]. However, these works mainly focus on the image classification task. They predict convolutional parameters dynamically, according to the input features. In contrast, our work presents the importance of dynamic convolution for image inpainting and predicts dynamic convolutional parameters based on the raw input and deep features jointly in an element-wise way. According to the results in Fig. 4, we see that the proposed dynamic operation is critical to the high-quality inpainting results. With the same training setups (See Sec. 4.4), the naive encoder-decoder network produces artifacts on the missing areas. The filled pixels lead to obvious structure mismatches, while semantic filtering can adapt to different scenes and fill pixels to have reasonable structures. Moreover, the complete model of MISF, which considers both semantic & image filtering, achieves much better results of semantic and detail recovery.

### 4.4. Implementation Details

**Network architectures.** Theoretically, we can conduct the MISF on all deep features. Nevertheless, this will lead to significant memory and time costs. Here, we use a 15-layer encoder-decoder, and only perform semantic filtering at the 3th layer (*i.e.*, $\mathbf{F}_3$). We detail the architectures in https://github.com/tsingqguo/misf and discuss semantic filtering on other features in the experiment section.

**Loss functions.** To get high-fidelity images in both image quality and semantic levels, we follow the work [23] and train the networks with four loss functions, *i.e.*, $L_1$ loss, GAN loss, Style loss, and perceptual loss. Specifically, given a corrupted image $\mathbf{I}$, the predicted completion $\hat{\mathbf{I}}$, and the ground truth $\mathbf{I}^*$, we have the loss function

$$\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}^*) = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_{\text{gan}} + \lambda_3\mathcal{L}_{\text{perc}} + \lambda_4\mathcal{L}_{\text{style}}. \quad (11)$$

We fix $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 0.1$, and $\lambda_4 = 250$. Please find the definitions of the loss functions in [23].

**Training details.** For all variants of our method, we use the same training setup: we employ Adam as the optimizer with the learning rate of 0.0001. We train the network for about 350,000 iterations with a batch size of 16. The experiments are implemented on the same platform with two NVIDIA Tesla V100 GPUs.

## 5. Experiments

### 5.1. Setups

**Datasets.** We evaluate our method on three datasets, *i.e.*, Places2 challenge dataset [35], CelebA dataset [20], and Dunhuang Challenge [33]. The Places2 dataset contains over eight million images captured under over 365 scenes. The CelebA dataset contains over 180 thousand face images. These datasets allow our method to be evaluated on the natural and facial scenes. The Dunhuang challenge provides practical data for image inpainting. We evaluate the proposed method on the standard test set of the CelebA and Dunhuang Challenge datasets. For the Places2 dataset, we follow the convention and choose 30,000 random images for testing.

**Metrics.** We follow the common setups in the image inpainting. We use the peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), $L_1$, and perceptual similarity (LPIPS [34]) for measuring the quality of image inpainting. PSNR, SSIM, and $L_1$ measures the quality of the recovered image. LPIPS measures the perceptual consistence between the recovered images and ground truth.

**Mask Setups.** For the Places2 and CelebA datasets, we use the irregular mask dataset [19], which has been used in many works [25], to generate the corrupted images. The mask images are classified into three categories (*i.e.*, $0\% - 20\%$, $20\% - 40\%$, and $40\% - 60\%$), based on the proportion of the image occupied by the holes. For the Dunhuang dataset, we follow its official setup.

Table 1. Comparison results on Places2, CelebA, and Dunhuang datasets. For PConv, the reported results are taken from [19].

| Datasets | Places2 | | | CelebA | | | Dunhuang | Places2 | | | CelebA | | | Dunhuang |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Mask Ratio | 0%-20% | 20%-40% | 40%-60% | 0%-20% | 20%-40% | 40%-60% | Default | 0%-20% | 20%-40% | 40%-60% | 0%-20% | 20%-40% | 40%-60% | Default |
| | PSNR ↑ | | | | | | | L1 ↓ | | | | | | |
| PConv [19] | 31.030 | 23.673 | 19.743 | - | - | - | - | 0.808 | 2.495 | 5.098 | - | - | - | - |
| StructFlow [25] | 29.047 | 23.092 | 19.408 | 31.618 | 25.283 | 20.829 | 35.199 | 0.976 | 2.811 | 5.444 | 0.737 | 2.171 | 4.533 | 0.475 |
| EdgeConnect [23] | 29.899 | 23.378 | 19.522 | 32.781 | 25.347 | 20.449 | 36.419 | 0.848 | 2.606 | 5.302 | 0.579 | 1.922 | 4.485 | 0.441 |
| RFRNet [17] | 29.281 | 22.589 | 18.581 | 33.573 | 25.635 | 20.539 | 36.485 | 1.009 | 3.218 | 6.719 | 0.521 | 1.811 | 4.346 | 0.401 |
| JPGNet [12] | 30.673 | 23.937 | 19.884 | 34.401 | 26.543 | 21.297 | 37.646 | 0.830 | 2.581 | 5.294 | 0.477 | 1.651 | 4.042 | 0.353 |
| CTSDG [14] | 30.658 | 23.701 | 19.751 | 32.677 | 24.945 | 20.123 | - | 1.568 | 4.987 | 10.29 | 1.161 | 3.972 | 9.231 | - |
| **MISF** | **31.335** | **24.239** | **20.044** | **34.494** | **26.635** | **21.553** | **38.383** | **0.726** | **2.340** | **4.965** | **0.474** | **1.616** | **3.826** | **0.341** |
| | SSIM ↑ | | | | | | | LPIPS ↓ | | | | | | |
| PConv [19] | 0.9070 | 0.7310 | 0.5325 | - | - | - | - | - | - | - | - | - | - | - |
| StructFlow [25] | 0.9343 | 0.8187 | 0.6740 | 0.9487 | 0.8598 | 0.7417 | 0.9559 | 0.0716 | 0.1714 | 0.2845 | 0.0746 | 0.1683 | 0.2662 | 0.0589 |
| EdgeConnect [23] | 0.9396 | 0.8225 | 0.6710 | 0.9586 | 0.8689 | 0.7362 | 0.9635 | 0.0572 | 0.1541 | 0.2748 | 0.0456 | 0.1265 | 0.2380 | 0.0480 |
| RFRNet [17] | 0.9283 | 0.7868 | 0.6137 | 0.9626 | 0.8746 | 0.7400 | 0.9648 | 0.0825 | 0.2161 | 0.3571 | 0.0400 | 0.1215 | 0.2335 | 0.0463 |
| JPGNet [12] | 0.9452 | 0.8348 | 0.6915 | 0.9674 | 0.8908 | 0.7697 | 0.9724 | 0.0817 | 0.2145 | 0.3535 | 0.0440 | 0.1316 | 0.2502 | 0.0469 |
| CTSDG [14] | 0.9451 | 0.8299 | 0.6768 | 0.9587 | 0.8649 | 0.7291 | - | 0.0525 | 0.1522 | 0.2714 | 0.0417 | 0.1267 | 0.2377 | - |
| **MISF** | **0.9506** | **0.8435** | **0.6931** | **0.9680** | **0.8911** | **0.7698** | **0.9735** | **0.0432** | **0.1298** | **0.2499** | **0.0315** | **0.0949** | **0.1911** | **0.0330** |

**Baselines.** We compare with five state-of-the-art inpainting methods, including PConv [19], StructFlow [25], EdgeConnect [23], RFR-Net [17], JPGNet [12], and CTSDG [14].

### 5.2. Comparison Results

**Quantitative comparison.** We compare our method with five state-of-the-art inpainting methods on three public datasets. As shown in Table 1, we have the following observations: ❶ Our method achieves better PSNR, SSIM, and $L_1$ scores across all datasets and mask ratios than other competitive methods. Compared to RFRNet, we achieve 7.01% relative higher PNSR under the $0\% - 20\%$ mask ratio on the Places2 dataset. Moreover, the relative gaps become larger, *i.e.*, 7.3% and 7.9%, under $20\% - 40\%$ and $40\% - 60\%$ mask ratios, respectively. It demonstrates that our method presents obvious advantages over existing methods on high-fidelity restoration. ❷ In terms of the LPIPS, we have similar observations. Our method gets 47.12% relative lower LPIPS than JPGNet under the $0\% - 20\%$ mask ratio on the Place2 dataset. This result demonstrates the impressive progress of our method on perceptual recovery. ❸ The consistent advantages across different datasets and mask ratios demonstrate that our method has high generalization capability.

**Qualitative comparison.** We provide the visualization results on five cases taken from three datasets (*i.e.*, CelebA, Place2, and Dunhuang) in Fig. 5 and Fig. 6. We find that: ❶ Our method generates more natural and high-fidelity images that are significantly close to the ground truth, even with large missing areas (See Fig. 5 case1). On the other hand, other methods introduce many artifacts like structure distortions and blur of large missing areas. ❷ Though all methods present similar results on small missing areas (*e.g.*, Fig. 6), our method provides fine-grained structures and recovers better details. For example, for the local structures around the green arrows in case3, RFRNet, EdgeCon., JPGNet and Str. Flow fail to restore the detailed structures. In contrast, our method completes all the details properly.

### 5.3. Ablation Study

**Quantitative results.** To validate the effectiveness of MISF, we consider three variants: Img-Filter (Sec. 3.1), Sem-Filter (Sec. 4.1), and MISF (Sec. 4.2). In Table 2 (Left), we observe that: ❶ By combining the image-level and semantic filtering, MISF yields better scores on all metrics across the three datasets and three missing sizes than other methods. ❷ Img-Filter yields poor scores on the Places2 and CelebA datasets but has good results in the Dunhuang dataset. Note that the results of Img-Filter on Dunhuang are even better than RFRNet. It is mainly because Img-Filter is good at completing small missing holes but failing to address large missing areas, as analyzed in Sec. 3.1.

**Qualitative results.** We visualize the results of Img-Filter in Fig. 3. It shows that Img-Filter completes small missing areas effectively. However, Img-Filter is less effective handling large missing areas. We also compare Sem-Filter and MISF in Fig. 4. It presents Sem-Filter loses many details while MISF produces rich details with natural structures.

### 5.4. Discussion

**Filtered vs. Pre-Filtered Features.** Given a corrupted image and its ground truth (GT), we feed them to MISF and get their deep features before & after filtering, respectively. Then, we calculate the similarity via cross-correlation between the filtered (or pre-filtered) features of the corrupted image and GT (See in Fig. 7). We randomly sample 1000 examples from different mask ratios and the whole dataset, respectively, and calculate each example's similarity. After filtering, the features of corrupted image become close to the ones of GT. As the mask ratio is larger, the similarity margin becomes larger, which infers the effectiveness of semantic filtering across different mask sizes.

**Semantic filtering with different deep features.** As the results reported in Table 2 (Right), the completion performance generally becomes better when the feature for semantic filtering is deeper. We have consistent conclusions on the four metrics and three missing sizes. This is because deeper features have better semantic representations. Nevertheless, whether the growth would continue as the network become deeper needs further study in the future.

**Importance of dynamic convolutional operations for**

**Table 2.** Left: Ablation study results on Places2, CelebA, and Dunhuang datasets. Right: Sem-Filter with different deep features on CelebA dataset.

| | Datasets | Places2 | | | CelebA | | | Dunhuang | | CelebA | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mask Ratio | 0%-20% | 20%-40% | 40%-60% | 0%-20% | 20%-40% | 40%-60% | Default | Latent Layer | 0%-20% | 20%-40% | 40%-60% |
| PSNR ↑ | Img-Filter | 25.489 | 17.663 | 13.773 | 27.314 | 19.020 | 14.837 | 37.021 | Sem-Filter($\mathbf{F}_1$) | 33.981 | 26.364 | 21.360 |
| | Sem-Filter | 31.010 | 24.117 | 19.944 | 34.253 | 26.518 | 21.486 | 37.897 | Sem-Filter($\mathbf{F}_2$) | 34.128 | 26.429 | 21.353 |
| | **MISF** | **31.335** | **24.239** | **20.044** | **34.494** | **26.635** | **21.553** | **38.383** | Sem-Filter($\mathbf{F}_3$) | **34.253** | **26.518** | **21.486** |
| | En-decoder-Filter | 31.187 | 24.107 | 19.898 | 34.330 | 26.484 | 21.428 | 38.195 | | | | |
| | En-decoder | 30.824 | 23.980 | 19.871 | 33.745 | 26.122 | 21.077 | 37.766 | | | | |
| SSIM ↑ | Img-Filter | 0.9180 | 0.7571 | 0.5911 | 0.9298 | 0.7830 | 0.6301 | 0.9692 | Sem-Filter($\mathbf{F}_1$) | 0.9651 | 0.8856 | 0.7612 |
| | Sem-Filter | 0.9487 | 0.8409 | 0.6910 | 0.9657 | 0.8871 | 0.7631 | 0.9696 | Sem-Filter($\mathbf{F}_2$) | 0.9651 | 0.8852 | 0.7589 |
| | **MISF** | **0.9506** | **0.8435** | **0.6931** | **0.9673** | **0.8909** | **0.7693** | **0.9735** | Sem-Filter($\mathbf{F}_3$) | **0.9657** | **0.8871** | **0.7631** |
| | En-decoder-Filter | 0.9499 | 0.8420 | 0.6907 | 0.9661 | 0.8871 | 0.7614 | 0.9734 | | | | |
| | En-decoder | 0.9466 | 0.8358 | 0.6841 | 0.9632 | 0.8805 | 0.7510 | 0.9711 | | | | |
| L1 ↓ | Img-Filter | 1.535 | 5.477 | 11.42 | 1.264 | 4.750 | 10.44 | 0.386 | Sem-Filter($\mathbf{F}_1$) | 0.503 | 1.692 | 3.977 |
| | Sem-Filter | 0.750 | 2.370 | 4.995 | 0.488 | 1.651 | 3.895 | 0.383 | Sem-Filter($\mathbf{F}_2$) | 0.497 | 1.673 | 3.952 |
| | **MISF** | **0.726** | **2.340** | **4.965** | **0.474** | **1.616** | **3.826** | **0.341** | Sem-Filter($\mathbf{F}_3$) | **0.488** | **1.651** | **3.895** |
| | En-decoder-Filter | 0.732 | 2.360 | 5.022 | 0.487 | 1.657 | 3.909 | 0.345 | | | | |
| | En-decoder | 0.769 | 2.432 | 5.104 | 0.518 | 1.740 | 4.117 | 0.362 | | | | |
| LPIPS ↓ | Img-Filter | 0.1154 | 0.3129 | 0.5168 | 0.1084 | 0.2910 | 0.4631 | 0.0508 | Sem-Filter($\mathbf{F}_1$) | 0.0359 | 0.1038 | 0.2043 |
| | Sem-Filter | 0.0465 | 0.1333 | 0.2527 | 0.0343 | 0.1008 | 0.2009 | 0.0379 | Sem-Filter($\mathbf{F}_2$) | 0.0355 | 0.1021 | 0.2028 |
| | **MISF** | **0.0432** | **0.1298** | **0.2499** | **0.0315** | **0.0949** | **0.1911** | **0.0330** | Sem-Filter($\mathbf{F}_3$) | **0.0343** | **0.1008** | **0.2009** |
| | En-decoder-Filter | 0.0444 | 0.1328 | 0.2540 | 0.0346 | 0.1016 | 0.2015 | 0.0332 | | | | |
| | En-decoder | 0.0497 | 0.1416 | 0.2637 | 0.0370 | 0.1066 | 0.2104 | 0.0373 | | | | |

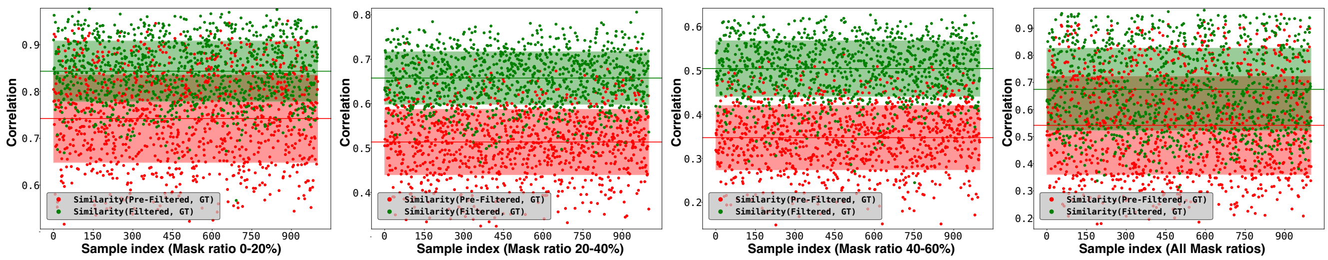


**Figure 7.** Similarity of filtered and pre-filtered features to ground truth features.

**the encoder-decoder network.** Our method can be regarded as an advanced encoder-decoder network, which contains dynamic convolutional operations. To validate this, we conduct an ablation study from the viewpoint encoder-decoder network. Specifically, we compare four variants, *i.e.*, En-decoder, En-decoder-Filter, Sem-Filter, and MISF. The first one is the naive encoder-decoder network without the dynamic convolutional operation. The second variant is built by adding an image-level predictive filtering to the output of En-decoder. As a result, En-decoder-Filter and Sem-Filter can be regarded as the encoder-decoder network that contains a single dynamic convolutional operation. MISF has two dynamic convolutional operations. Comparing all variants, we see that the networks with more dynamic convolutional operations yield better inpainting results under the four metrics across all datasets and missing areas.

## 6. Conclusions

We proposed multi-level interactive siamese filtering (MISF) for high-fidelity image inpainting. We used a single predictive network to conduct predictive filtering at the image level and deep feature level, simultaneously. The image-level filtering is to recover details while the deep feature-level filtering is to complete semantic information, which leads to high-fidelity inpainting results. In addition, the dynamically predicted kernels make our method have high generalization capability. Our method outperforms state-of-the-art methods on three public datasets. Furthermore, the extensive experiments demonstrate the effectiveness of different components of our approach. One potential limitation of this work is that we validate and train our model on the widely used public datasets that may only cover a part of the real-world scenarios. In the future, we could develop our model to see more scenarios, *i.e.*, the cloud removal for remote sensing images [29], and further enhance its generalization capability.

**Acknowledgments**: This work was supported by the National Natural Science Foundation of China under Grant U1803264, 62072334, and the General Project of Tianjin (No. 18JCYBJC15200). It was also supported in part by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-019), Singapore National Cybersecurity R&D Program No. NRF2018NCR-NCR005-0001, National Satellite of Excellence in Trustworthy Software System No.NRF2018NCR-NSOE003-0001, and NRF Investigatorship No. NRFI06-2020-0022-0001. We gratefully acknowledge the support of NVIDIA AI Tech Center (NVAITC) and AWS Cloud Credits for Research Award.

## References


[1] Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony DeRose, and Fabrice Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Transactions on Graphics*, 36(4):97:1–97:14, 2017.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 28(3):24, 2009.

[3] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[4] T. Brooks and J. T. Barron. Learning to synthesize motion blur. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6833–6841, 2019.

[5] Tony F. Chan and Jianhong Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.

[6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11030–11039, 2020.

[7] Ding Ding, Sundaresh Ram, and Jeffrey J Rodríguez. Image inpainting using nonlocal texture matching and nonlinear filtering. *IEEE Transactions on Image Processing*, 28(4):1705–1719, 2018.

[8] Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks VS Lakshmanan, and Xuemin Lin. Efficient algorithms for densest subgraph discovery. *arXiv preprint arXiv:1906.00341*, 2019.

[9] Lan Fu, Changqing Zhou, Qing Guo, Felix Juefei-Xu, Hongkai Yu, Wei Feng, Yang Liu, and Song Wang. Auto-exposure fusion for single-image shadow removal. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[11] Qing Guo, Felix Juefei-Xu, Changqing Zhou, Yang Liu, and Song Wang. Sparta: Spatially attentive and adversarially robust activation. *arXiv preprint arXiv:2105.08269*, 2021.

[12] Qing Guo, Xiaoguang Li, Felix Juefei-Xu, Hongkai Yu, Yang Liu, et al. Jpgnet: Joint predictive filtering and generative network for image inpainting. *arXiv e-prints*, pages arXiv–2107, 2021.

[13] Qing Guo, Jingyang Sun, Felix Juefei-Xu, Lei Ma, Xiaofei Xie, Wei Feng, Jianjun Zhao, and Yang Liu. Efficientderain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining. In *AAAI Conference on Artificial Intelligence*, 2021.

[14] Xiefan Guo, Hongyu Yang, and Di Huang. Image inpainting via conditional texture and structure dual generation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14134–14143, 2021.

[15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36(4):1–14, 2017.

[16] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[17] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent Feature Reasoning for Image Inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7757–7765, 2020.

[18] Kangshun Li, Yunshan Wei, Zhen Yang, and Wenhua Wei. Image inpainting algorithm based on tv model and evolutionary algorithm. *Soft Computing*, 20(3):885–893, 2016.

[19] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3730–3738, 2015.

[21] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2502–2510, 2018.

[22] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[23] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Qureshi, and Mehran Ebrahimi. Edgeconnect: Structure guided image inpainting using edge prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[24] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context Encoders: Feature Learning by Inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[25] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[26] Mehdi SM Sajjadi, Bernhard Scholkopf, and Michael Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4491–4500, 2017.

[27] Linsen Song, Jie Cao, Lingxiao Song, Yibo Hu, and Ran He. Geometry-aware face completion and editing. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 2506–2513, 2019.

[28] Zhaoyi Yan, Xiaoming Li, Mu Li, Wangmeng Zuo, and Shiguang Shan. Shift-net: Image inpainting via deep feature rearrangement. In *European Conference on Computer Vision (ECCV)*, pages 1–17, 2018.

[29] Jingyu Yang, Jianhua Guo, Huanjing Yue, Zhiheng Liu, Haofeng Hu, and Kun Li. Cdnet: Cnn-based cloud detec-

tion for remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):6195–6211, 2019.

[30] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5485–5493, 2017.

[31] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5505–5514, 2018.

[32] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4471–4480, 2019.

[33] Tianxiu Yu, Shijie Zhang, Cong Lin, Shaodi You, Jian Wu, Jiawan Zhang, Xiaohong Ding, and Huili An. Dunhuang grottoes painting dataset and benchmark. *arXiv preprint arXiv:1907.04589*, 2019.

[34] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[35] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.